PAPER

Distinctive Phonetic Feature (DPF) Extraction Based on MLNs and Inhibition/Enhancement Network

Mohammad Nurul HUDA[†a)], **Hiroaki KAWASHIMA**[†], *Nonmembers*, *and* **Tsuneo NITTA**[†], *Member*

SUMMARY This paper describes a distinctive phonetic feature (DPF) extraction method for use in a phoneme recognition system; our method has a low computation cost. This method comprises three stages. The first stage uses two multilayer neural networks (MLNs): MLN_{LF-DPF}, which maps continuous acoustic features, or local features (LFs), onto discrete DPF features, and MLN_{Dyn}, which constrains the DPF context at the phoneme boundaries. The second stage incorporates inhibition/enhancement (In/En) functionalities to discriminate whether the DPF dynamic patterns of trajectories are convex or concave, where convex patterns are enhanced and concave patterns are inhibited. The third stage decorrelates the DPF vectors using the Gram-Schmidt orthogonalization procedure before feeding them into a hidden Markov model (HMM)-based classifier. In an experiment on Japanese Newspaper Article Sentences (JNAS) utterances, the proposed feature extractor, which incorporates two MLNs and an In/En network, was found to provide a higher phoneme correct rate with fewer mixture components in the HMMs.

key words: *distinctive phonetic feature, hidden Markov model, multilayer neural network, inhibition/enhancement network, local features*

1. Introduction

Conventional automatic speech recognition (ASR) systems use stochastic pattern matching techniques in which word candidates are matched against word templates represented by hidden Markov models (HMMs). Although these techniques perform adequately in certain limited applications, they always reject a new vocabulary or a so-called out-of-vocabulary (OOV) word. Therefore, an accurate phonetic typewriter or a phoneme recognizer is expected to assist next-generation ASR systems in resolving this OOV-word problem via a short interaction (talk-back) by automatically adding the word into a word lexicon from the phoneme string of an input utterance [1], [2].

Various methods have been proposed to realize phoneme recognition [3]–[6]. Although some of them perform adequately, most HMM-based methods have several limitations. For example, (a) they require a large number of speech parameters and a large speech corpus to solve coarticulation problems using context-sensitive triphone models, and (b) they require a higher computational cost to achieve an acceptable performance. On the other hand, a system based on articulatory features or distinctive phonetic features (DPFs) can model coarticulatory phenomena more naturally [7].

A previous study introduced a DPF-based feature extraction method that used a multilayer neural network (MLN) to extract DPFs [8]. This DPF-based method (i) provides robust features under different acoustic environments with fewer mixture components in HMMs, and (ii) it improves the margin between acoustic likelihoods. Figures 1 (a) and 1 (b) show the phoneme distances of five Japanese vowels in an utterance /ioi/ that are calculated with a mel frequency cepstral coefficient (MFCC)-based ASR system and a DPF-based system, respectively. In both systems, each distance is measured using the Mahalanobis distance between a given input vector and the corresponding vowel set of mean and covariance in a single-state model. The input sequence in the figures, /i/../i//o/../o//i/../i/, specifies a phoneme for each frame and has total 20 frames in

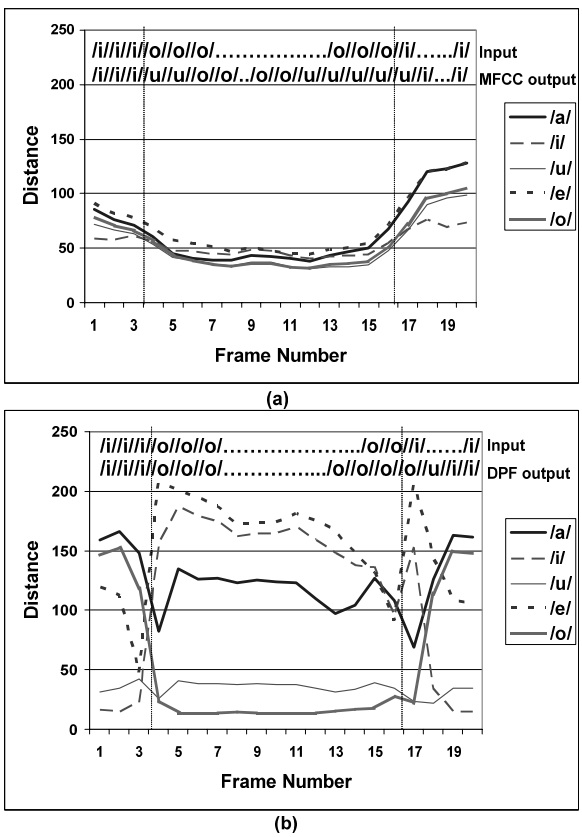

Fig. 1 Phoneme distances for input vectors of utterance /ioi/ with a) MFCC-based system and b) DPF-based system using MLN.

Manuscript received November 14, 2008.
Manuscript revised December 26, 2008.

