Training Method and Speaker Verification Measures for Recurrent Neural Network based Speaker Verification System

Tae-Hyung Kim* ***Regular Member***

ABSTRACT

This paper presents a training method for neural networks and the employment of MSE (mean scare error) values as the basis of a decision regarding the identity claim of a speaker in a recurrent neural networks based speaker verification system. Recurrent neural networks (RNNs) are employed to capture temporally dynamic characteristics of speech signal. In the process of supervised learning for RNNs, target outputs are automatically generated and the generated target outputs are made to represent the temporal variation of input speech sounds. To increase the capability of discriminating between the true speaker and an impostor, a discriminative training method for RNNs is presented. This paper shows the use and the effectiveness of the MSE value, which is obtained from the Euclidean distance between the target outputs and the outputs of networks for test speech sounds of a speaker, as the basis of speaker verification. In terms of equal error rates, results of experiments, which have been performed using the Korean speech database, show that the proposed speaker verification system exhibits better performance than a conventional hidden Markov model based speaker verification system.

Key Words : Speaker verification; RNN; Neural networks learning; Discriminative training; HMM.

Ⅰ. Introduction

The goal of speaker verification (SV) is to decide whether a given speech utterance has been pronounced by a claimed client or by an imposter. Automatic SV can be widely used in security and forensic applications. Based on the text to be spoken, applications of SV can be roughly grouped into text-dependent (TD) and text-independent (TI) cases[1]. In the TD case, it is required for the speaker to produce speech for the same text in both training and testing for the SV system, and the machine for SV knows the lexical content (keyword) of the utterance used for verification. The focus of this paper is on the TD speaker verification (TDSV) system using fixed-text.

TDSV involves detection of the valid keyword and extraction of the speaker-specific information from the input speech signal. In order to extract acoustic features for speaker recognition, current systems often use acoustic parameters that have been developed for the use in speech recognition. LPC (linear predictive coding) parameters (or LPC cepstra), which have fallen out of favor in automatic speech recognition because of their strong dependence on individual speaker characteristics, tend to be preferred in speaker recognition for this very reason[2].

The state-of-the-art TDSV models are based on a hidden Markov model (HMM), Gaussian mixture model (GMM), dynamic time warping (DTW), and Neural Networks (NNs), etc[1-7]. DTW based approach is simpler and requires relatively little computational resources during the enrollment phase of SV system. It has been the basis of several commercial products[2]. HMM based approaches have generally been found to be more accurate than the simpler DTW[3,6]. HMM based SV systems create a generative model for the utterance of each client and this generative

* 국방과학연구소(thyunkim@pusan.ac.kr)
논문번호 : KICS2008-11-501, 접수일자 : 2008년 11월 12일, 최종논문접수일자 : 2009년 2월 11일

model is prone to overfitting. In other words, for TDSV of good performance, the HMM based SV system requires huge amounts of training data in the enrollment phase. In most cases obtaining abundant utterances of each user is restrained because of customer convenience. NNs can represent any distribution of inputs without complicated modeling methods[8-9], and have been frequently used in classifying speech sounds into phonemes because they have a good ability for classification. NNs based TDSV systems have been found in [7,10-11]. In [10-11], vowels which are good to distinguish speakers are extracted and used for SV. Above mentioned NNs based methods need additional preprocessing (e.g., speech segmentation and phoneme recognition) for TDSV. Therefore, in some cases, a hybrid HMM-MLP SV algorithm has been used in TDSV, where HMM is used for speech segmentation and MLP networks use the segmented speech utterance for SV[12-13]. In [12], HMM is used for generating the MLP networks' inputs and target outputs that are needed in training for MLP networks; static characteristics of short-time intervals of a speech utterance are only considered and time-varying characteristics of the speech sounds are not considered.

The speech is basically nonstationary for long-time intervals and the consideration of the dynamics changes between speech frames (short-time intervals of a speech utterance) improves the SV performance. Therefore, a segmental HMM has been used for representing segments of features and incorporating the concept of trajectories to describe the time-varying characteristics of different speech sounds[14]. A recurrent neural network (RNN) can be seen as a nonlinear dynamic system, which may express both the static and dynamic features of the signal at hand[9]. In additions, a RNN may express speech dynamics and duration just like a segmental HMM, and may capture individual differences in nonstationary speech segments. As a result, a recurrent time delay NN, which is a form of RNN, has been used for TDSV[15]. But NNs of [15] have the structure that can accept only the isolated words utterance. It is difficult for NNs of [15] to accept the connected words utterance and the long-time-interval utterance at hand. That is, NNs of [15] need speech segmentation by HMM.

In this paper, a RNN based SV system is proposed. The proposed RNN based SV system accepts the connected words utterance and does not need additional preprocessing such as speech segmentation and phoneme recognition by the HMM module. Then, unlike hybrid HMM-MLP systems, the configuration of the proposed system is not so complicated. In addition, in the process of supervised training for RNN, the target outputs are automatically generated and the generated target outputs are made to represent the temporal variation of input speech sounds. To increase the capability of discriminating between the true speaker (customer) and an imposter, a discriminative training method between the true speaker and cohort speakers, whose voice characteristics are close to the true speaker's voice characteristics and that are therefore representative of the population near the true speaker, is presented.

