Performance of a Backpropagation Neural Network in Diagnostic Rhyme Test Word Recognition

Chit-Sang Tsang and Carlos R. Villamar

SIMULATION 1998; 70; 167
DOI: 10.1177/003754979807000303

TECHNICAL ARTICLE

SIMULATION 70:3, 167-174

ISSN 0037-5497/98 $3.00 + .10
Printed in the United States of America

Performance of a Backpropagation Neural Network in Diagnostic Rhyme Test Word Recognition

Chit-Sang Tsang
Department of Electrical Engineering
California State University
Long Beach, California 90840

Carlos R. Villamar
Neural Solutions
43727 Lively Street
Lancaster, California 93536

This paper investigates isolated speaker-dependent word recognition of Diagnostic Rhyme Test words using a backpropagation neural network classifier. The performance of K-nearest neighbors and closest-class-mean classifiers are compared for several signal-to-noise ratios. The test and training data consisted of 40 frames of weighted eighth-order cepstral coefficients extracted from each word utterance. The neural network classifier correctly classified more than 92% of 2,400 testing examples not contained in the training data for the noise-free case. This performance was better than that of the K-nearest neighbor classifier, which was greater than 83%, and that of the closest class mean classifier, which was greater than 85%.

Keywords: Neural networks, backpropagation, speech recognition, diagnostic rhyme test, DRT

1. Introduction

In this paper Diagnostic Rhyme Test (DRT) words are used for evaluation of a speech recognition system in an attempt to standardize test words for evaluation of such systems, leading to reproducible results in subsequent research. The DRT has achieved wide acceptance for evaluating digital voice communication systems in correctly transmitting basic phonemic attributes—voicing, nasality, sustention, sibilation, graveness and compactness [1]. Sixteen word pairs for each classification of phonemic attributes comprise the corpus of stimulus words [2] used in the DRT (96 words total). Individual word pairs differ only in their initial phonemes (e. g., *bean* versus *peen*). The following 12 DRT words are randomly chosen for this paper:

Voicing:	Voiced—bean
	Unvoiced—peen
Nasality:	Nasal—need
	Oral—deed
Sustention:	Sustained—sheet
	Interrupted—cheat
Sibilation:	Sibilated—sing
	Unsibilated—thing
Graveness:	Grave—moon
	Acute—noon
Compactness:	Compact—coop
	Diffuse—poop

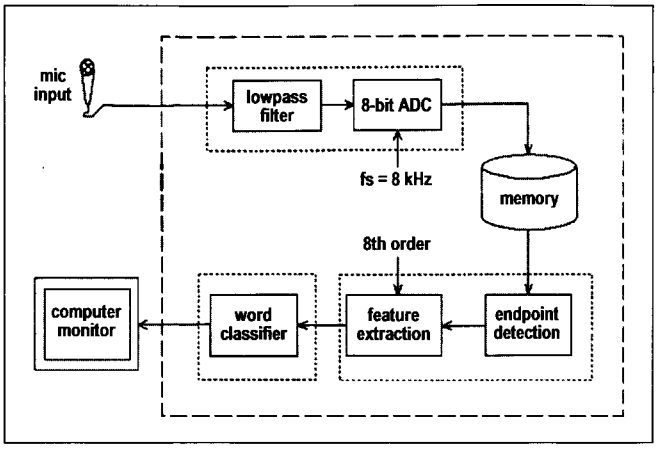


Figure 1. Speech recognition system block diagram

2. System Architecture

A generic speech recognition system is given in Figure 1. The system includes units for speech input, feature extraction and word classification. The function performed by each unit is described in the following sections. The system could be implemented by personal computer with common peripherals such as a microphone and sound card.

3. Time and Frequency Domain Plots

DRT words used as a stimulus for speech recognition systems quantitatively measure the performance of the speech recognition system in precisely determining the presence or absence of a given phonetic attribute. Sample time and frequency domain plots for the DRT words used in this experiment are shown in Figures 2 through 13. Please note that in some cases the amplitude scale is not consistent for a given pair of plots. This inconsistency was due to the auto-scaling feature in the signal processing software used to generate these plots. It is important to note the general shape of the plots in the time domain and frequency domains.

4. Endpoint Detection

Endpoint detection of a speech signal is the process used to determine the start and endpoints of the speech word. Endpoint detection could be nontrivial, especially for the case of continuous speech where the signal of the contiguous words may not be clearly separated. For isolated word recognition, this process is much easier. An endpoint detection procedure is described in references [3] and [4]. The endpoints are first determined based on the energy of the speech relative to the silence energy where the background noise is dominated. The background noise energy level is used to determine a threshold, which in turn is used to determine the existence of a word. If the speech energy starts to cross this threshold, we tentatively assign it to be the starting point of a word. When the energy falls below the threshold, we tentatively assign it to be the endpoint of a word.