[†]The authors are with the Graduate School of Engineering, Toyohashi University of Technology, Toyohashi-shi, 441–8580 Japan.
a) E-mail: huda@vox.tutkie.tut.ac.jp

DOI: 10.1587/transinf.E92.D.671

which first three frames, middle 13 frames, and last four frames are phonemes /i/, /o/, and /i/, respectively. The MFCC-based system (Fig. 1 (a)) shows seven misclassification of phonemes (/u/ output for /o/ and /i/ input) for frames 4, 5, 13, 14, 15, 16, and 17, while two misclassification (/o/ and /u/ output for /i/ input) for frames 17 and 18 are observed by the DPF-based system (Fig. 1 (b)). Therefore, the DPF-based system outputs few misclassification. However, because some errors caused by coarticulation still remain, as shown in Fig. 1 (b), the DPF-based system using a single MLN requires further modifications. On the other hand, a DPF extraction method can be implemented by using tandem MLNs to reduce training times and number of parameters, but S. Sivadas, et al. [9] pointed out that their feature extraction method based on tandem MLNs does not show a higher recognition accuracy over a single monolithic MLN.

In this study, we propose a DPF extraction method for constructing a more accurate phoneme recognizer by solving coarticulation problems; our method has a low computation cost. This method incorporates (i) an extra MLN, MLN_{Dyn}, to reduce misclassification at phoneme boundaries by restricting DPF context resulted from the first MLN, MLN_{LF-DPF} [8], and then embeds (ii) an inhibition/enhancement (In/En) network to obtain more precise DPF patterns for an HMM-based classifier by enhancing convex patterns (DPF peaks) and by inhibiting concave patterns (DPF dips). The effects of two MLNs (MLN+MLN or MLN_{LF-DPF}+MLN_{Dyn}) and the In/En network in our method increase phoneme recognition performance significantly over the other existing methods [8], [10]. A phoneme recognition method with low cost can be obtained by reducing the required number of Gaussian mixture components in the HMMs. In this study, we investigate and evaluate two types of DPF extraction methods along with the conventional MFCC-based method from the viewpoint of phoneme recognition performance. These methods are (i) DPF using an MLN [8], [10] and (ii) our proposed method using two MLNs and an In/En network.

This paper is organized as follows. Section 2 explains the DPF extraction methods along with the proposed method. Section 3 describes speech databases and the experimental setup. Section 4 presents the experimental results and some discussions. Finally, Sect. 5 presents the conclusions and remarks on our future works.

2. DPF Extraction Methods

A phoneme can easily be identified by using its unique DPF set [11], [12]. The Japanese balanced DPF set [10] for classifying Advanced Telecommunications Research Institute International (ATR) phonemes has 15 elements. The following subsections describe two feature extraction methods based on the balanced DPF set.

2.1 Using a Multilayer Neural Network

T. Fukuda, et al. [8] implemented a DPF-based feature extractor with an input acoustic vector of local feature (LF) using an MLN, and it was shown that LF is superior to MFCC as the input to the MLN; Fig. 2 shows the system diagram for this method. At the acoustic feature extraction stage, the input speech is first converted into LFs that represent a variation in spectrum along the time and frequency axes [13]. Two LFs are then extracted by applying three-point linear regression (LR) along the time (t) and frequency (f) axes on a time spectrum pattern, respectively. After compressing these two LFs with 24 dimensions into LFs with 12 dimensions using discrete cosine transform (DCT), a 25-dimensional (12 Δt, 12 Δf, and ΔP, where P stands for the log power of a raw speech signal) feature vector called LF is extracted. LFs are then entered into an MLN with four layers including two hidden layers after combining a current frame x_t with the other two frames that are three points before and after the current frame (x_{t-3}, x_{t+3}). The MLN has 45 output units (15×3) corresponding to a set of triphones, or to a context-dependent DPF vector that comprises three DPF vectors (a preceding context DPF, a current DPF, and a following context DPF) with 15 dimensions each. The two hidden layers comprise 256 and 96 units, respectively. The MLN is trained using the standard back-propagation algorithm. The DPF-based method has a recognition performance comparable to that of the MFCC-based method although it requires fewer Gaussian mixture components in the HMM. However, because a single MLN suffers from the inability to handle a longer context, it exhibits some misclassification at the phoneme boundaries.

2.2 Proposed Method

Figure 3 shows the proposed feature extraction method. This method comprises three stages. The first stage extracts 45-dimensional DPF vectors from the LFs of an input speech using two MLNs. The second stage incorporates In/En functionalities to obtain modified DPF patterns. The third stage decorrelates the DPF vectors using the Gram-Schmidt (GS) orthogonalization [10] before connecting with an HMM-based classifier.

