Speaker–Independent Word Recognition with Backpropagation Networks

Bernd Freisleben
Dept. of Computer Science
University of Darmstadt
Alexanderstr. 10
D–6100 Darmstadt, Germany
Email: freisleb@isa.informatik.th-darmstadt.de

Christian–Arved Bohn
Dept. Scientific Visualization of HLRZ
GMD Birlinghoven
P.O. Box 1316
D–5205 Sankt Augustin 1, Germany
Email: bohn@viswiz.gmd.de

Abstract

This paper presents a system that recognizes a limited vocabulary of spoken words in a speaker–independent manner. The system requires only minimal hardware support for acoustic preprocessing. In contrast to other approaches to word–level recognition, it reduces the information content of the speech signals by a compression algorithm before presenting them as inputs to a standard 3–layer backpropagation network. The network learns to recognize the utterances of the speakers in the training set, and the trained network is then used to recognize the spoken words of unknown speakers. Recognition rates of up to 91% were obtained for unknown speakers of the same sex and up to 72% for a mix of both male and female speakers. Since the training times are fast and the system is very cost effective, the approach is practically feasible for a variety of applications.

1 Introduction

The ability to identify spoken words is desirable in a variety of application areas, such as manufacturing, telecommunication and medicine [12], but high–quality speech recognition systems are not easy to built. The challenging computational problems associated with speech recognition and the limited success of the conventional pattern matching techniques proposed to solve them have fostered the development of neural network approaches to speech recognition tasks. The intention is that the generalization properties of neural network learning algorithms are useful to improve the recognition performance.

The proposals made in the literature differ mainly in how the speech signals are converted to a format which can be used as the network input and what the network should learn to recognize. For example, some of the proposals focus on phoneme recognition networks [6, 8, 17], the outputs of which are then processed by further networks or other techniques, such as hidden markov models [3, 14], to achieve word level recognition, while others attempt to recognize words directly [1, 4, 5, 9, 10, 16]. The phoneme recognition systems are conceptually capable of recognizing an unlimited number of words, but they are far more complex, time consuming to develop and difficult to use than word recognition networks, due to the large postprocessing overhead associated with them.

This paper presents an automatic speech recognition system which achieves speaker–independent recognition of a limited vocabulary of spoken words. The system is based on only low cost hardware components (a microphone, a preamplifier and an 8–bit A/D converter) for preprocessing the speech signals, and at the heart of its software design is a backpropagation neural network [15]. The network learns to recognize the spoken words of a set of speakers and is then used to recognize the words of unknown speakers. Several experiments with a speech database containing the recordings of 45 German words from each of 16 different male and female speakers have been conducted to test the performance of the network. The results indicate that the network succeeds in learning all training sets perfectly, but in an experiment with speakers of the same sex it needs to "listen" to 7 speakers before it can recognize the words of unknown speakers with a satisfactory recognition accuracy (91%). For a mix of both male and female speakers recognition rates of up to 72% were obtained. The performance of our system appears to be quite competitive to other results reported in the literature on speaker–independent word recognition, particularly when considering that the number of words included in the speech data used in these approaches was significantly smaller than in our proposal. Apart from the hardware components mentioned above, our system has been fully implemented in *C*. Since the training times of our network are fast and the system is very cost effective, the proposed approach is applicable to various application scenarios.

The paper is organized as follows. Section 2 describes the speech data and the steps taken to preprocess the speech signals in order make them amenable as neural network inputs. In section 3 the network architecture used is presented. The implementation of the proposed speech recognition system and the performance results obtained in the experiments are discussed in section 4. Section 5 concludes the paper and outlines areas for further research.

2 Preprocessing the Speech Data

The speech data used consists of isolated utterances of 45 German words which represent commands for controlling a simple drawing program. A total of 10 male and 6 female speakers were recorded in ambient noise conditions (a normal office environment) with a low cost microphone, preamplified by a 7 kHZ low–pass filter and digitized with an 8–bit analog–to–digital converter sampled at 15625 Hz. Apart from these three components, no other hardware equipment was used to preprocess the speech signals. Thus, our system is very cost effective in comparison to other approaches where higher precision A/D converters or fast signal processors were employed [8].

The individual steps taken to further process the digitized version of each speaker's recording, the sequence of all 45 words with short silences in between them, are graphically illustrated in Figure 1.