In the verification phase, a SV system calculates a matching score between some speaker model and a speech utterance. The resulting score is compared to a given threshold, based on that the test speaker is accepted or rejected. Development of the most pertinent method for calculating the matching score will lead to the good performance of a SV system, and the estimation of the optimal threshold is often critical for a good performance of the system[16]. A threshold is empirically determined so a trade off between false alarms (false acceptances) and miss detections (false rejections) is obtained. As a convenience for system comparison, the performance of SV systems is often measured in terms of equal error rates (EER), corresponding to the decision threshold in which the false rejection rate is equal to the false acceptance rate. EER is often approximated as half of the sum of the two error rates. Outputs of NNs trained by using a MSE (mean squared error) criterion approximate posterior

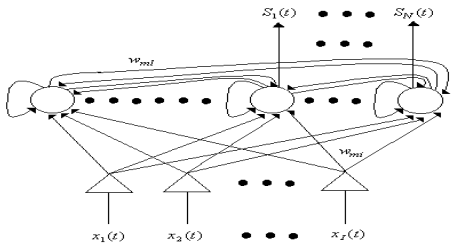

Fig. 1 The structure of a recurrent neural network

class probabilities[8]. Therefore, output values obtained from outputs of NNs may be used as the matching score. But, because NNs are trained by MSE criterion, using a MSE value as the matching score can show better performance than using other values in the RNN based SV system. This paper presents the use and effectiveness of MSE values as the matching score.

We have performed SV experiments using the Korean 4-connected-digits speech database. In terms of EER, the experimental results show that our NNs based SV system exhibits better performance than the conventional HMM based SV system in circumstances where the number of training data is few. In addition, we demonstrate the RNN's ability for capturing temporally dynamic characteristics of input speech signal. this done through the comparison of experimental results for MLP (static NNs) or RNN (dynamic NNs) based SV systems.

Ⅱ. Recurrent neural networks

MLP networks or static NNs only have the static mapping capability, i.e. output is a function of current inputs only. The consideration of the dynamics changes between speech frames improves the SV performance. In NN literature, NNs with one or more feedback loops are referred to as recurrent networks[9]. A RNN responds temporally to an externally applied input signal. The application of feedback enables recurrent networks to acquire state representations, which make them suitable devices for speech processing[9]. The role of the feedback delay units is to provide the network with dynamic memory, so as to encode the information contained in the sequence of phonemes. We will build a RNN based TDSV system that encodes the dynamics changes between speech frames contained in the connected-digits speech. Fig. 1 shows the structure of an RNN used in this paper, where the RNN has M neuron nodes, receives the I-dimensional input vector $\mathbf{X}(t)=[x_1(t),x_2(t),...,x_I(t)]$, and emits the N-dimensional output vector $\mathbf{Y}(t)=[s_1(t),s_2(t),...,s_N(t)]$. In addition, $M>N$, and the output of m-th neuron node is

$$s_m(t)=1/(1+\exp^{-\alpha\, net_m(t)}),\ m=1,2,\cdots,M \quad (1)$$

In Eq. 1, α is a constant and

$$net_m(t)=\sum_{l=1}^{M} w_{ml}s_l(t-1)+\sum_{i=1}^{I} w_{mi}x_i(t)+bias_m \quad (2)$$

where w_{ml} is the synaptic weight connecting l-th neuron node to m-th neuron node, w_{mi} is the synaptic weight connecting i-th input node to m-th neuron node, and $bias_m$ is the bias applied to m-th neuron node.

For training RNNs, the RTRL (real-time recurrent learning) algorithm is used[17]. The RTRL algorithm adjusts the synaptic weights of a fully connected recurrent network in real time, that is, while the network continues to perform its signal processing function. Through using the RTRL algorithm, the proposed RNN based SV system can continue to adapt to the gradual change of each individual speaker's voice characteristics, while the system continues to perform SV functions for a long time.

Ⅲ. Neural networks based TDSV

In this section, we describe the basic structure of a proposed TDSV system using NNs. In addition, the key methods used in the enrollment and the verification phases are represented.

3.1 Basic structure

The components of the proposed RNNs based TDSV system are shown in Fig. 2. In the SV

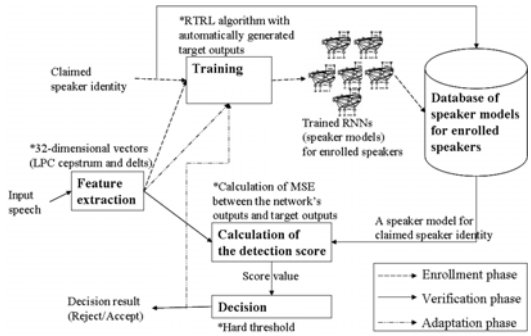

Fig. 2 RNNs based speaker verification system

system of Fig. 2, RNNs, which compose the module for speaker models, can be replaced with MLP networks, if the BP (backpropagation) algorithm is used as the learning algorithm and the adaptation phase is removed. Feature extraction transforms the raw signal into a sequence of 32-dimensional feature vectors, which consist of LPC-cepstrum (LPCC) coefficients and their deltas.