2.2.1 DPF Extractor

In this method, two MLNs instead of a single MLN are used to construct the DPF extractor. The first MLN, MLN_{LF-DPF}, maps acoustic features, or LFs, onto discrete DPF features [8] and the second MLN, MLN_{Dyn}, reduces misclassification at phoneme boundaries by constraining the DPF context. Here, MLN_{LF-DPF} has the same architecture as that described in Sect. 2.1, and it is trained using the same learning algorithm. The 45-dimensional context-dependent DPF vector provided by the MLN_{LF-DPF} at time t, and its corresponding Δ and $\Delta\Delta$ vectors calculated by three-point LR are appended into the subsequent MLN_{Dyn} with four layers including two hidden layers of 300 and 100 units, respectively. The MLN_{Dyn} is trained using the standard back-propagation algorithm and outputs a 45-dimensional DPF

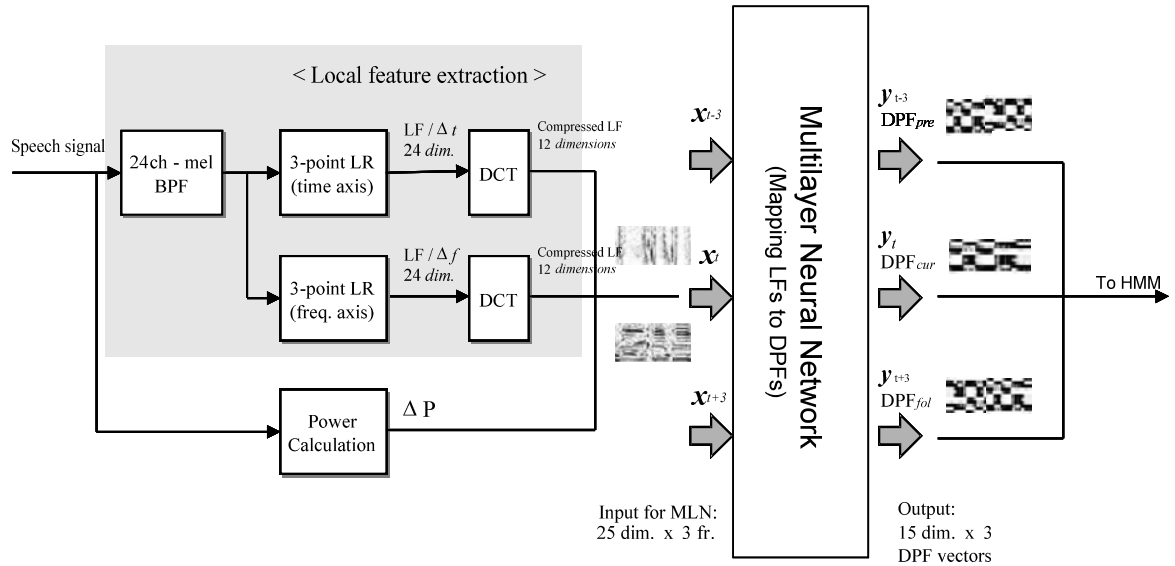

Fig. 2 DPF extractor using MLN.

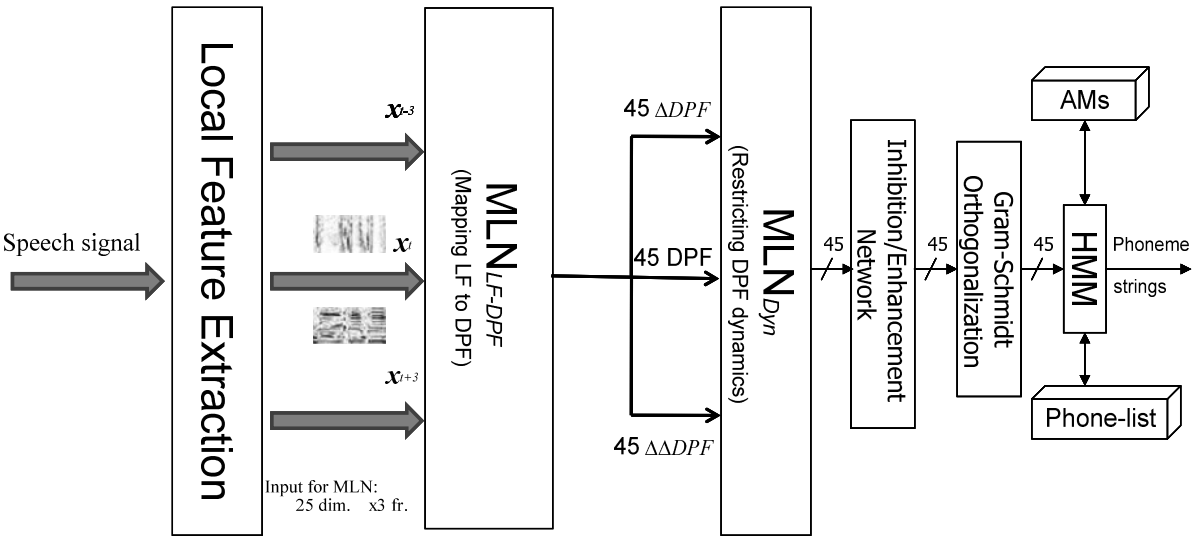

Fig. 3 Proposed orthogonalized DPF extractor.

vector in which context effects for the current "*t*"-th frame are reduced.