In the first step, the preprocessing software of our system automatically extracts the individual words from the speech signal under consideration (1). The extraction algorithm developed succeeds in finding the word boundaries with an error rate of less than 5%, which is quite satisfactory considering that no provisions have been taken to avoid environmental noise.

A 512–point fast Fourier transform analysis, computed every 10.9 milliseconds using a Hamming window [13] (2/3 overlap between successive windows), is then performed to obtain the short–time frequency spectrums of each extracted word (2).

Each spectrum is subsequently transformed into a 15–dimensional speech vector (3) by integrating the (logarithmicly scaled) spectral amplitudes centered at the following frequencies in Hz (taken from [5]; bandwith indicated in parentheses): 130 (30), 164 (38), 206 (48), 260 (60), 327 (76), 412 (96), 520 (121), 655 (152), 828 (192), 1040 (242, 1310 (305), 1650 (384), 2078 (485), 2619 (611), 3300 (770).

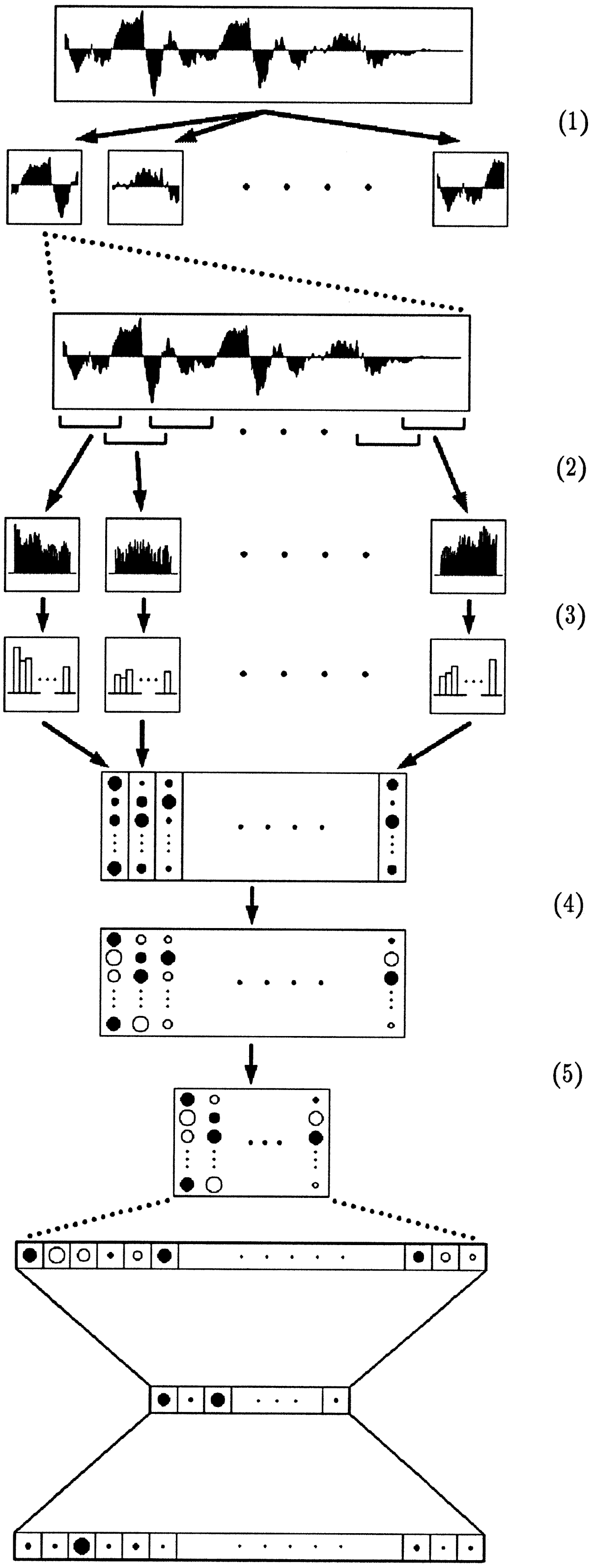
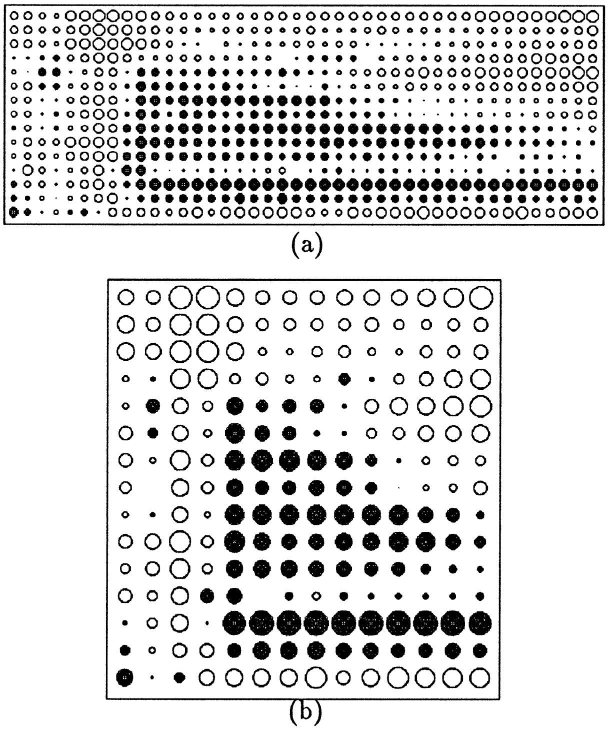


