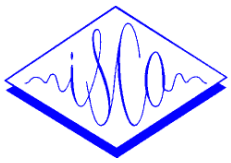

First International Conference on
Spoken Language Processing (ICSLP 90)
Kobe, Japan
November 18-22, 1990

A NEURAL NETWORK FOR SPEAKER-INDEPENDENT ISOLATED WORD RECOGNITION

Kouichi Yamaguchi, Kenji Sakamoto
Toshio Akabane, Yoshiji Fujimoto

Central Research Laboratories, SHARP Corporation
2613-1 Ichinomoto-cho, Tenri-shi, Nara 632, Japan

ABSTRACT

This paper presents a new, speaker-independent word recognition system based on three kinds of multilayer neural networks hierarchically arranged. The bottom neural networks act as identifiers of acoustic events and align time distortion. The middle neural networks output similarity measures for the input words. The top neural network is a classifier and outputs the recognition candidates. Speaker-independent recognition experiments using 28 isolated Japanese words were carried out using data uttered by 150 speakers (100 speakers for training and 50 speakers for testing). As a result, we obtained a 97.1% recognition accuracy and a 1.0% error rate.

1. INTRODUCTION

It has been recently proved in [1] that any continuous mapping can be approximately realized by multilayer neural networks with at least one hidden layer and whose output functions are sigmoid functions. In most speech recognition systems, speech is dealt with as a time sequence of feature parameters. However, being a static model, a multilayer neural network is not capable of modeling signals with an inherent time variability of speech.

To deal with this problem, several ideas have been proposed. Sakoe [2] proposed the Dynamic Programming Neural Network (DNN) which is based on the integration of DP and multilayer neural networks. Levin [3] proposed the Hidden Control Neural Network (HCNN) which combines the nonlinear prediction of a conventional neural network with hidden Markov modeling. Iso [4] proposed the Neural Prediction Model (NPM) which uses a sequence of multilayer networks as a separate nonlinear predictor for each class.

In this paper we propose a new neural network architecture. The proposed architecture has three kinds of multilayer neural networks. The bottom neural network acts as a neural identifier of acoustic events and aligns time distortion. The middle neural network outputs similarity measure for the input word. The top neural network is a nonlinear classifier and outputs the recognition candidates.

Section 2 introduces the proposed architecture. Section 3 presents the experiment results compared with the preliminary architecture which adopts precise manual time alignment by transcription. Section 4 contains a summary of our work and describes some directions for future research and development.

2. PROPOSED NEURAL NETWORK ARCHITECTURE

2.1 Outline of the Architecture

The neural network considered here is a multilayer feed-forward type network [6] with connections between adjoining layers only. Fig.1 shows the proposed neural network architecture. The system has three kinds of neural networks. Each neural network is a three-layer perceptron and the whole system structure is arranged hierarchically.

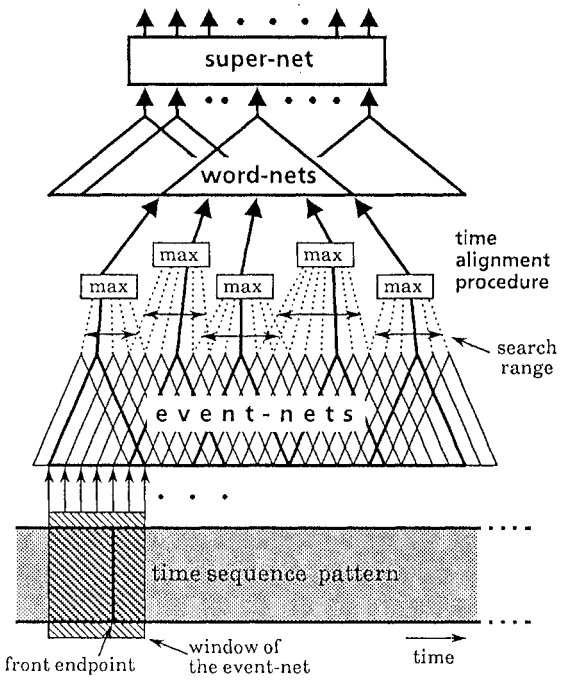

Fig. 1. Proposed Neural Network Architecture.