2.2.2 Inhibition/Enhancement Network

The DPF extractor, MLN+MLN, generates 45 DPF patterns (15 preceding context DPF patterns, 15 current context DPF patterns, and 15 following context DPF patterns) for each input speech. Because all of these 45 DPF patterns may not follow the input DPF patterns of a phoneme string exactly, there exists an ambiguity among some phonemes for classifying the target phoneme in the HMM-based classifier. Consequently, some phonemes are not correctly recognized. An ambiguity sometimes occurs when the values of consecutive DPF peaks and DPF dips in a DPF time pattern of a phoneme string are closer to each other. For example, left peak, middle dip, and right peak values generated by a DPF extractor are <0.7, 0.7, 0.7, 0.7>, <0.4, 0.4, 0.4>, and <0.7, 0.7, 0.7>, respectively, where {<0.7, 0.7, 0.7, 0.7>, <0.4, 0.4, 0.4>, <0.7, 0.7, 0.7>} or <0.7, 0.7, 0.7, 0.7, 0.4, 0.4, 0.4, 0.7, 0.7, 0.7> is a set of output DPF values of 10 consecutive frames along time axis for a labeled DPF pattern, <1, 1, 1, 1, 0, 0, 0, 1, 1, 1>. The classifier faces a problem to decide whether the output pattern is <1, 1, 1, 1, 0, 0, 0, 1, 1, 1> or <1, 1, 1, 1, 1, 1, 1, 1, 1, 1>, while labeled DPF pattern was <1, 1, 1, 1, 0, 0, 0, 1, 1, 1>. Here, the value, 0.4 is assumed as either zero or one, while the value, 0.7 is considered as one. So, there must have clear distinction between a DPF peak and dip along time axis. If there exist a mechanism that enhances DPF peak values upto a certain level and that suppresses DPF dip values accordingly, then a distinction between a peak and dip is found. We have in-

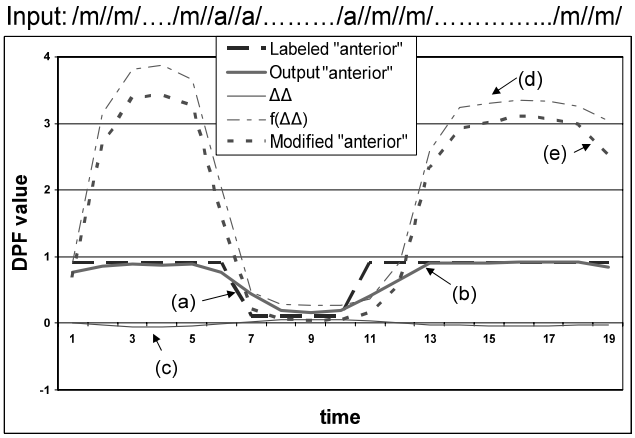

Fig. 4 Working mechanism of the In/En network. Five curves are denoted by (a), (b), (c), (d), and (e), respectively. The curves: a) Labeled "anterior" DPF for input utterance, /mam/, b) Output "anterior" DPF by MLN+MLN, c) ΔΔ for output "anterior", d) f(ΔΔ) for ΔΔ, and e) Modified "anterior" by multiplying curve (b) with curve (d).

corporated an In/En network to get this type of effect. An algorithm for this network is given below.

Step1: For each element of the DPF vectors, find the acceleration (ΔΔ) parameters by using three-point LR.
Step2: Check whether ΔΔ is positive (concave pattern) or negative (convex pattern) or zero (steady state).
Step3: Calculate $f(\Delta\Delta)$.
if pattern is convex,

$$f(\Delta\Delta) = \frac{C_1}{1 + (C_1 - 1)e^{\beta\Delta\Delta}} \tag{1}$$

if pattern is concave,

$$f(\Delta\Delta) = C_2 + \frac{2(1 - C_2)}{1 + e^{\beta\Delta\Delta}} \tag{2}$$

if steady state,

$$f(\Delta\Delta) = 1.0 \tag{3}$$

Step4: Find modified DPF patterns by multiplying the DPF patterns with f(ΔΔ).

Figure 4 shows the working mechanism of the In/En network using the "anterior" DPF pattern of an input utterance, /mam/ along time axis. In the figure, five curves, which represent (a) labeled "anterior" DPF, (b) output "anterior" generated by the MLN+MLN, (c) ΔΔ for the output "anterior" values, (d) $f(\Delta\Delta)$ for ΔΔ values, and (e) modified "anterior" DPF, are observed. Here, the curve (e) is obtained by multiplying curve (b) with curve (d). After applying the In/En network algorithm on curve (b), the DPF values of frames 1-6 and 13-19 (convex pattern or DPF peak) are enhanced, and frames 7-11 (concave pattern or DPF dip) are inhibited. It should be noted that the DPF pattern of curve (e) shows a clear distinction between a DPF peak and dip.