Figure 1: Preprocessing stages

The resulting vectors, between 40 and 80 for each word, are normalized to the interval [-0.5, 0.5] (4).

In a final step, the sequence of speech vectors of each word is compressed in time by accumulating and averaging them until the sum of their distances exceeds a threshold (5), as proposed in [2]. When this threshold is reached, the sequence of speech vectors is replaced by the average value, leading to 10–17 of such 15–dimensional speech vectors for each word considered. These are used as the neural network input.

The use of a compression algorithm is one of the major differences of our approach to other word recognition endeavours [1, 4, 5, 9, 10, 16] which seem to be based on the assumption that as much as possible of the speech information should be kept to achieve high recognition rates. However, compression does not only reduce the dimensionality of the neural network inputs, which enables the network to learn faster, but also provides a more uniform representation of the utterances, which seems to be beneficial for improving the generalization ability of the network. Figure 2 shows the normalized speech vectors of the German word *"zwei"* before (a) and after compression (b). Large filled circles represent large positive values, and large unfilled circles represent large negative values. The individual speech vectors are displayed vertically, with the lowest frequency at the bottom. The word starts at the left hand side, and the time axis is along the horizontal direction.

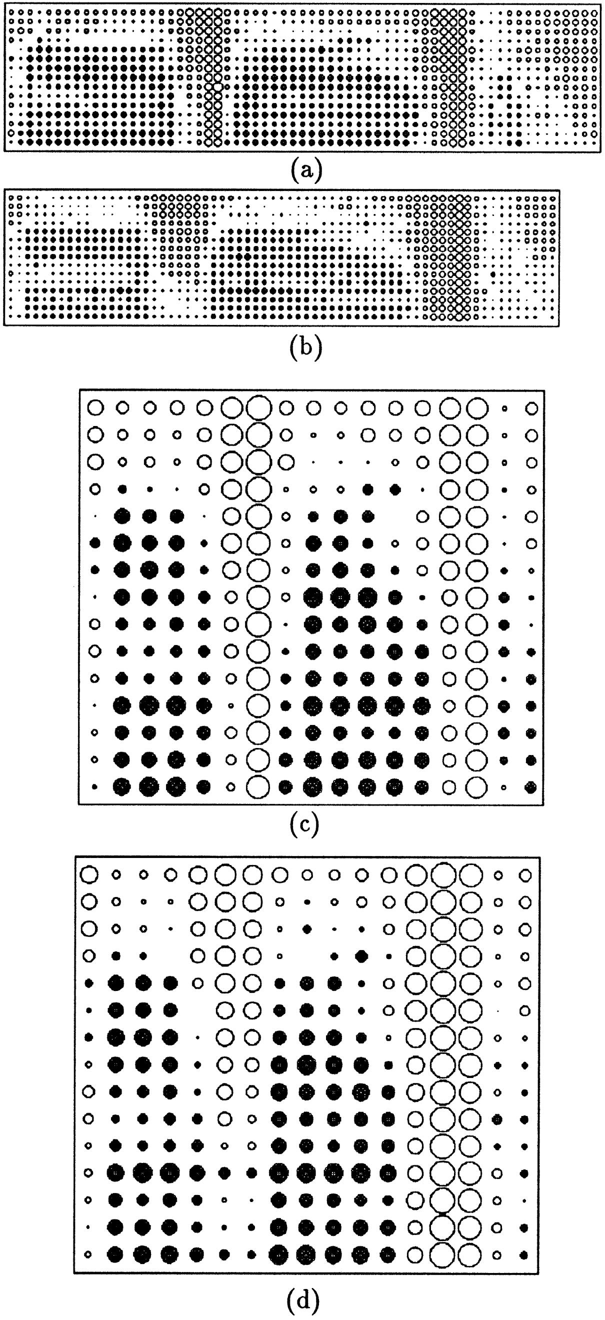


(a)

(b)

Figure 2: The German word *"zwei"* before (a) and after compression (b)

It is evident that the characteristics of the original word are retained in the compressed version, although the information content is reduced by the compression algorithm. Moreover, small variations in the utterances and the effects of different rates of speech are somewhat eliminated through compression, as shown in Figure 3, where the speech vectors of the German word *"arbeit"*, spoken by the same speaker at two different speeds, are displayed before (a),(b) and after compression (c),(d).



(a)

(b)

(c)

(d)

Figure 3: The German word *"arbeit"* before (a),(b) and after compression (c),(d)

3 Network Architecture

We have studied several network architectures with different numbers of hidden units/layers and parameter settings for the single speaker word recognition task and used the net which gave the best performance results in all experiments. The resulting network architecture is a 3–layer feed–forward network with 240 input units, 18 hidden units and 45 output units. The input layer receives the speech vectors of one word ordered into a linear array and the output layer uses a simple 1–out–of–45 coding where the output unit with the highest activation corresponds to the word recognized by the network, as shown at the bottom of Figure 1. If the linearly arranged speech vectors of a word yield a neural network input vector with less than 240 components, the remaining components are set to 0.0.