In the enrollment phase, a speaker model is created by training an RNN (or an MLP network) through the learning process of this paper. If the module for speaker models is composed of MLP networks, the BP algorithm is used for training NNs. And if the model for speaker models is composed of RNNs, the RTRL algorithm is used for training NNs. These supervised learning algorithms[9], such as BP and RTRL, need target outputs. The formation of target outputs, which are automatically generated, and the special features for the formation of target outputs sequence are described in section III.2. Fig. 3 represents the learning process of NNs (RNNs or MLP networks). The learning process consists of two steps, from basic training to discriminative training. Each step of the learning process is explained in section III.3. The result of the learning process is a speaker model for each client. For each enrolled speaker, one speaker model, which is built of a trained NN, is stored in the system database. The dimension of the input vector $\mathbf{X}(t)$ for a NN is $I = 32$ which is also the dimension of the feature vector of input speech. The number of the output neuron nodes

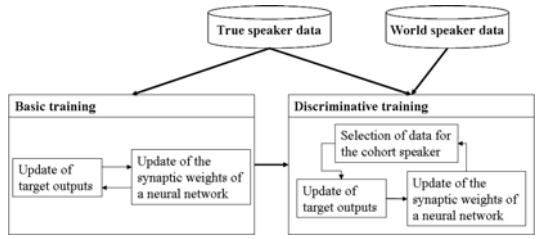

Fig. 3 The learning process of neural networks in the proposed TD speaker verification system

of NNs is determined in accordance with the number and time-varying characteristics of phonemes contained in the pronounced sentence (the 4-connected-digits speech). More details are explained in section III.2.

In the verification phase, a claimed client pronounces the keywords. Then the NN based speaker model for the claimed identity of the client is selected. The matching score is calculated from this selected NN's outputs for the pronounced keywords. By the hard thresholding of the matching score, the TDSV system decides whether the claimed client pronounces exactly the keywords and whether to accept or reject the speaker. More details for calculation of the matching score are explained in section III.4.

The adaptation phase can have a place, when the TDSV system is based on RNNs. After the verification phase, the adaptation phase trains RNNs for the speaker's utterances which have passed the SV test. While the system continues to perform SV functions for a long time, the adaptation phase also continues to adapt to the gradual change of characteristics of each individual speaker's voice that has passed the SV test.

3.2 The generation of sequence of target outputs

In order to automatically generate target outputs of a NN for a speech utterance, the state transition model of Fig. 4 is used and models the transitions between speech frames. States and state transitions represent phones and the transitions between phones of any speech utterance. If the number of states of the state transition model is *N*, the number of output neuron nodes of the NN is *N*. In the state transition model of Fig. 4, if

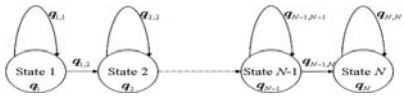

Fig. 4 The state transition model for generating target outputs of a neural network

the number of states is N=3, the state transition probability matrix is determined as follows:

$$\begin{bmatrix} q_{1,1} & q_{1,2} & q_{1,3} \\ q_{2,1} & q_{2,2} & q_{2,3} \\ q_{3,1} & q_{3,2} & q_{3,3} \end{bmatrix} = \begin{bmatrix} \pi & \pi & 0 \\ 0 & \pi & \pi \\ 0 & 0 & \pi \end{bmatrix}, \quad (3)$$

where state transition probability $q_{i,j}=p(\text{State } j \mid \text{State } i)$ and π is a constant. Values of outputs of the NN for speech frame t are used as the state probabilities, $q_n=p(\mathbf{X}(t) \mid \text{State } n)$, of the state transition model at speech frame t (where an uniform distribution of the probabilities of the phones, i.e. equal class probabilities, is assumed). If $\mathbf{Y}(t)=[s_1(t),s_2(t),...,s_N(t)]$ is output vector of the NN for speech frame t, state probabilities of the state transition model are determined as $\mathbf{Z}(t)=[q_1(t)=s_1(t),q_2(t)=s_2(t),...,q_N(t)=s_N(t)]$. For the input speech, $\mathbf{U}=[\mathbf{X}(1),\mathbf{X}(2),...,\mathbf{X}(T)]$, that has a total of T speech frames, the sequence of outputs of the NN is generated as $\mathbf{O}=[\mathbf{Y}(1),\mathbf{Y}(2),...,\mathbf{Y}(T)]$, and the sequence of state probabilities is generated as $\mathbf{P}=[\mathbf{Z}(1),\mathbf{Z}(2),...,\mathbf{Z}(T)]$. From the sequence **P** and the state transition probability matrix of Eq. 3, the Viterbi algorithm produces the most likely sequence $\mathbf{Q}=[n(1),n(2),...,n(T)]$ of states for the observed sequence $\mathbf{U}=[\mathbf{X}(1),\mathbf{X}(2),...,\mathbf{X}(T)]$. The sequence **Q** can be used to divide the input speech into speech segments, and each segment consists of speech frames that have the same phone. That is, the input speech is segmented into phones by the sequence **Q**. The target outputs of the NN are determined by the sequence **Q**. The target outputs, $\mathbf{G}(t)$, of output neuron nodes of the NN for the speech frame t is determined as follows:

$$\mathbf{G}(t)=[g_1(t),g_2(t),...,g_{N-1}(t),g_N(t)], \quad (4)$$

where $g_n(t)=\begin{cases}1.0, & \text{when } n(t)=n \\ 0.0, & \text{when } n(t)\neq n\end{cases}$.

In Eq. 4, n is the output neuron node index or the state index, and $g_n(t)$ is the target output of the output neuron node n.

When Eq. 4 is used to generate the target outputs of a NN for an observed sequence **U**, the sequence $\mathbf{C}=[\mathbf{G}(1),\mathbf{G}(2),...,\mathbf{G}(T)]$ of the target outputs is formed as the lower part of Fig. 5. Target outputs are generated according to time-varying characteristics of the input speech sounds, shown in Fig. 5. When a NN is trained by using target outputs of Eq. 4, the sequence **O** of outputs of the NN for the input speech will reflect time-varying characteristics of the input speech sounds.

For an input speech, the updating of target outputs and the training of a NN are iteratively conducted. That is, shown in the left side (the basic training) of Fig. 3, the NN is iteratively trained by the target outputs which are iteratively updated by outputs of the NN that has been trained by the previous target outputs. Two processes of the NN training and the target outputs updating are executed alternatively and iteratively. Through this iterative training process, the input speech is more and more accurately segmented into phones by the sequence **Q** which is obtained from outputs of the NN and the Viterbi algorithm. In addition, through the iterative training process, the NN captures effectively time-varying characteristics of the input speech sounds, and the sequences **O** or **C** represent more accurately time-varying characteristics of the input speech sounds.

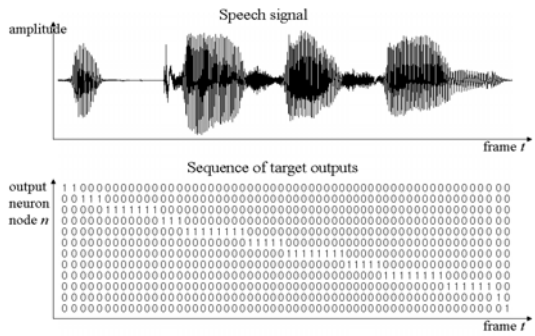

Fig. 5 An input speech signal and the sequence of target outputs of a neural network for the input

When a NN is trained for a speech utterance of a speaker by using target outputs of Eq. 4, the sequence of outputs of the NNs for the speech utterance follows the pattern of the sequence **C** for the speech utterance. On the contrary, the sequence of outputs of the NNs for a speech utterance of another speaker does not follow the pattern of the sequence **C** for the speech utterance of another speaker. This provides an advantage for speaker verification. More description is provided in section 3.4.

3.3 The training procedure for neural networks in the enrollment phase

3.3.1 The initialization of NNs' weights and the basic training process

The basic training shown in Fig. 3 is represented in more detail by Fig. 6. In the basic training process, a NN is trained for speech data of a true speaker. Before the iterative processes that rotate between the NN training and the target outputs updating, the initialization of the NN is performed. In the initialization of the NN, the speech input of the true speaker is segmented equally into N intervals, where N is the number of output neuron nodes of the NN and the value of N is determined to be three times as many as the number of syllables that are contained in the pronounced keywords for the TDSV. From this initial segmentation, the sequence **Q** (of N states) is obtained. Then target outputs are obtained by Eq. 4 according to this sequence **Q**. Finally, the initialization of the NN's weights is performed by training the NN with the RTRL (or BP) algorithm and these target outputs. The iterative training process is performed for the NN, which has the initial synaptic weights obtained through the initialization of the NN.

3.3.2 The discriminative training process

Typical state-of-the-art SV systems build background models from speaker independent databases[3,6]. Some studies advocate that the background model should be derived from speakers randomly selected from speaker

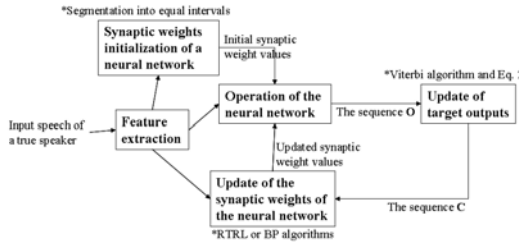

Fig. 6 The basic training of a neural network for a speaker verifier

independent databases[3]. Such a background is called world model. Others suggest to select speakers (cohort speakers) that are close to the customer. Those representatives of the population near the claimed speaker compose the cohort model[18], which is expected to improve the selectivity of the system against voices similar to the customer. In this paper, to improve the selectivity of our system against voices similar to the customer, NNs are made to pass through the process of discriminative training between the true speaker data and the cohort speaker data, where the true speaker data are speech data of the customer and the cohort speaker data are speech data of cohort speakers whose voice characteristics are close to the customer's voice characteristics.