A decision based solely on the energy threshold is not necessarily accurate because speech energy within a word may sometimes fall below the energy threshold. A more accurate decision requires knowledge of the rate of zero-crossing of the signal. This is due to the fact that the unvoiced speech may have relatively low energy, but exhibits a high zero-crossing rate. If the zero-crossing rate is high beyond

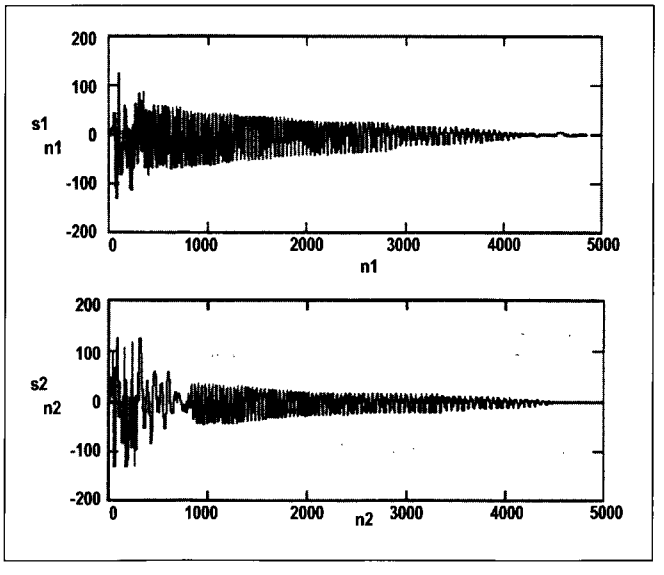


Figure 2. Time domain plot of "bean" and "peen" (s1 and s2, respectively)

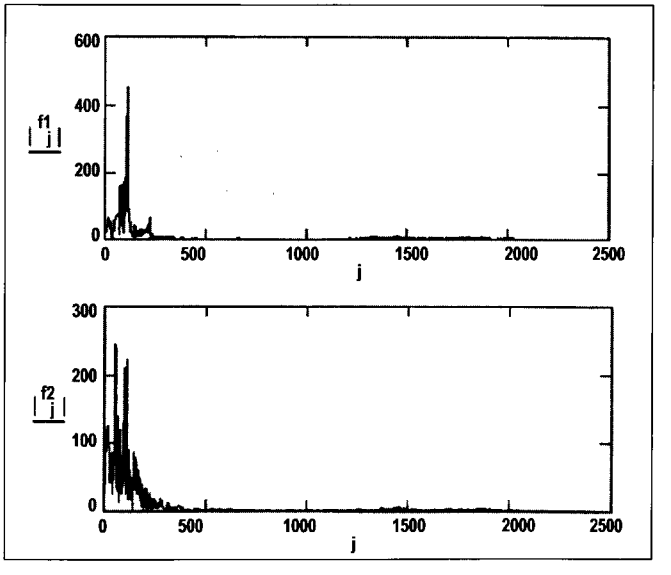


Figure 3. Frequency domain plot of "bean" and "peen" (f1 and f2, respectively)

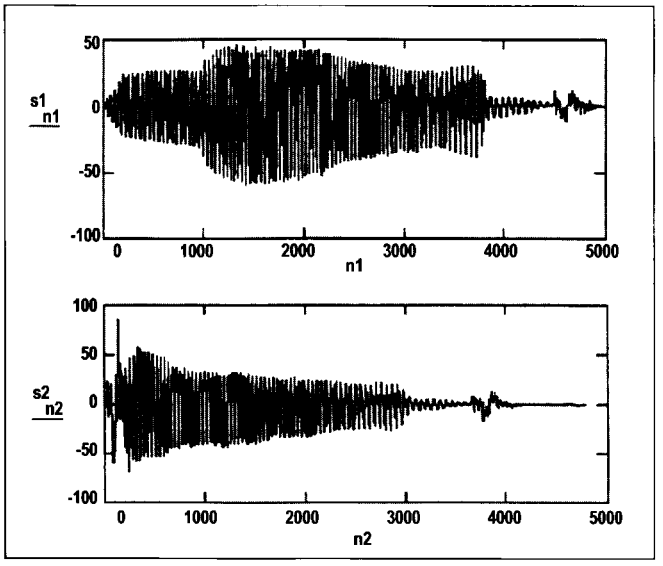


Figure 4. Time domain plot of "need" and "deed" (s1 and s2, respectively)

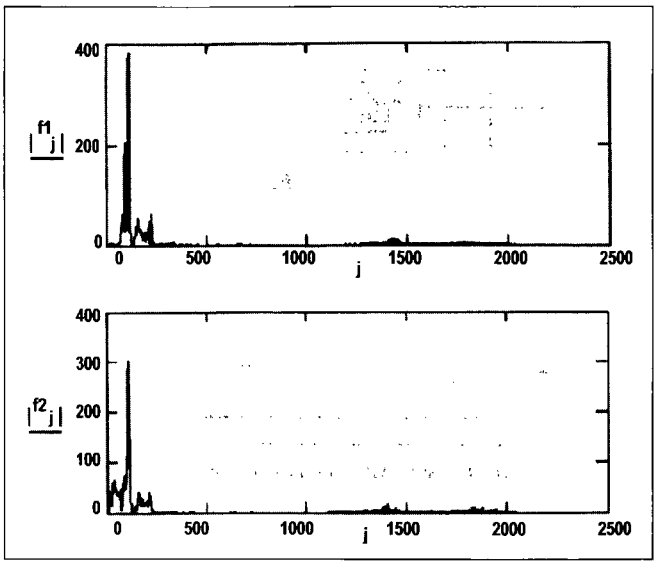


Figure 5. Frequency domain plot of "need" and "deed" (f1 and f2, respectively)