The learning rule is the standard backpropagation algorithm with the following properties: a) the usual quadratic error function as described in [7] is used; b) errors are accumulated after each input in the training set; c) the learning rate lies between 0.4 and 0.9; d) the momentum term is 0.7; e) the weights are initialized to random values in the range between -0.3–0.3; and f) the input vectors are always presented in the same order.

4 Implementation and Performance

The software for the entire system was written in *C*, and the utterances were recorded and digitized on a Commodore Amiga, equipped with a low cost microphone and an off–the–shelf 8-bit A/D module. A SUN Sparcstation was later used to perform all computations on the digitized speech signals, including preprocessing and neural network training.

In order to test the recognition performance of the network, two different problem areas have been investigated: single–speaker word recognition and multi–speaker word recognition. The results obtained are presented in the following subsections.

4.1 Single–Speaker Word Recognition

In the single–speaker word recognition experiment, all 45 words were spoken by and recorded from a single speaker four times, resulting in four different input sets. Three of them were used as the training set, while the remaining one served as the test set. After only 20–30 presentations of the training set (equivalent to 2–4 minutes computation time on a SUN Sparcstation), the error function was minimized, i.e. the backpropagation algorithm converged pretty fast. This compares favourably to other approaches where several hundreds of iterations were required [4, 16].

The recognition accuracy of the training set was 100%, i.e. the network had properly learned all 45 words. A recall with the test set achieved a recognition rate of 96%, which seems to be slightly superior to the performance results reported in the literature for the single–speaker word recognition problem (see section 4.3).

4.2 Multi–Speaker Word Recognition

The speech data used for the multi–speaker word recognition task consisted of 16 different speakers (10 male and 6 female), each of them recorded once for all 45 words.

In a first experiment, only male speakers were considered. The 10 available input sets were divided into two groups, the first group being the training set and the second group being the test set. Table 1 shows the performance results when the training set contained the speech vectors of 1, 2, 4 and 7 speakers, where in each case the remaining speakers were used as the test set (the percentages represent averages of the speakers in the test set).

#words	#speakers	training set	test set
45	1	100%	37%
45	2	100%	49%
45	4	100%	70%
45	7	100%	91%

Table 1: Recognition rates for male speakers

The results indicate that the network succeeded in learning all training sets perfectly, but it needs to "listen" to 7 speakers before it can recognize the words of unknown speakers with a satisfactory recognition accuracy (91%).