The first neural network, called the "event-net," receives the input time sequence. It has only one output unit and it outputs the similarity between the given input sequence and the part of phoneme series which the event-net has been trained with. The event-net can be thought of as an identifier of the phoneme series or acoustic events. The second neural network, called the "word-net," usually chooses the highest output of the event-net among the event-nets located in its search range, and outputs the similarity measure between the

given input word and the conceptual reference stored in the form of weighting coefficients. The third neural network, called the "super-net," is a nonlinear classifier and receives outputs from the word-nets and outputs the recognition candidates. Finally, the recognition result is chosen by maximum selection and rejection decision rules among recognition candidates from the super-net. The sigmoid function was chosen as the nonlinear output function in the hidden and the output layers of each neural network.

The performance of the recognition system generally depends on the accuracy of the endpoint detection. Especially in noisy environments, the accuracy of the back endpoint decision may decrease drastically and the recognition performance goes down. Therefore we propose an architecture which does not need the back endpoint detection and can be evolved into word spotting.

2.2 Preprocessing

Fig.2 shows a block diagram of our whole recognition system. At the lowest level, 7-channel

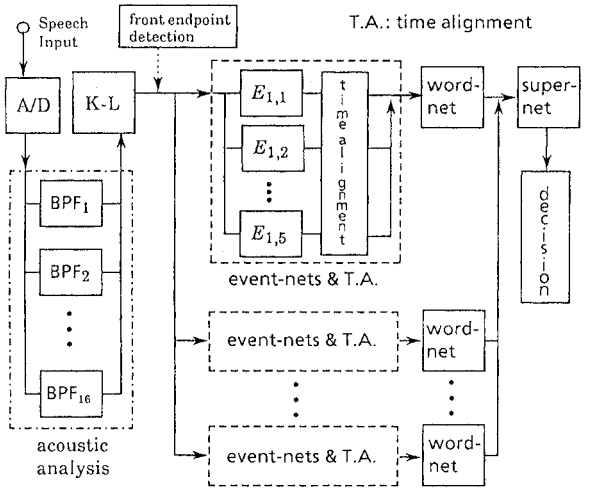

Fig. 2. Proposed Recognition System.

compressed parameters serve as inputs to the event-net. Input speech, sampled at 8 kHz, is Hamming windowed and fed into the 16-channel melscale bandpass filters which cover a 0.3 to 3.4 kHz frequency band. 16 power coefficients are computed from the output waveforms of the filters every 10 msec. In order to save computing costs, these coefficients are compressed by K-L transformation, that is, 2 frames of the 16-channel coefficients are compressed into 7 coefficients.

Front endpoints have already been marked by observing spectrograms. The architecture uses this front endpoint information.

2.3 Event-Net

A neural network is not capable of treating time sequence patterns if they contain globally blurred temporal shifts as mentioned above. But it is able to treat time sequence patterns containing temporally small shifts like phoneme recognition using Time-Delay Neural Networks (TDNN) introduced by Waibel [5]. The input layer has 49 units, the hidden layer has 15 units and the output layer has 1 unit. There are 7 compressed coefficients in each frame. The input layer looks at a 7-frame window. Thus it has 49 units. The choice of a 7-frame (equivalent to 140 msec) window, was motivated by the intuition that the event-net should learn to discover a series of two or three phonemes, or one syllable. The input layer is partially interconnected to the hidden layer as shown in Fig.3.

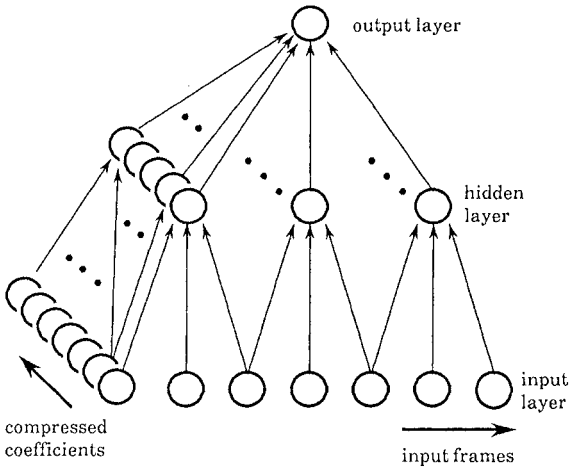

Fig. 3. Event-Net Structure.