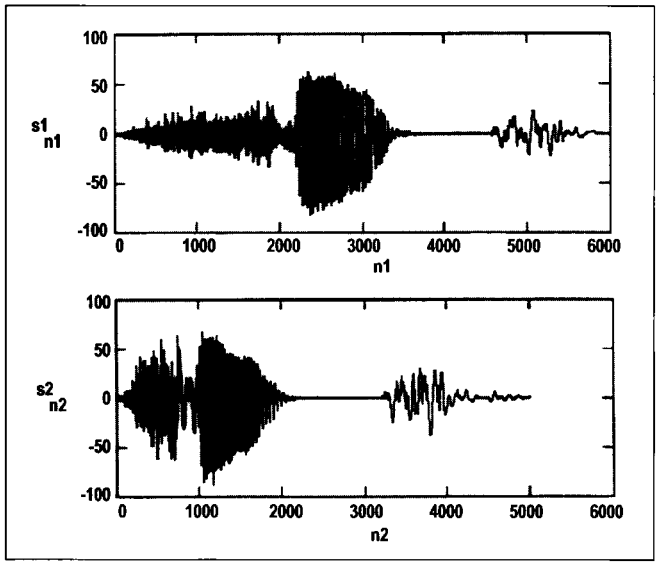


Figure 6. Time domain plot of "sheet" and "cheat" (s1 and s2, respectively)

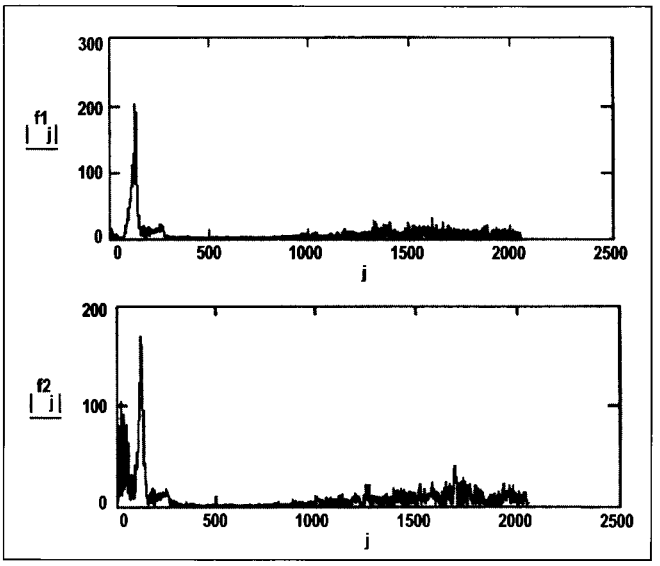


Figure 7. Frequency domain plot of "sheet" and "cheat" (f1 and f2, respectively)

the tentative endpoints set by the signal energy, we extend the endpoints to include the signal components that exhibit high zero-crossing rate. The endpoint detection algorithm flowchart is presented in Figure 14.

5. Feature Extraction

In speech recognition we seldom use the time domain speech signal directly without first extracting the speech features characterizing that word. One simple reason is that the speech signal, when spoken at different times, is very likely to have different durations and energy levels. If we try to use a time domain signal template to characterize a word and use a simple comparison method to make a decision, then the recognition rate could very likely be low, simply because of the mismatch of the speech durations and energy levels. In order to avoid these problems (among others), we first extract the speech features that characterize the speech and that are relatively independent of speech duration, energy levels and other factors. Another advantage is that it is more economical in terms of signal storage to use speech features as compared to the entire speech signal. In fact, speech signal compression, for economical signal storage and transmission, makes use of the same idea.

Algorithms used for feature extraction include Linear Predictive Coding (LPC) and Neural Network Analysis [5, 6, 7]. In this paper we elect to use the LPC approach for feature extraction. It extracts a set of LPC coefficients and a good characterization of the word. This method has been used for some time in speech recognition [8], although a neural network-based method could be used [9]. The following method of feature extraction is adopted from [5]. Sections of N consecutive speech samples (N = 160, corresponding to a 20-millisecond (ms) frame of signal and a sampling rate of 8,000 samples per second) are used as signal frames. Consecutive frames are spaced M samples apart (M = 80, corresponding to a 10-ms frame spacing, or a 10-ms frame overlap). Weighted eighth-order cepstral coefficients are extracted from each frame to produce an observation vector o. Observation vectors are extracted for the first 40 frames of speech and stored as speech feature vectors. A block diagram of the processing steps is given in Figure 15.

6. Word Classification

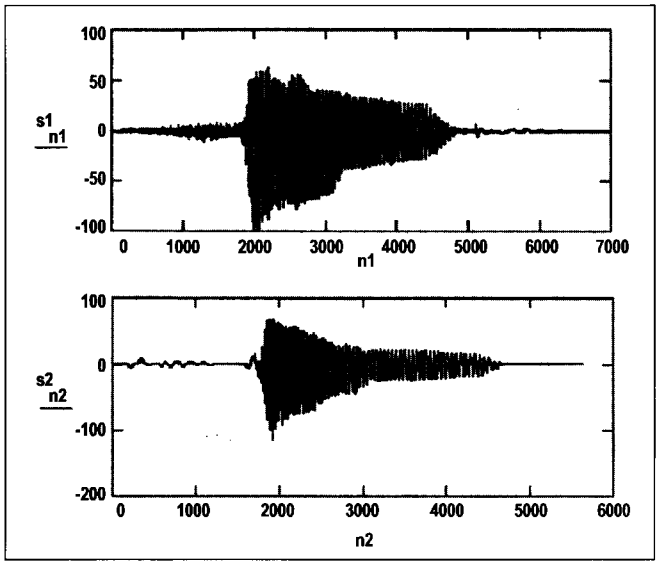


Figure 8. Time domain plot of "sing" and "thing" (s1 and s2, respectively)