2.2.3 Gram-Schmidt Orthogonalization

Because each of the three 15-dimensional context vectors outputted by the In/En network is not orthogonal to each other, these three context vectors should be decorrelated using the GS orthogonalization [10] with respect to the current context vector.

3. Experiments

3.1 Speech Database

The following two clean data sets are used in our experiments.

D1. Training data set
A subset of the Acoustic Society of Japan (ASJ) Continuous Speech Database comprising 4503 sentences uttered by 30 different male speakers (16 kHz, 16 bit) is used [14].
D2. Test data set
This test data set comprises 2379 Japanese Newspaper Article Sentences (JNAS) [15] uttered by 16 different male speakers (16 kHz, 16 bit).

In the experiments, the same training data set D1 is used for the MLN_{LF-DPF}, MLN_{Dyn}, and HMM.

3.2 Experimental Setup

In this study, two types of acoustic features are used: LFs and MFCC. The frame length and frame rate are set to 25 ms and 10 ms, respectively, to obtain acoustic features from an input speech. MFCC feature consists of a vector of 38 dimensions (12 MFCC, 12 delta and 12 acceleration coefficients of MFCC, delta and acceleration coefficients of log energy of speech signal). On the other hand, LFs are a 25-dimensional vector consisting of 12 delta coefficients along time axis, 12 delta coefficients along frequency axis, and delta coefficient of log power of a raw speech signal [13].

Since our goal is to design a more accurate phoneme recognizer, phoneme correct rate (PCR) for D2 data set is evaluated using an HMM-based classifier. The D1 data set is used to design 38 Japanese monophone HMMs with five states, three loops, and left-to-right models. Input features for the classifier are MFCC features or DPFs. In the HMMs, the output probabilities are represented in the form of Gaussian mixtures, and diagonal matrices are used. The mixture components are set to 1, 2, 4, 8, and 16.

In our experiments of the MLN and MLN+MLN, the non-linear function is a sigmoid from 0 to 1 ($1/(1+\exp(-x))$) for the hidden and output layers. For evaluating the performance of a DPF extractor, we measure DPF correct rate (DCR) using D2 data set. Here, a DPF value, which is in current frame (middle 15 of 45-dimensional output vector), below 0.5 is considered to be a negative feature; otherwise, it is a positive feature. The phoneme-wise DCR is obtained

Table 1 Phonemes and their frequencies in the test data set.

Phoneme	Count	Phoneme	Count
a	151319	t	52111
i	93117	k	81168
u	51338	ts	17233
e	83847	ch	17722
o	145644	b	10736
N	41279	d	19388
w	22585	g	20383
y	15088	z	7572
j	20942	m	27714
my	186	n	40239
ky	8189	s	50261
dy	15	sh	45300
by	472	h	17465
gy	1704	f	5294
ny	1180	r	24332
hy	1720	q	16525
ry	2566	silB	108388
py	441	silE	107628
p	4294	sp	133972

by Eq. (4) after counting the total number of correctly recognized DPF values, N_c, and total number of DPF values, N, for that phoneme. The overall DCR is calculated in the same manner by taking summation over the all phonemes. Table 1 shows phonemes and their frequencies in the test data set.

$$DCR = \frac{N_c}{N} \times 100\% \tag{4}$$

For the In/En network, the value of the enhancement coefficient, C_1, is set to 4.0 after evaluating the proposed method, $DPF(MLN + MLN + In/En + GS, dim : 45)$, for different values of C_1, such as 2, 4, and 6, and the value of the steepness coefficient, β, is set to 80. The value of inhibitory coefficient, C_2, is fixed to 0.25 after observing the DPF data patterns to keep the values of $f(\Delta\Delta)$ between 0.25 and 1.0.

In the calculation of ΔDPFs for the current "t"-th frame, the width of each window is set to seven frames, $(t-3, .., t, .., t+3)$. Then, ΔΔDPFs for the "t"-th frame are calculated from the ΔDPFs by setting a window of same width. It should be noted that ΔΔDPFs cover up to 13 frames, $(t-6, .., t-3, .., t, .., t+3, .., t+6)$, of DPF values along time axis.

We have investigated DCR for MLN and MLN+MLN to show the advantages of two MLNs over a single MLN. Two measures are considered for both the DPF extractors: phoneme-wise DCR and overall DCR.

In our experiments, the phoneme recognition tests are carried out for the following methods to compare the performance of the DPF extractor, MLN+MLN, with the baseline system, MFCC-based method, which inputs a feature vector with 38 dimensions to the HMM classifier and the method proposed by T. Fukuda, et al. [8], which feeds a 45-dimensional feature vector into the classifier.

(a) MFCC (baseline,dim:38).
(b) DPF (MLN,dim:45) [8].
(c) DPF (MLN+MLN,dim:45).