The hidden layer is fully interconnected to the output layer.

The number of event-nets can be freely set, for example, according to the number of syllables in the word. We let each word reference have 5 event-nets in our present simulation system. These 5 event-nets are enough to cover almost the whole input word utterance. But when the input utterance is too long, the event-nets might not cover the whole utterances. The event-net is trained to look for its own acoustic events. Therefore it fires, i.e., outputs a high value near 1.0 when its window covers its own acoustic events and it does not fire, i.e., outputs a low value near zero when its window does not cover its own acoustic events.

Training samples have already been transcribed by observing their spectrograms. The j-th event-net representing the i-th category word is named E_{ij}, where $j = 1,2,\cdots,5$. In learning the event-net E_{ij}, the training samples belonging to the i-th category should make E_{ij} fire. These are called the "regular" training samples. They are taken from the reference position determined by hand transcription. That is to say, the regular training samples are precisely time-aligned. The training samples which should not make the event-net fire, that is, those not belonging to the i-th category, are called the "counter-" training samples. They are taken from the middle of the search range. The search range covers the frames of the input sample successively passed over, from left to right, by the window of the event-net. Because they are selected successively from the front endpoint according to the average interval duration between the E_{ij} and E_{ij+1}, the position of the countertraining samples will be located in the middle of the search range of each event-net in the recognition.

The back propagation learning procedure [6] is applied to the event-nets. Among the countertraining

samples, there are some which are very similar to the regular training samples. Their output errors are very large. Therefore, we try to neglect these countertraining samples. For example, event-net E_{i5} of the i-th category /ichi/ and the other event-net E_{j5} of the j-th category /hachi/ both represent the tail of the phoneme sequence /chi/. At learning E_{i5}, the countertraining samples of the word /hachi/ often output a very high value more than θ_{neg}. We ignore the countertraining samples over θ_{neg}. We set the threshold θ_{neg} at 0.8 in the present system.

2.4 A Time Alignment Procedure

Fig.1 also shows an example of the left-to-right time alignment procedure. E_{ij} passes over the input sample frame by frame and E_{ij} in the l-th position is renamed E_{ijl} ($l = 1,2,\cdots,L_{ij}$ where L_{ij} is the search range size). The weighting coefficients of E_{ijl} ($l = 1,2,\cdots,L_{ij}$) are equal to each other. E_{ijl} is located in the next frame on the right side of E_{ijl-1}. Let the output value of E_{ijl} be C_{ijl}. The output value C_{ij} of the event-net E_{ij} is basically defined as the maximum value of C_{ijl}. However there are exceptions as described below.

- Rule 1:If all C_{ijl} ($l = 1,2,\cdots,L_{ij}$) are almost the same, then the system selects the middle of the search range.
- Rule 2:If all C_{ijl} ($l = 1,2,\cdots,L_{ij}$) are very low, the system expands the search range. ($L_{ij} \leftarrow [(1+\alpha)\times L_{ij}]$)

Rule 1 avoids the needless time alignment against the countertraining samples. Rule 2 saves the regular training samples spoken very slowly. Here, we set the constant α at 0.3.

Search range decisions are made as follows: First, the search range of E_{i1} is decided to within a few frames around the front endpoint. In this example shown in Fig.1, two frames around the front endpoint are chosen. Then L_{i1} is set at 5. Next, the search range of E_{ij} ($j>1$) is decided using the mean value m_{ij} and the standard deviation σ_{ij} of the interval duration between E_{ij-1} and E_{ij} calculated from the training data set. Let E_{ij-1} be already decided to choose $E_{ij-1,n}$ ($1 \leqq n \leqq L_{ij-1}$). In Fig.1, if j is 2, then $E_{i1,2}$ is chosen, which is indicated by the bold triangle and line. The other, unchosen event-nets are indicated by fine triangles and dotted lines. The middle of the search range of E_{ij} is defined as the m_{ij}-th next frame on the right side of the frame of $E_{ij-1,n}$. The size of the search range of E_{ij} is set at $2k\sigma_{ij}$. But if $m_{ij} \leqq k\sigma_{ij}$, then the search range begins at the next frame on the right side of the frame of $E_{ij-1,n}$, that is, the frame of $E_{ij-1,n+1}$. Here, we set the constant k at 2.9.