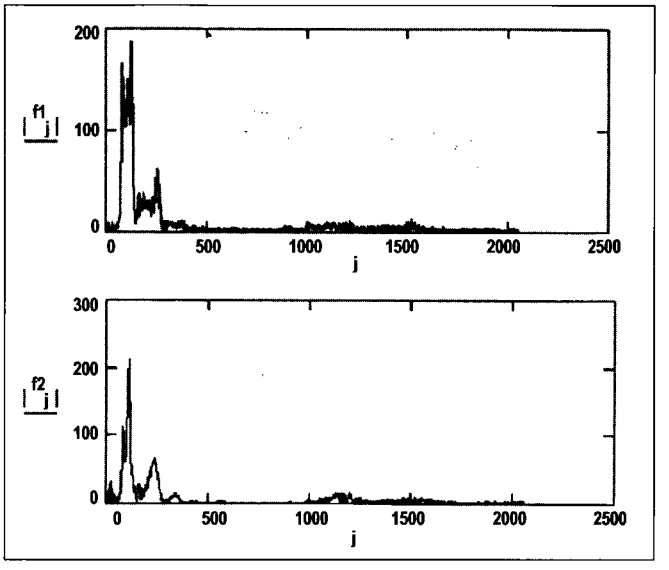


Figure 9. Frequency domain plot of "sing" and "thing" (f1 and f2, respectively)

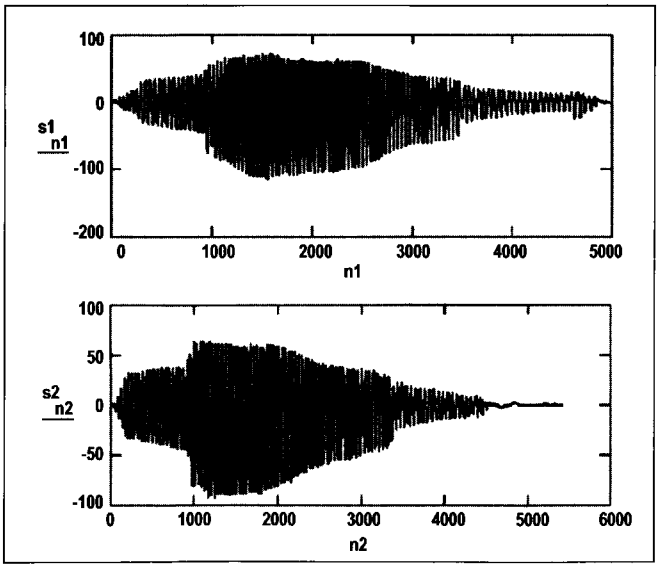


Figure 10. Time domain plot of "moon" and "noon" (s1 and s2, respectively)

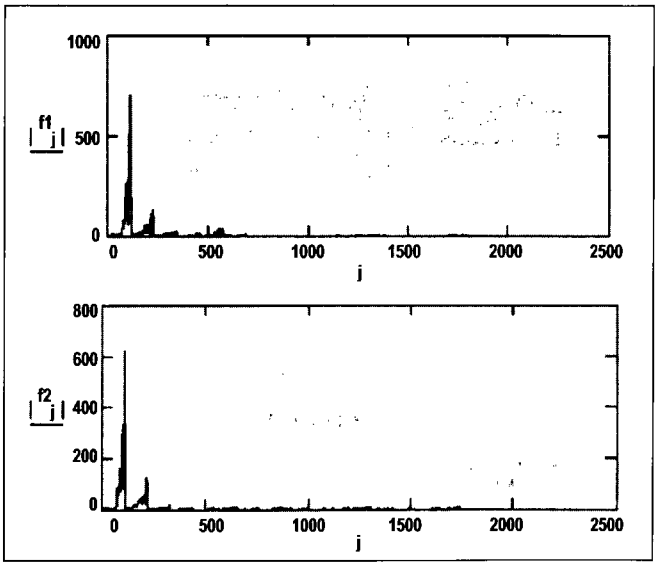


Figure 11. Frequency domain plot of "moon" and "noon" (f1 and f2, respectively)

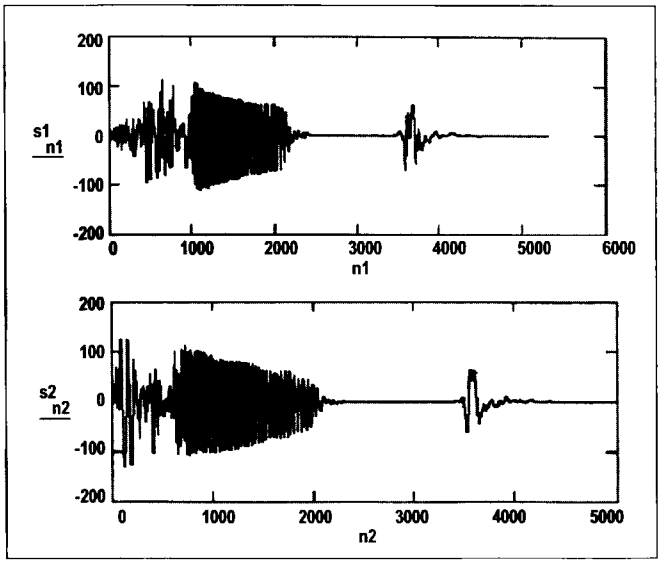


Figure 12. Time domain plot of "coop" and "poop" (s1 and s2, respectively)