On the other hand, T. Fukuda, et al. [10] incorporated the Karhunen-Loeve Transform (KLT) in their method before applying GS orthogonalization, and extracted 33-dimensional feature vector for the HMM classifier. For comparison purposes, we have further designed some phoneme recognition tests for the following methods with GS orthogonalization; all the methods except [10] input a 45-dimensional feature vector for the classifier.

(d) DPF (MLN+GS,dim:45).
(e) DPF (MLN+KLT+GS,dim:33) [10].
(f) DPF (MLN+MLN+GS,dim:45).

To observe the effect of the In/En network on phoneme recognition, we have evaluated the following methods, which input a 45-dimensional feature vector for the HMM-classifier, including the proposed method.

(g) DPF (MLN+In/En+GS,dim:45).
(h) DPF (MLN+MLN+In/En+GS,dim:45)[Proposed].

4. Experimental Results and Discussion

4.1 DPF Detection Performance

Segmentation for a clean /jiNkoese/ utterance is shown in Figs. 5 and 6 for a balanced-DPF set [10] using a single MLN and MLN+MLN, respectively. In both the figures, "Solid thin line" and "Solid bold line" represent "ideal segmentation" and "output segmentation", respectively; "nasal", "nil (high/low)", and "high" of phoneme /N/, and "unvoiced", "coronal", and "anterior" of phoneme /s/ are denoted by ①, ②, ③, ④, ⑤, and ⑥, respectively. After observing ①, ②, ③, ④, ⑤, and ⑥ marked places, we can say that the MLN+MLN exhibits more precise segmentation (less deviation from ideal boundary) than the single MLN, reduces some fluctuations caused by the first MLN, MLN_{LF-DPF} and provides more smoothed DPF curves, and hence, it misclassifies fewer phonemes.

Figure 7 shows the phoneme-wise DCRs using DPF extractors implemented by MLN and MLN+MLN. Here, the MLN+MLN provides higher DCRs for all the phonemes except /o/ and /silB/. The overall DCRs for the MLN and MLN+MLN are shown in Table 2; the MLN+MLN exhibits 1.2% improvement of DCR over the DPF extractor implemented by a single MLN. Because the second MLN, MLN_{Dyn}, resolves coarticulation effects more widely by taking dynamic feature parameters, ΔDPF and $\Delta\Delta DPF$, as input, the MLN+MLN exhibits better segmentation as well as higher DCR.

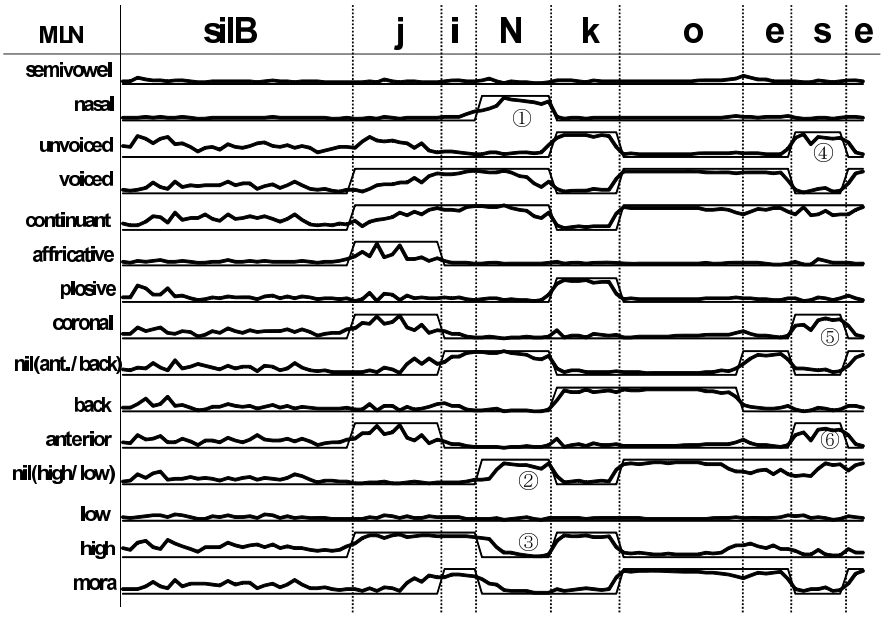

Fig. 5 Segmentation of utterance, /jiNkoese/ for DPF extractor based on single MLN.