2.5 Word-Net

The word-net W_i is also a three-layer perceptron, where i is the category number ($i = 1,2,\cdots,N$ where N is the vocabulary size). The input layer has 5 units and is fully interconnected to a layer of 5 hidden units. The hidden layer is also fully interconnected to a layer of one output unit. We also adopt the back propagation learning procedure here. The word-net W_i has been trained to fire when training samples belonging to the i-th category, called the "regular" training samples, are input and not to fire when training samples not belonging to the i-th category, called the "counter-" training samples, are input.

When making training samples of word-nets, the time alignment procedure of event-nets adopts the left-to-right time alignment procedure when the regular training samples are passed over and adopts the middle position of the search range when the countertraining samples are passed over. The word-net W_i absorbs the fluctuation of C_{i1}, C_{i2}, $\cdots$, C_{i5}. Consequently, the word-net W_i outputs the similarity measure between the given input word and the i-th conceptual reference stored in the form of weighting coefficients.

2.6 Super-Net

The input layer has N units and is fully interconnected to a layer of $N+1$ hidden units. The hidden layer is also fully interconnected to a layer of $N+1$ output units. In our simulation system, N is set to 28. The first to the N-th output units correspond to the recognition category number and the $(N+1)$-th output unit corresponds to the rejection flag.

The back propagation learning procedure is also adopted here. When the input sample belongs to the I-th recognition category, the training pattern is set as

$s_i=1$ if $i=I$
$s_i=0$ if $i \neq I$, $1 \leqq i \leqq N+1$,

where s_i denotes the desired output value of the i-th output unit. The input samples come from word-nets. But a word-net with a regular training sample occasionally does not fire. Then, all the outputs of the word-nets become low. In this special case, the training pattern is set as

$s_{N+1}=1$
$s_i=0$ $1 \leqq i \leqq N$.

This training schedule is useful in improving the rejection ability.

2.7 Decision Algorithm

The super-net absorbs the fluctuation, confused outputs, of word-nets. But it sometimes leaves errors, confused output values. In order to reject these errors and improve recognition accuracy, the system decides the recognition result using rejection thresholds. Let S_k be the highest output value of the super-net and S_l be the second highest output value of the super-net ($1 \leqq k,l \leqq N+1$). If $S_k > \theta_{rej}$ and $S_k - S_l > \theta_{next}$, then the recognition result is the k-th category. Otherwise the system rejects that input sample. θ_{rej} and θ_{next} are rejection thresholds. We set 0.5 and 0.1 respectively in the present system.

3. RECOGNITION EXPERIMENTS

3.1 Experiment Conditions

We now turn to an experimental evaluation of the proposed architecture described in the previous sections. Speaker-independent, 28 isolated Japanese word recognition experiments were carried out using data uttered by 150 speakers. This data is derived from the PCM version of JEIDA JAPANESE COMMON SPEECH DATA CORPUS. Table 1 shows the vocabulary lists of the data. They are composed of 10 digits and typical control words. Each speaker uttered a word four times. We used the third utterance of each word. 7-channel compressed parameters were calculated in the same way as described in section 2.2. The samples were divided into two sets. 100 speakers' samples, that is, 2800 samples, were used for training and 50 speakers' samples, that is, 1400 samples, were used for testing. All front endpoints were marked by observing the spectrograms.

Table 1. Recognition Vocabulary Lists Composed of 10 Digits and Typical Control Words.

zero, ichi, ni, saN, yoN, go, roku, nana, hachi, kyuu, ue, shita, migi, hidari, dai, shou, kyou, jaku, mae, ushiro, zeNshiN, koutai, tsukeru, kesu, akeru, shimeru, hai, iie

3.2 Preliminary Experiments

Before evaluating the proposed architecture, we carried out preliminary experiments in order to investigate the effectiveness of the event-nets and the left-to-right time alignment procedure. The architecture used here, called the "preliminary" architecture, has the same structure as the proposed architecture. Each neural network also adopts the back propagation learning procedure. But there are some differences between them which are described below.