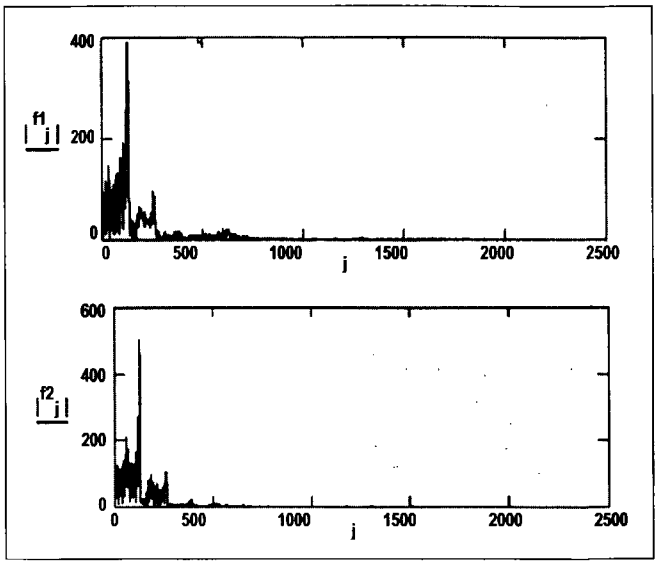


Figure 13. Frequency domain plot of "coop" and "poop" (f1 and f2, respectively)

The final processing performed by a speech recognition system is the word classification process, which is a special case of the more general pattern recognition process. Classifiers used for word and pattern recognition include Dynamic Time Warping (DTW), Hidden Markov Model (HMM), Backpropagation Neural Network (BNN), and K-Nearest Neighbor (KNN) algorithms [5, 7, 8, 9]. In [5], application of HMMs in speaker-independent isolated word recognition is compared with DTW-based methods. In this paper the performance of the BNN is compared to that of the KNN and the Closest Class Mean (CCM) classifiers in speaker-dependent isolated word recognition.

The word classification process classifies a word by making use of a generalized "distance measure." The classifier compares the input speech feature vector against the reference vectors to make a classification decision. For instance, a simple Euclidean distance measure can be used to compute the distances between the input feature vector and the reference vectors; one reference vector represents one word in the vocabulary. The word associated with the reference vector that has the smallest distance to the input vector is the recognized word. This approach is used in KNN and CCM classifiers.

The distance measure used in the BNN approach, however, is not defined in a simple manner. The knowledge of reference words is built into the processing unit weights in the training stage. The output of the BNN is directly mapped to the recognized word. In other words, the input feature vector