Our system needs the world speaker data that are speech data of speakers randomly selected from speaker independent databases. The world speaker data will contain the cohort speaker data with which a speaker verifier (speaker model) confuses the true speaker data. In the discriminative training process, the cohort speaker data are extracted from the world speaker data. To extract the cohort speaker data of a speaker model k, our system computes score values of the world speaker data. Score values are MSE values between target outputs (by Eq. 4) and outputs of the speaker model k for the world speaker data. Speech data, which finish in the top L in the score ranking of the world speaker data, become the cohort speaker data for the speaker model k, where L is the number of the cohort speaker data. Shown in the right side (the discriminative training) of Fig. 3, three processes (i.e. selection of the cohort speaker

data, update of target outputs, and update of the synaptic weights of a NN) are executed alternatively and iteratively. In the discriminative training for the speaker model k, target outputs for the true speaker data are determined by Eq. 4 and target outputs for the cohort speaker data are determined in contrast with the true speaker data. By using the above described method, we can obtain sequences **Q** for the cohort speaker data from the NN of the speaker model k. From the obtained sequences **Q**, a target output for each output neuron node of the NN of the speaker model k is determined as follows: for a speech frame t of the cohort speaker data,

$$g_n(t) = \begin{cases} 0.0, & \text{when } n(t) = n \\ 1.0, & \text{when } n(t) \neq n \end{cases}. \quad (5)$$

Then the sequence $\mathbf{C}_l = [\mathbf{G}_l(1), \mathbf{G}_l(2), ..., \mathbf{G}_l(T)]$ of target outputs for the cohort speaker data is formed, where l is the index of the cohort speaker data and by Eq. 5, $\mathbf{G}_l(t) = [g_1(t), g_2(t), ..., g_{N-1}(t), g_N(t)]$ is determined. In addition, $\mathbf{C}_r$ is determined by Eq. 4, where r is the index of the true speaker data. By using Eqs. 4 and 5, target outputs are determined differently for the true and the cohort speaker data, and the discriminative training between the true and the cohort speaker data is performed by these target outputs. This discriminative training will make the NN of the speaker model k learn speech data with discrimination.

The number of the true speaker data is different from the number of the cohort speaker data. This imbalance between the true and the cohort speaker data can make trouble in the discriminative training. To overcome an obstacle caused by this imbalance, the cost function d_k of the RTRL algorithm (or BP algorithm) for the discriminative training of the speaker model k is defined as follows:

$$d_k = \frac{1}{L/R}\sum_{l=1}^{L} E_k(\mathbf{U}_l) + \frac{1}{R/L}\sum_{r=1}^{R} E_k(\mathbf{U}_r), \quad (6)$$

where $\mathbf{U}_l$ is the l'th cohort speaker data, $\mathbf{U}_r$ is the r'th true speaker data, L is the number of the cohort speaker data, R is the number of the true speaker data, and $L > R$. In addition, $E_k(\mathbf{U}_l)$ is the MSE value between target outputs (by Eq. 5) and outputs of the speaker model k for the cohort speaker data l, and $E_k(\mathbf{U}_r)$ is the MSE value between target outputs (by Eq. 4) and outputs of the speaker model k for the true speaker data r. Moreover, $E_k(\mathbf{U}_r)$ is identical to the cost function of the basic training of the speaker model k for the true speaker data r. The MSE, $E_k(\mathbf{U}_r)$, is obtained as follows:

$$E_k(\mathbf{U}_r) = \sum_{t=1}^{T} e_{k,r}(t), \quad (7)$$

where $e_{k,r}(t)$ is mean squared error between target outputs and outputs of the NN of the speaker model k at the speech frame t of $\mathbf{U}_r$. When target outputs and outputs of the NN of the speaker model k for $\mathbf{U}_r$ are $\mathbf{G}_r(t)$ and $\mathbf{Y}_r(t)$ at the speech frame t, $e_{k,r}(t)$ is represented as follows: drop subscripts k and r for the brief sign of the numerical formula, without confusion,

$$e(t) = \frac{1}{N}\left\|\mathbf{G}(t) - \mathbf{Y}(t)\right\|^2 = \frac{1}{N}\sum_{n=1}^{N}\left(g_n(t) - s_n(t)\right)^2. \quad (8)$$

Therefore, in the discriminative training, the adjustment, which is applied to the synaptic weight w_{ij}^k between the neuron nodes i and j of the NN of the speaker model k, is represented for a speech data $\mathbf{U}_a$ (where a is the index of speech data) as follows:

$$\Delta w_{ij}^k = -\eta\frac{\partial d_k}{\partial w_{ij}^k} = -\eta\frac{\partial d_k}{\partial E_k(\mathbf{U}_a)}\frac{\partial E_k(\mathbf{U}_a)}{\partial w_{ij}^k}, \quad (9)$$

where $\dfrac{\partial d_k}{\partial E_k(\mathbf{U}_a)} = \begin{cases} \dfrac{1}{R/L}, & \text{when } \mathbf{U}_a = \mathbf{U}_r \\ \dfrac{1}{L/R}, & \text{when } \mathbf{U}_a \neq \mathbf{U}_r \end{cases}$.