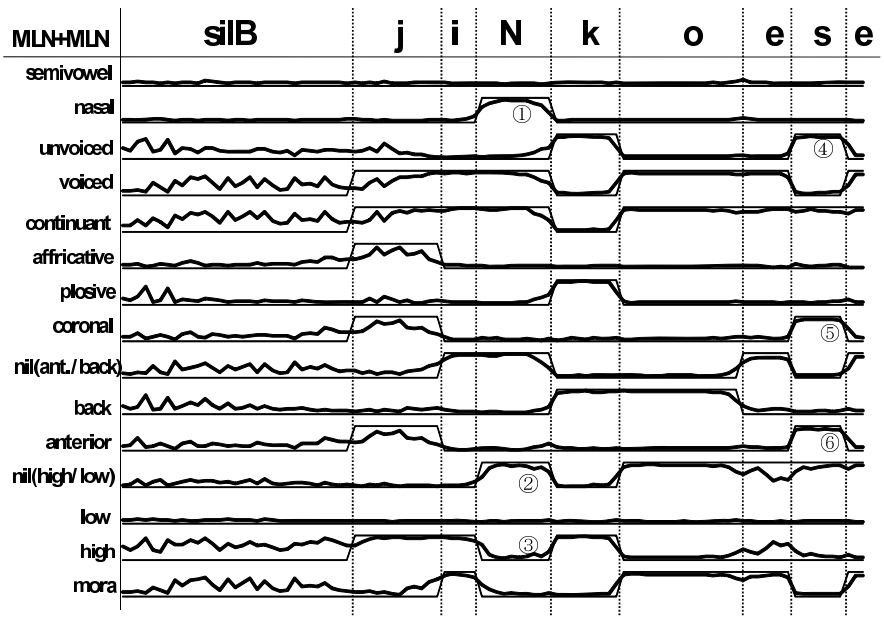

Fig. 6 Segmentation of utterance, /jiNkoese/ for DPF extractor based on MLN+MLN.

Table 2 Overall DPF Correct Rate for MLN and MLN+MLN.

DPF extraction method	DPF Correct Rate(%)
MLN	89.8
MLN+MLN	91.0

4.2 Phoneme Recognition Performance

Figure 8 shows the phoneme recognition performance using the methods (b) and (c). In the figure, we can observe that the method (c) exhibits a better performance for all investigated mixture components. For example, at mixture component 16, (c) has a PCR of 81.47%, while (b) exhibits 78.54%. It should be noted that (c) provides 2.93% higher PCR compared to (b) at mixture component 16. The phoneme recognition performance after applying the GS orthogonalization using the methods (d), (e), and (f) are given in Fig. 9; (f) exhibits its highest performance (81.60%) for mixture component 2, while (d) and (e) show 78.06% and 79.12% PCRs, respectively. At mixture component 2, (f) shows an improvement of 3.54% over the method (d). Since the MLN+MLN produces more accurate segmenta-

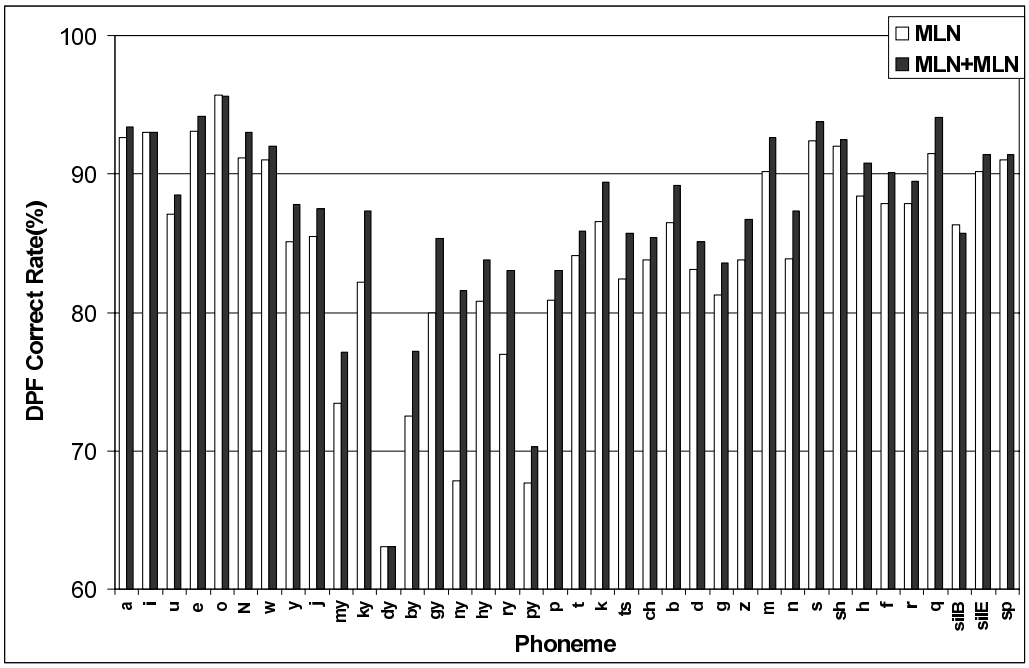

Fig. 7 Phoneme-wise DPF correct rates for MLN and MLN+MLN.