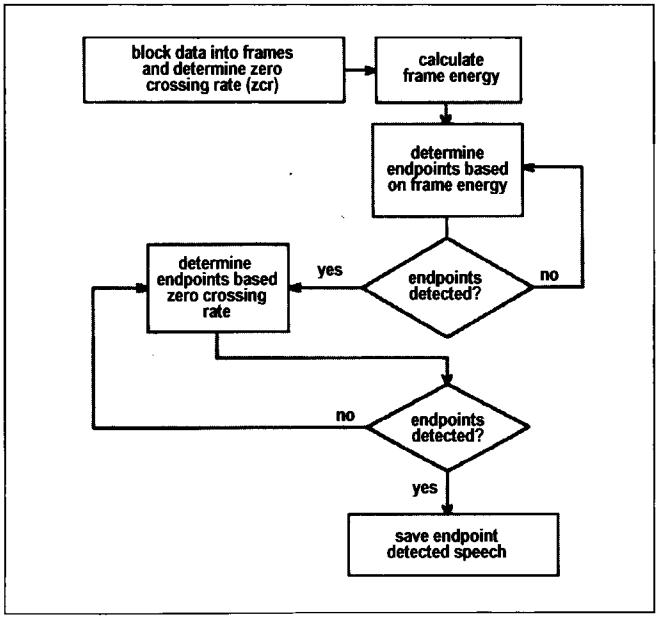


Figure 14. Endpoint detection algorithm flowchart

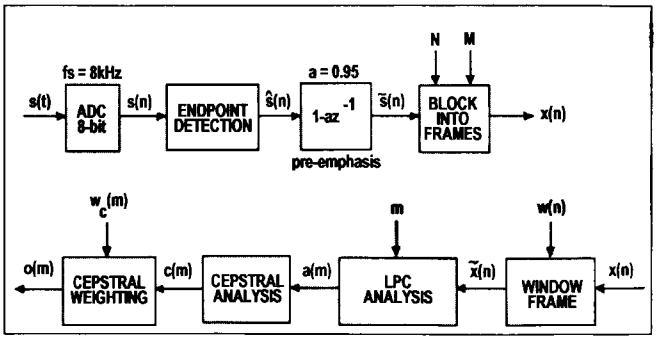


Figure 15. Feature extraction block diagram

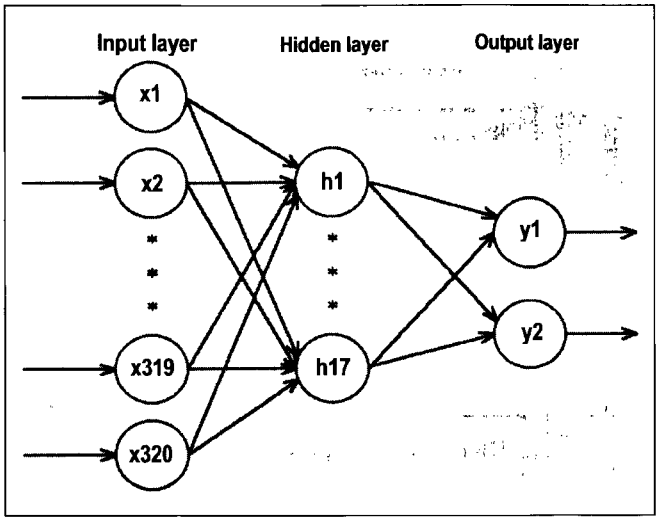


Figure 16. Backpropagation network architecture used for DRT word recognition

of an unknown word goes through a nonlinear transformation performed by the BNN, and its output points to the recognized word.

The BNN architecture, used in DRT word recognition, is a commonly-used three-layer, feed-forward network. In general, BNNs, having a greater number of hidden processing units, have a more powerful capability in word classification. The training time for a large number of processing units could be very long, however. Therefore, a balance between the two has to be made. [10] gave a guideline in selecting the minimum number of processing units. In practice, however, a larger number of units should be selected. In this study, the learning rates for networks having two, 11, 17, 23 and 29 hidden processing units were investigated. Among these, 17 hidden units appeared to be the best compromise between the system performance and the training time. Therefore, the BNN network architecture chosen consists of 320 inputs (40 frames of eighth-order cepstral coefficients), 17 hidden processing elements, and two output processing elements. The BNN architecture is shown in Figure 16.

7. Performance Results

For each DRT category, 200 utterances of each word in the category were stored for a total of 2,400 utterances (six categories times two words per category times 200 utterances). The recorded words were analyzed by the endpoint-detection algorithm to detect the speech endpoints [3]. Reference feature vectors were extracted for each word utterance. Zero-mean Gaussian white noise was added to the reference feature vectors to generate data for Signal-to-Noise Ratio (SNR) testing. For each DRT category, the reference vectors were divided into two groups of data. The classifiers were trained using one group and tested using the other. Then the group used for testing was used for training, and the group for training was used for testing. The trained classifiers