In Eq. 10, the constant η is learning rate and the term $\partial E_k(\mathbf{U}_a)/\partial w_{ij}^k$ is computed through the RTRL algorithm (or BP algorithm). In addition, when $\mathbf{U}_a = \mathbf{U}_r$, $E_k(\mathbf{U}_a)$ is computed as $E_k(\mathbf{U}_r)$, and when $\mathbf{U}_a \neq \mathbf{U}_r$, $E_k(\mathbf{U}_a)$ is computed as $E_k(\mathbf{U}_l)$.

3.4 The matching score for speaker verification

In the verification phase, for verifying the claimed identity of the client k or for detecting an imposter, the matching score is computed from outputs of the NN of the speaker model k for the pronounced keywords. The resulting score is compared to a given threshold. In cases of hybrid HMM-MLP or HMM based SV systems, a conventional method for calculating the matching score is to compute the likelihood score by using the forward algorithm or the Viterbi algorithm. Since our system has the state transition model of Fig. 4, our system can use a matching score computed from the Viterbi algorithm in the same way as HMM based systems. But, in this paper, the MSE values are used as the matching score. NNs are trained by the MSE criterion of Eqs. 7 and 8. Such supervised training decreases the error of Eq. 8 between target outputs and outputs of a NN at each speech frame t. In addition, for true speaker data, such supervised training decreases the error between the sequences $\mathbf{C}_r$ and $\mathbf{O}_r$ of the NN of the speaker model. Moreover, for the cohort speaker data, the discriminative training uses the target outputs obtained by Eq. 5. Therefore, for the speech utterance of an imposter, such discriminative training increases the error between the sequences $\mathbf{C}$ (by Eq. 4) and $\mathbf{O}_l$ of the speaker model of the client that the imposter impersonates. Shown in the MSE criterion of Eqs. 7 and 8, the training for NNs is performed on all the output neuron nodes of NNs. The value of the MSE of Eq. 7 contains information obtained from all the output neuron nodes of a NN at each speech frame, and impartially reflects information from each output neuron node. On the contrary, the matching score computed by the Viterbi algorithm (this paper will call this score the Viterbi score) contains information obtained from only one node of output neuron nodes of a NN at each speech frame (i.e., the maximum likelihood path is used[19]), and the use of matching score obtained by the forward algorithm incurs unexpected danger of deterioration of the SV performance in our SV system (our system uses very simple state transition model. More detailed and complex transition models may lead to much better results). Most of all, because NNs are trained by the MSE criterion of Eqs. 7 and 8, using MSE values as the matching score will show good SV performance. Therefore, as the matching score, our system uses the MSE score, E_{score}, defined as follows:

$$E_{SCORE,k}(\mathbf{U}_a) = \frac{1}{T}\sum_{t=1}^{T} e_{k,a}(t), \tag{10}$$

where $E_{SCORE,k}(\mathbf{U}_a)$ is the MSE score of a test data $\mathbf{U}_a$ for a speaker model k. In addition, when target outputs and outputs of the NN of the speaker model k for the test data $\mathbf{U}_a$ are $\mathbf{G}_a(t)$ and $\mathbf{Y}_a(t)$ (where $\mathbf{G}_a(t)$ is obtained by Eq. 4) at the speech frame t, $e_{k,a}(t)$ is represented as $e_{k,a}(t) = (1/N)\|\mathbf{G}_a(t) - \mathbf{Y}_a(t)\|^2$ by using Eq. 8. In this paper, through the SV experiments, the MSE score (by Eq. 10) and the Viterbi score are compared in terms of EER.

Ⅳ. Experiments and results

A HMM based SV system, a MLP network based SV system, and a RNN based SV system are compared in terms of the SV performance (EER).

4.1 Test database

Speech database for the TDSV test contains the Korean 4-connected-digit-words speech[20]. Speech data was recorded in a soundproof room with HMD224X and KAY CSL 4300B was used in A/D conversion. In addition, speech data was sampled in 16 kHz and quantized in 16 bits. For the true speaker data, speech data of ten male and ten female speakers are used. The imposters for the SV test are composed of ten male and ten female speakers. For the world speaker data, speech data of 17 male and 11 female speakers are used. The number of the cohort speaker data extracted from the world speaker data is nine for each speaker model. Three utterances out of four

utterances of each speaker are used in the training phase, and one utterance out of four utterances of each speaker is used in the test phase. Speech utterances of speakers are downsampled in 8 kHz and endpoints of speech are detected automatically using energy of signal. All utterances are pre-emphasized with a factor of 0.97. A Hamming window with 32ms window length and 16ms window shift is used for each speech frame. Feature vectors (32-dimensional vector) of speech are extracted into 16 LPCC coefficients and their deltas.