In a second experiment, both male and female speakers were considered. The training and the test sets were assembled in the manner described above. The results are presented in Table 2.

#words	#speakers	training set	test set
45	2	100%	42%
45	4	100%	56%
45	6	100%	59%
45	8	100%	68%
45	10	100%	72%

Table 2: Recognition rates for male and female speakers

The performance results obtained for the training sets are identical to the experiment where only male speakers were investigated. The recognition accuracy for the test sets is not as good, but still acceptable. It should be mentioned that in most cases the "correct" output units had almost equally high activations as the (wrongly) winning unit.

The results suggest that it might be conceivable to use two separate networks, one for male and the other one for female speakers, in order to achieve high recognition accuracies for unknown speakers independent of their sexes.

4.3 Discussion

In this section we discuss some of the observations made during the experiments and compare our results to other results reported in the literature for the word-level recognition.

The number of hidden units seems to have some impact on the generalization ability of the network. Our experiments indicate that a large number of hidden units is usually beneficial to improve the recognition rates of the training set, but leads to worse results for the test set. Since a large number of hidden units also decreases the convergence speed, less hidden units are clearly favourable.

The network behaviour is relatively immune to the settings of the learning and momentum parameters, because the results did not significantly change when the parameters where modified (in a reasonable range). The presentation of the input vectors in a different order, in some applications useful to avoid local minima, did not have an observable effect on the convergence of the network.

Table 3 summarizes the results of other approaches to word–level recognition, in order to allow a comparison to the recognition rates obtained with our proposal.

#words	#speakers	training	test	ref.
146	1	90.6%	58.2%	[16]
17	3	100%	90%	[5]
17	3	100%	94%	[5]
13	1	100 %	92%	[9]
10	1	97.5%	70–100%	[1]
7	1	—	92%	[11]
5	3	84%	62%	[4]

Table 3: Recognition rates of other approaches

Our results are quite competitive to those shown in Table 3. It is worth noting that the speech database in the majority of these approaches (except for the first one) contained fewer words than our speech database, but nevertheless the 96% recognition rate obtained with our network for the test set in the single–speaker case is slightly better, and the 91% recognition rate for multi–speaker word recognition is similar. When our network is trained to recognize up to 30 words, recognition accuracies of 100% for both the training and the test set are obtained. We cannot compare our results for a mix of both male and female speakers to the above approaches, because they do not include information whether similar experiments have been conducted.

The recognition rates alone, however, might not be sufficient to allow a fair comparison, because the recognition rates are clearly dependent on the recording conditions and the articulation of the words. It is always possible to pronounce a word in a manner such that the network is not able to recognize it, and on the other hand, the recognition rate will be improved when particular emphasis is put on pronouncing the same words similarly. A description of the relevant details of the experimental conditions is not provided in the literature on the alternative approaches, but we did not at all try to eliminate environmental noise or take influence on the speakers' pronounciation of the words.

To summarize, our work demonstrates the practical feasibility of building a low cost but high quality speech recognition system which can be used in a flexible manner. The system can be easily trained to recognize the desired words with high accuracy and does not require the assistance of somebody who is deeply familiar with the issues involved in speech processing. It is well suited for applications where a limited vocabulary needs to be recognized, such as in some control applications with voice entry of commands or in tools for supporting the physically handicapped. The system can also be used for *speaker* recognition [12] by simply training the network to learn the mapping between a set of words and a number of speakers, and letting the network recognize who is saying unknown words contained in a test set. Experiments with 5 speakers and a test set of 50 unknown utterances have shown that the network is able to identify the speakers in up to 82% of the cases correctly.

The time for running the preprocessing software and training the network with the backpropagation algorithm is pretty short (about 10 minutes), and it may further be reduced by simply implementing parts of the preprocessing functionality in hardware. An appropriately trained network would then allow word recognition under real–time conditions.

5 Conclusions

In this paper we have presented a speech recognition system that allows to recognize a limited vocabulary of spoken words in a speaker–independent manner. Apart from a few low cost hardware components required for acoustic preprocessing, the system has been implemented in software. A standard 3-layer backpropagation neural network has been used to learn the utterances of the words from a set of speakers, and the trained network was employed to recognize the spoken words of unknown speakers. Before the suitably preprocessed speech signals were presented to the input units of the network, their information content was reduced by a compression algorithm, leading to improvements of the generalization ability and the convergence speed of the network. Experiments have shown that the network performance is quite competitive to other approaches: recognition rates of up to 91% were obtained for unknown speakers of the same sex and up to 72% for a mix of both male and female speakers. Since the system is very cost effective, it is useful in a number of applications.