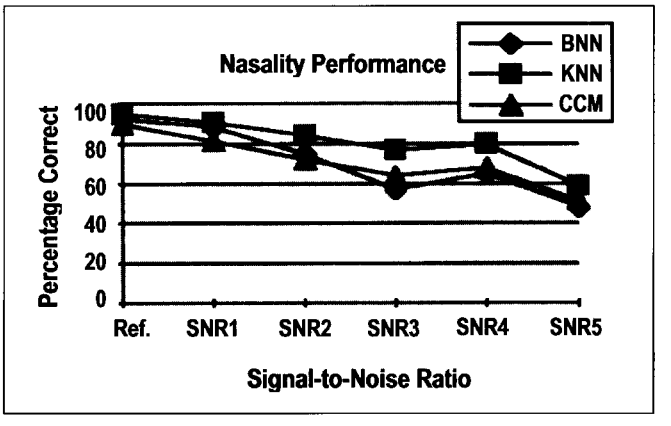


Figure 18. DRT word recognition performance in Nasality

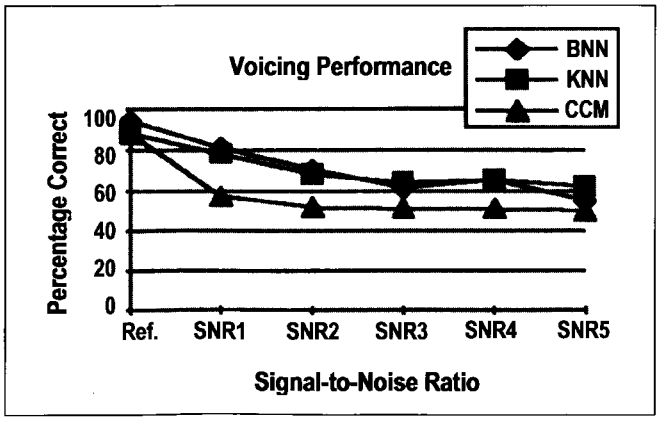


Figure 17. DRT word recognition performance in Voicing

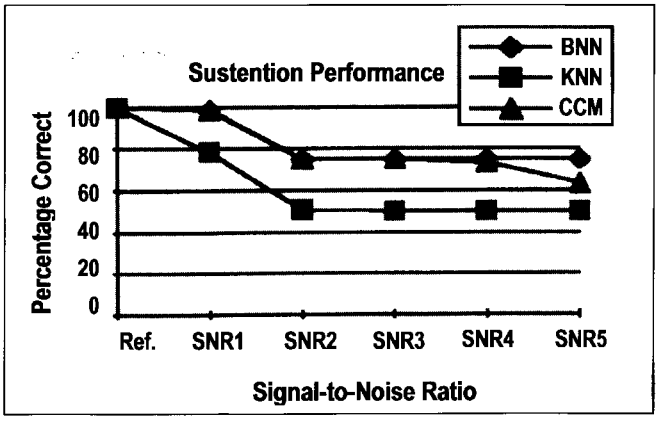


Figure 19. DRT word recognition performance in Sustention

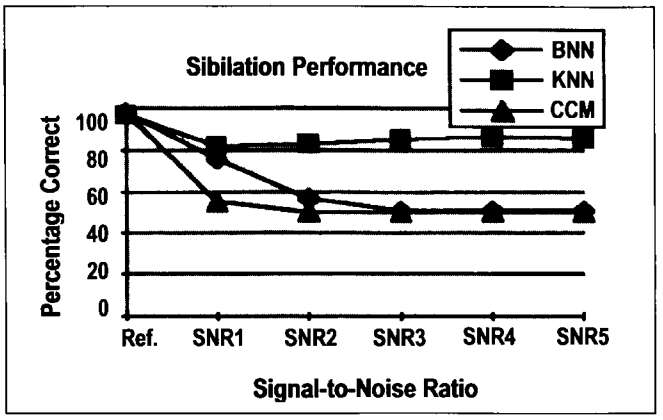


Figure 20. DRT word recognition performance in Sibilation

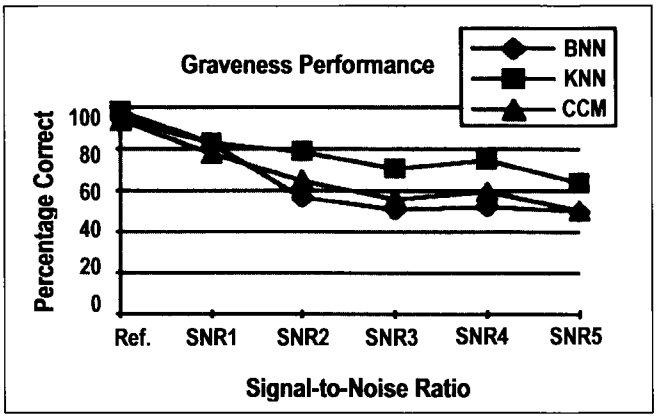


Figure 21. DRT word recognition performance in Graveness

were tested with noisy test vectors for several normalized SNR values. The average of the percentage correct was taken from both runs and the results are presented in Figures 17 through 22. The normalized SNR is defined as follows:

$$SNR = 10log\left(\frac{1}{1+\sigma_n^2 / \sigma_s^2}\right) dB$$

where σ_s^2 is the variance of the feature vector and σ_n^2 is the variance of the noise. The normalized SNR values used for each category of DRT are tabulated in Table 1.

For the reference SNR (noise-free) case, all three classifiers correctly classified 100% of the testing data not contained in the training examples for the DRT category of sustention. The BNN classifier performance was greater than 92% in all the DRT categories for the noise-free case. This performance exceeded that of the KNN classifier, which was greater than 83%, and the CCM classifier, which was greater than 85%. Both the KNN and CCM classifiers had the worst performance in the DRT compactness category.

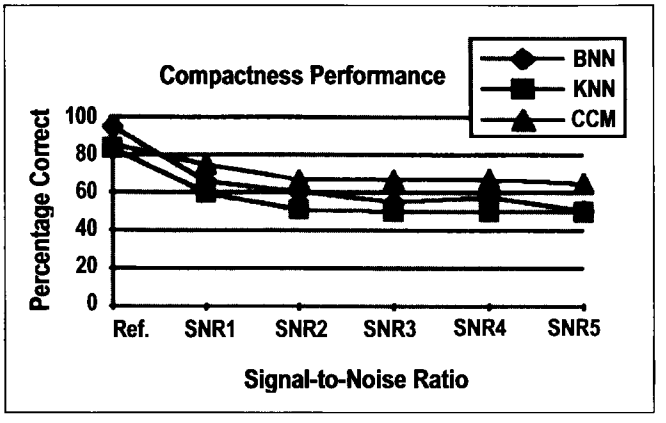


Figure 22. DRT word recognition performance in Compactness

In [7], a trained BNN achieved a performance of 90.4% in classification of sonar returns, and the KNN classifier achieved a performance of 82.7%. These results are consistent with the BNN and KNN classifier performance in the DRT categories of voicing and compactness for the reference SNR case. The SNR performance of the KNN classifier was greater than 82% in the sibilation category for all SNR cases. Similarly, the BNN performance for all SNR cases was greater than 75% in the category of

Table 1. Normalized SNR definition in Figures 17 through 22

	Ref. (dB)	SNR1 (dB)	SNR2 (dB)	SNR3 (dB)	SNR4 (dB)	SNR5 (dB)
Voicing	0	-5	-8	-10	-12	-14
Nasality	0	-4	-7	-9	-11	-13
Sustention	0	-13	-18	-21	-23	-25
Sibilation	0	-16	-21	-24	-27	-28
Graveness	0	-4	-7	-9	-11	-13
Compactness	0	-3	-5	-7	-9	-10