4.2 The configuration of neural networks and HMM

MLP networks of the MLP networks based SV system have 32 input neuron nodes, one hidden layer having 20 hidden neuron nodes, and 12 output neuron nodes. The MLP network of each speaker model is initialized by the initialization method of Fig. 6 (by 50 iterations). Then the MLP network is trained by the basic training process during 400 iterations. Finally, the MLP network is trained by the discriminative training process during 400 iterations. The learning rate is fixed as η = 0.7 for all the training processes.

For the HMM based SV system, the whole word model in HMM is used. For segmentation of speech, HMM models with the segmental K-means algorithm are used. In the case of the HMM based SV system, only the likelihood score can be used as the matching score and the HMM system cannot use the MSE score.

RNNs of the RNN based SV system have 14 neuron nodes and 32 input neuron nodes. Twelve neuron nodes out of 14 neuron nodes are output neuron nodes. That is, in Fig. 1, M = 14, N = 12, and I = 32. The RNN of each speaker model is initialized by the initialization method of Fig. 6 (by 200 iterations) with the learning rate η = 0.03. Then the RNN is trained by the basic training process with η = 0.07 during 200 iterations. Finally, the RNN is trained by the discriminative training process with η = 0.07 during 200 iterations.

4.3 The experimental results

Only after the basic training process, the performance of systems is compared in terms of EER, and after the basic and discriminative training processes of NNs, the performance is also compared in terms of EER. In Table 1, only after the basic training process, the performance of each TDSV system is represented in terms of EER. In the HMM based SV systems of Table 1, the method used for segmentation of speech is represented in parentheses. In NNs based SV systems, the performance of the MSE score is compared with the performance of the Viterbi score. The MSE score shows better performance than the Viterbi score in terms of EER. When MLP networks and RNNs are compared, RNNs show better performance than MLP networks, because of dynamic properties of RNNs. In addition, the RNN based system with the MSE score shows better performance than HMM based systems with the likelihood score. In Table 2, the performance of NNs based systems, which have the basic and discriminative training processes, is compared with the performance of NNs based systems which have only the basic training process. Table 2 shows the effect of the discriminative training. The discriminative training gives more improved performance to NNs based SV systems which use the MSE score as the matching score. But the discriminative training deteriorates the performance of NNs based systems which use the Viterbi score as the matching score. We can find out the reason of these phenomena from Fig. 7. Fig 7 represents the output value of each output node of a NN for a frame of input speech. In Fig. 7, after only the basic training process, output

Table 1. EER (equal error rate) of each TDSV system.

Speaker models	EER	
	The Viterbi score or likelihodd score	The MSE score
MLP networks	6.87%	4.20%
RNNs	14.76%	1.05%
HMMs (with segmental K-means algorithm)	1.92%	N/A

Table 2. EER of neural networks based TDSV systems.

Speaker models	Only Basic training		Basic and discriminative training	
	Viterbi score	MSE score	Viterbi score	MSE score
MLP networks	6.87%	4.20%	11.74%	1.45%
RNNs	14.76%	1.05%	15.14%	0.66%

values of output nodes of a NN for a frame of input speech are plotted by using square markers on the n-s_n coordinate plane, where n is the index of output nodes of the NN and s_n is the output value of the output node n. After the basic and the discriminative training processes, circular markers represent output values of output nodes of the NN for the frame of input speech. In Fig. 7, the more the value of MSE (between target outputs and outputs of the NN for the true speaker data) is decreased, the more output values of the NN tend to be plotted toward the graph of the solid line, rather than the graph of the dotted line. In addition, the MSE score is computed by using output values of all the output nodes of a NN for each frame of input speech. On the contrary, the Viterbi score is computed by using the output value of one output node which generates the maximum output value at each frame of input speech. Therefore, output values represented by the graph of the dotted line produce good results in NNs based TDSV systems with the Viterbi score, and output values represented by the graph of the solid line produce good results in NNs based TDSV systems with the MSE score. That is, because NNs are trained by MSE criterion decreasing the value of MSE, using

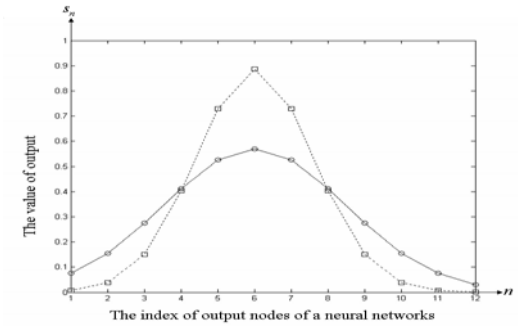

Fig. 7 Output values of output nodes of a neural network for a speech frame

the MSE score as the matching score can show better performance than using the Viterbi score in NNs based TDSV systems. The better the performance by using the MSE score becomes, the more the performance by using the Viterbi score is deteriorated. Shown in Tables 1 and 2, the proposed RNN based TDSV systems show better performance than HMM based TDSV systems, especially in circumstances of having few training data for the true speaker.