Among the issues for future research are an evaluation of the network performance when the size of the speech database is increased, the integration of the system in particular application environments, and the study of other learning architectures, such as recurrent networks [7], for word recognition.

References

[1] Behme H, *'A Neural Net for Recognition and Storing of Spoken Words'*, In: Parallel Processing in Neural Systems and Computers, pp. 379–382, Elsevier Science Publishers, 1990.

[2] Bengio Y, Cardin R, and De Mori R, *'Speaker Independent Speech Recognition with Neural Networks and Speech Knowledge'*, In: Advances in Neural Information Processing Systems, Vol. 2, pp. 218–225, Morgan Kaufman Publishers, 1990.

[3] Bourlard H, and Morgan N, *'A Continuous Speech Recognition System Embedding MLP into HMM'*, In: Advances in Neural Information Processing Systems, Vol. 2, pp. 186–193, Morgan Kaufman Publishers, 1990.

[4] Franzini M A, *'Learning to Recognize Spoken Words: A study in Connectionist Speech Recognition'*, In: Proceedings of the 1988 Connectionist Models Summer School, pp. 407–416, Morgan Kaufman Publishers, 1988.

[5] Grajski K A, Witmer D P, and Chen C, *'A Preliminary Note on Static and Recurrent Neural Networks for Word–Level Speech Recognition'*, In: Proceedings of the 1990 International Joint Conference on Neural Networks, Vol. 2, pp. 245–248, Lawrence Erlbaum Publishers, 1990.

[6] Hampshire II J B, and Waibel A, *'Connectionist Architectures for Multi–Speaker Phoneme Recognition'*, In: Advances in Neural Information Processing Systems, Vol. 2, pp. 203–210, Morgan Kaufman Publishers, 1990.

[7] Hertz J A, Krogh A, and Palmer R, *'Introduction to the Theory of Neural Computation'*, Addison–Wesley, Reading, Massachusetts, 1991.

[8] Kohonen T, *'The Neural Phonetic Typewriter'*, IEEE Computer, 3:11–22, 1988.

[9] Kowalewski F, and Strube H, *'Word Recognition with a Recurrent Neural Network'*, In: Parallel Processing in Neural Systems and Computers, pp. 390–394, Elsevier Publishers, 1990.

[10] Lee K, *'Context–Dependent Phonetic Hidden Markov Models for Speaker–Independent Continuous Speech Recognition'*, IEEE Transactions on Acoustics, Speech, and Signal Processing, 38(4), 1990.

[11] Lee Y, and Lippmann R P, *'Practical Characteristics of Neural Network and Conventional Pattern Classifiers on Artificial and Speech Problems'*, In: Advances in Neural Information Processing Systems, Vol. 2, pp. 168–177, Morgan Kaufman Publishers, 1990.

[12] Peacocke R D, and Graf D H, *'An Introduction to Speech and Speaker Recognition'*, IEEE Computer, 8:26–33, 1990.

[13] Rabiner L R, and Gold B, *'Theory and Applications of Digital Signal Processing'*, Prentice–Hall, 1975.

[14] Rigoll G, *'Neural Network Based Continous Speech Recognition by Combining Self Organizing Maps and Hidden Markov Modelling'*, In: Lecture Notes in Computer Science, Vol. 134, pp. 58–65, Springer–Verlag, Berlin, 1990.

[15] Rumelhart, D E, Hinton, G, and Williams, R E, *'Learning Internal Representations by Error Propagation'*, In: Parallel Distributed Processing: Explorations in the Microstructures of Cognition, Vol. 1, 318–362, MIT Press

[16] Sung C, and Jones W C, *'A Speech Recognition System Featuring Neural Network Processing of Global Lexical Features'*, In: Proceedings of the 1990 International Joint Conference on Neural Networks, Vol. 2, pp. 437–440, Lawrence Erlbaum Publishers, 1990.

[17] Waibel A, Hanazawa T, Hinton G, Shikano K, and Lang K, *'Phoneme Recognition Using Time-Delay Neural Networks'*, IEEE Transactions on Acoustics, Speech, and Signal Processing, 37(3):328–339, 1989.