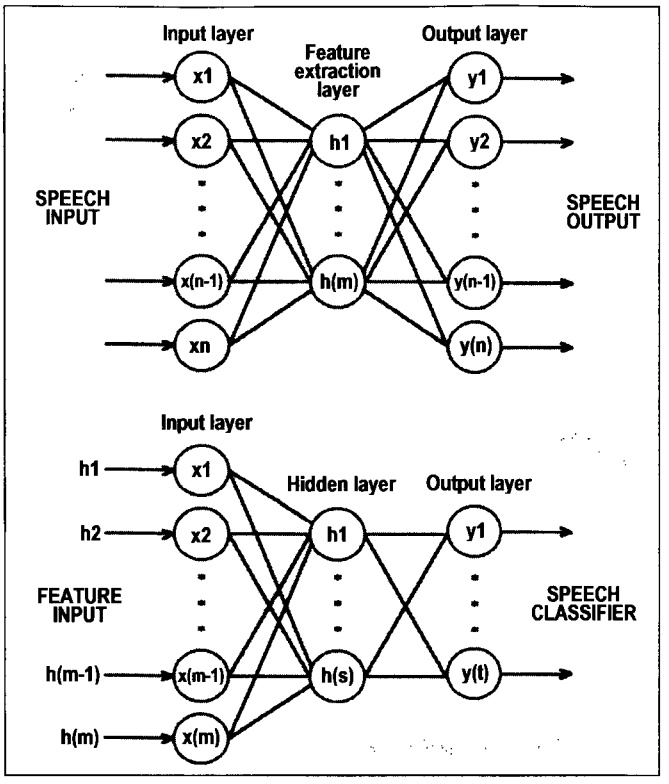


Figure 23. Possible neural network architecture for speech compression and classification

sustention, and the CCM performance was greater than 64% in the category of compactness.

8. Conclusion

The performance results indicate that for the task of speaker-dependent word recognition, the BNN classifier performance exceeds that of the KNN and CCM classifiers for the reference SNR case. Once noise is introduced into the system, however, the KNN classifier performs better than the others in three of the six DRT categories, while the BNN performs better in two of the six, and the CCM classifier in one of the six. Because the DRT tests six categories of speech attributes and is a good indicator of the overall performance of a speech communication system [1], a BNN classifier would be the best choice for a speaker-dependent word recognition application in a less noisy environment, while the KNN classifier might be a better choice in a more noisy environment.

Suggestions for Further Research

This research has shown the performance of the BNN, KNN and CCM classifiers in the task of speaker-dependent DRT word recognition. Subjects for future research include a comparison of the performance of these classifiers in the task of speaker-independent DRT word recognition, where the data would tend to be less well clustered. In this case, the BNN performance is expected to exceed that of the other two classifiers because of the distributed memory property inherent to highly parallel structures. Based on past research [6, 7, 9], an auto-associative neural network-based feature extractor is a viable alternative in a speech recognition system. A possible system is presented in Figure 23. The proposed network would not only perform speech classification, but could also perform speech compression at the same time. A possible application could be a "smart" telephone system that not only transmits speech and data, but listens to it as well.

References

[1] Voiers, W. D. 1979. "Intelligibility Testing at Dynastat: The Diagnostic Rhyme Test." *Dynastat Report*, December 1979, pp. 1-10.

[2] Voiers, W. D. 1983. "Evaluating Processed Speech Using the Diagnostic Rhyme Test." *Speech Technology Magazine*, Jan/Feb 1983, pp. 30-39.

[3] Rabiner, L. R. and Sambur, M. R. 1975. "An Algorithm for Determining the Endpoints of Isolated Utterances," *Bell Systems Technical Journal*, vol. 54, pp. 297-315.

[4] Dell, R. D., Proakis, J. G. and Hansen, J. H. L. 1993. *Discrete-Time Processing of Speech Signals*, Macmillan Publishing Company.

[5] Rabiner, L. R. 1989. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." *Proceedings of IEEE*, vol. 77, no. 2, pp. 276-278.

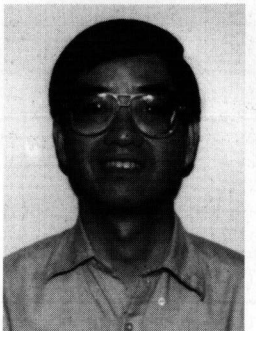
[6] Cottrell, G. W., Munro, P. W. and Zipser, D. 1987. "Learning Internal Representations from Gray Scale Images: An Example of Extensional Programming." *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, pp. 461-473.

[7] Gorman, R. P. and Sejnowski, T. J. 1988. "Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets." *Neural Networks*, vol. 1, pp. 75-89.

[8] Itakura, F. 1975. "Minimum Prediction Residual Principle Applied to Speech Recognition." *IEEE Transactions on ASSP*, vol. ASSP-23, no. 1, pp. 67-72.

[9] Elman, J. L. and Zipser, D. 1987. "Learning the Hidden Structure of Speech." *ICS Report 8701*, Feb. 1987, pp. 1-17.

[10] Mirchandini, G. and Cao, W. 1989. "On Hidden Nodes in Neural Nets." *IEEE Transactions on Circuits and Systems*, vol. 36, pp. 661-664.



DR. CHIT-SANG TSANG (Senior Member, IEEE) is a Professor of Electrical Engineering at California State University, Long Beach. He has published more than 40 papers in the areas of communications and digital signal processing. He received a PhD degree in 1982 from the University of Southern California, Los Angeles.