Ⅴ. Conclusion

This paper proposed a RNN based TDSV system using MSE score. The training methods for RNN of the system were presented. In addition, for making a decision regarding the identity claim of a speaker, the MSE score was presented as the matching score. In both of the enrollment and verification phases, the target outputs were automatically generated both for the training of NNs and for the calculation of the MSE score, and the sequence of the generated target outputs represented time-varying characteristics of input speech sounds. These target outputs played an important role in the improvement of the TDSV performance. RNNs were employed to capture temporally dynamic characteristics of speech signal, and the virtue of the employed RNNs was shown through experiments for comparing RNNs with MLP networks. In NNs based TDSV systems, the proposed MSE score showed better performance than the Viterbi score in terms of EER. The proposed discriminative training gave more improved selectivity (against similar voices with customers) and performance to NNs based TDSV systems. In circumstances of having few voice data for customers, proposed RNN based TDSV systems showed better performance than conventional HMM based systems in terms of EER.

References

[1] S. Furui, "An overview of speaker recognition technology," *ESCA workshop on automatic speaker*

recognition, identification and verification, pp. 1-9, Apr. 1994.

[2] Ben Gold, Nelson Morgan, *Speech and audio signal processing, processing and perception of speech and music*, John wiley & Sons, Inc., 2000.

[3] D.A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech communication*, pp. 91-108, 1995.

[4] B. Yegnanarayana, S.R.M. Prasanna, J.M. Zachariah, C.S. Gupta, "Combining evidence from source, suprasegmental and spectral features for a fixed-text speaker verification system," *IEEE Transactions on speech and audio processing*, 13(4), pp. 575-582, July 2005.

[5] Aleš Padrta, Vlasta Radová, "On the background model construction for speaker verification using GMM," *LNCS* 3206, pp. 425-432, 2004.

[6] C.O. Dumitru, I. Gavat, R. Vieru, "Speaker verification using HMM for Romanian language," *48th International Symposium ELMAR-2006 focused on multimedia signal processing and communications*, pp. 131-134, June 2006.

[7] H-S. Liou, R. Mammone, "A subword neural tree network approach to text-dependent speaker verification", in *ICASSP, IEEE*, 1995.

[8] M.D. Richard , R.P. Lippmann, "Neural network classifiers estimate Bayesian a posteriori probabilities," *Neural Computation*, 3, pp. 461-483, 1991.

[9] Simon Haykin, Neural networks, a comprehensive foundation, 2nd ed., *Prentice-Hall, Inc.*, pp. 635-789. 1999.

[10] D.P. Delacretaz, J. Hennebert, "Text-prompted speaker verification experiments with phoneme specific MLPs," in *ICASSP, IEEE,* 2, pp. 777-780, 12-15 May 1998.

[11] C.S. Gupta, S.R. Mahadeva Prasanna, B. Yegnanarayana, "Autoassociative neural network models for online speaker verification using source features from vowels", in *Proc. of IJCNN '02*, 2, pp. 1252-1257, 12-17 May 2002.

[12] J.M. Naik, D.M. Lubensky, "A hybrid HMM-MLP speaker verification algorithm for telephone speech," in *ICASSP, IEEE*, 1, pp. I/153-I/156, 19-22 April 1994.

[13] M.F. Benzeghiba, H. Bourlard, "Hybrid HMM/ANN and GMM combination for user-customized password speaker verification," in *ICASSP, IEEE*, 2, pp. II/225- II/228, 6-10 April 2003.

[14] Y. Liu, M. Russell, M. Carey, "The role of dynamic features in text-dependent and -independent speaker verification," in *ICASSP 2006*, IEEE, pp. 669-672, 2006.

[15] X. Wang, "Text-dependent speaker verification using recurrent time delay neural networks for feature extraction," in *Proc. of IEEE-SP workshop neural networks for signal processing III '93*, pp. 353-361, 6-9 Setp. 1993.

[16] J. R. Saeta, J. Hernando, "Weighting scores to improve speaker-dependent threshold estimation in text-dependent speaker verification," *LNCS* 3817, pp. 81-91, 2006.

[17] R.J. Williams, D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, 1, pp. 270-280, 1989.

[18] A.E. Rosenberg, J. DeLong, C.H. Lee, B.H. Juang, F.K. Soong, "The use of cohort normalized scores for speaker verification," In *Proc. Int. Conf. on spoken language processing*, Banff, Alberta, Canada, pp. 599-602, 1992.

[19] L. Rabiner, B.H. Juang, Fundamentals of speech recognition, *Prentice-Hall International, Inc.*, 1993.

[20] Korean speech database CD-ROM, *the Korean Language Engineering Center*, 1998.

김 태 형 (Tae-Hyung Kim) 정회원

1999년 2월 부산대학교 전자공학과 석사

2007년 2월 부산대학교 전자공학과 박사

2007년~현재 국방과학연구소 연구원

<관심분야> 음성인식, 영상처리, 레이더 신호처리, 패턴인식, 지능정보처리