(1) The regular training samples of word-nets are precisely time-aligned by hand transcription. That is to say, they do not adopt the left-to-right time alignment procedure. Since the regular samples used here are not blurred by temporal shifts, we can examine the individual performance of event-nets.
(2) The countertraining samples of event-nets and word-nets are taken at equal intervals from the front endpoints to the back endpoints of the input samples.
(3) In recognition, the regular test samples are also precisely time-aligned and the countertest samples are also taken at equal intervals from their front endpoints to their back endpoints. Therefore there is no mistake in the time alignment.

3.3 Results

The recognition results of two architectures are shown in Table 2. Method I is the preliminary architecture and adopts precise manual time alignment by transcription. Method II is the proposed architecture and adopts the left-to-right automatic time alignment. 97.1% recognition accuracy was obtained with the proposed architecture. This performance is comparable to that of the preliminary architecture. It can be seen that the event-nets and the time alignment procedure work successfully. An event-net is invariant against temporally small shifts.

There are several similar words in the recognition vocabulary. For example, /ni/ and /migi/ (/ni/ is sometimes pronounced as /nii/ and /g/ in /migi/ is often nasalized.), /dai/ and /hai/, /shou/ and /kyou/. Many of the errors or reject samples come from these similar words. The former is almost a rhyme for the latter. Almost all 5 event-nets of the former sometimes fire when the latter samples are input. The super-net solves the "partial matching problem," which often occurs in an endpoint free recognition system. For example, the first half of /koutai/ is very similar to /go/. Therefore both the /koutai/ and /go/ word-nets fire when a sample of /koutai/ is input. On the other hand, the /go/ word-net fires alone when a sample of /go/ is input. The super-net correctly judges from the above-mentioned firing patterns.

Table 2. Recognition Results for the Open Data Set.
The time alignment of methods I and II is done by transcription and the proposed procedure, respectively.

Method	Time Alignment	Correct	Reject	Error
I	Manual	97.8%	1.7%	0.5%
II	Automatic	97.1%	1.9%	1.0%

4. SUMMARY

This paper introduces a new speech recognition model using multilayer neural networks arranged hierarchically. It realizes time alignment operation. The left-to-right time alignment procedure is very simple, thus it reduces computation costs. Consequently, the architecture can be implemented on a single Digital Signal Processor (DSP).
Features of the proposed architecture are:

(1) Since event-nets are trained with countertraining samples in addition to regular training samples, they can identify the acoustic events and are invariant against temporally small shifts.
(2) Since an event-net fires fairly exactly when its window covers its own acoustic events, it can be used for left-to-right time alignment, which reduces computation costs.
(3) The super-net solves the partial matching problem based on the various fired output patterns of word-nets.

We are going to evolve the proposed architecture so as not to need the front endpoint detection, that is, to realize word spotting and develop a robust recognition technique under noisy environments. We are also going to implement the proposed architecture on a DSP.

ACKNOWLEDGEMENTS

Special thanks are due to Dr. Funahashi in Toyohashi University of Technology for numerous comments and helpful advice. Useful discussions with S. Nakamura and A. Kito in our department are greatly appreciated.

REFERENCES

[1] K. Funahashi, "On the Approximate Realization of Continuous Mapping by Neural Networks", *Neural Networks*, 2, pp.183-192, 1989.
[2] H. Sakoe, R. Isotani, K. Yoshida, K. Iso and T. Watanabe, "Speaker-Independent Word Recognition Using Dynamic Programming Neural Networks", *Proc. ICASSP-89*, pp.29-32, Glasgow, 1989.
[3] E. Levin, "Word Recognition Using Hidden Control Neural Architecture", *Proc. ICASSP-90*, pp.433-436, Albuquerque, 1990.
[4] K. Iso and T. Watanabe, "Speaker-Independent Word Recognition Using a Neural Prediction Model", *Proc. ICASSP-90*, pp.441-444, Albuquerque, 1990.
[5] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. Lang, "Phoneme Recognition Using Time-Delay Neural Networks", *Proc. ICASSP-88*, pp.107-110, New York, 1988.
[6] D.E. Rumelhart and J.L. McClelland, "Parallel Distributed Processing", MIT Press, 1986.