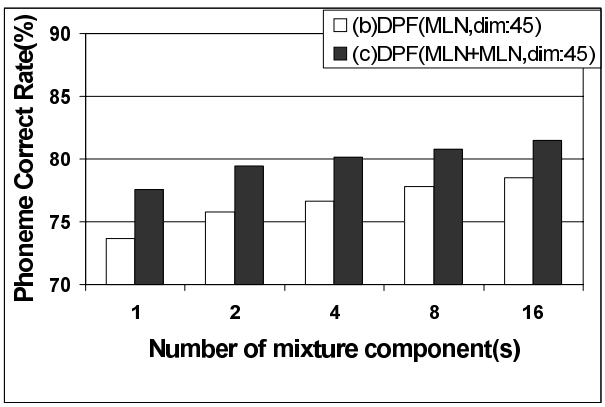

Fig. 8 Phoneme recognition performance without GS orthogonalization.

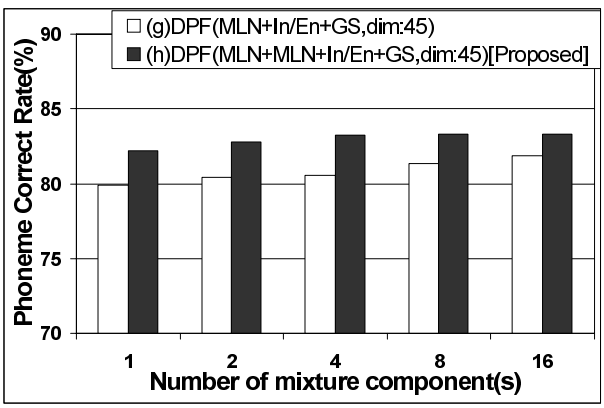

Fig. 10 Phoneme recognition performance using In/En network and GS orthogonalization.

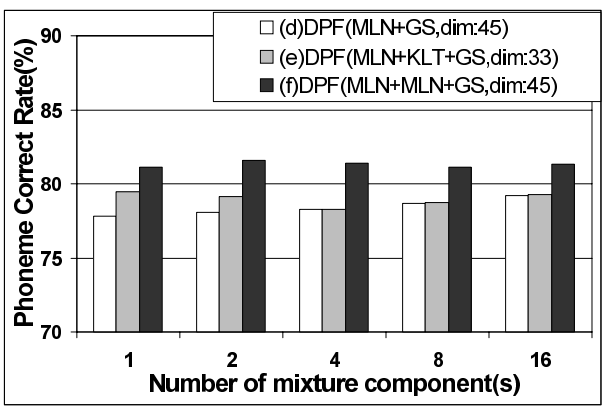

Fig. 9 Phoneme recognition performance using GS orthogonalization.

tion for an input utterance (Fig. 6) than the single MLN, better DPF patterns are obtained and hence, a higher PCR is achieved.

After incorporating the In/En network with the methods (d) and (f) of Fig. 9, we can evaluate the orthogonalized DPF, and the recognition results are shown in Fig. 10. From the figure, we can observe that the proposed method (h) provides a higher PCR over the method (g) for all mixture components. The proposed method exhibits its best PCR (83.33%) for mixture component 16. Again, Fig. 11 shows the effect of the In/En network on phoneme recognition performance in the proposed method. An improvement of 2.01% PCR at mixture component 16 by the proposed method over the method (f) illustrates the advantage of an In/En network. From Figs. 9 and 10, it should be noted that an addition of In/En network into the methods (d) and (f)

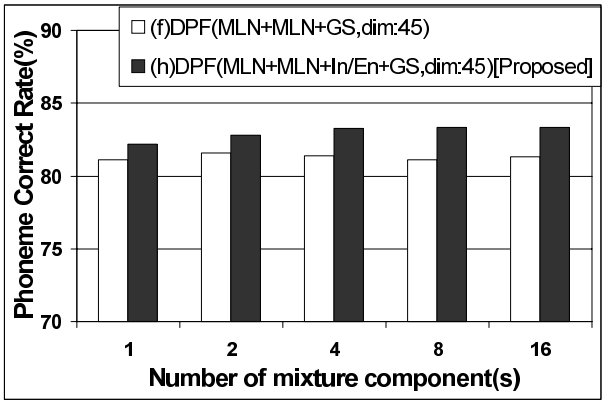

Fig. 11 Effect of the In/En network on phoneme recognition performance in the proposed method.

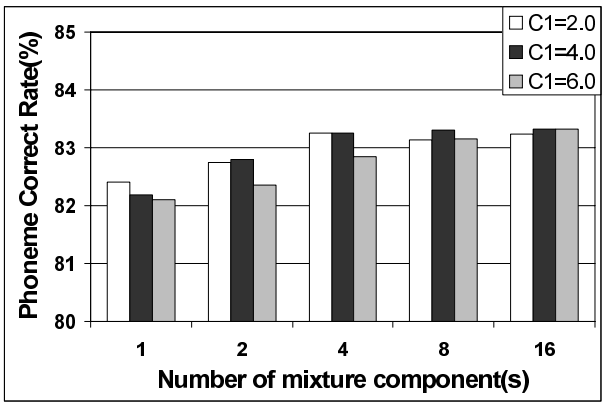

Fig. 12 Variation of recognition performance for different values of enhancement coefficient in the proposed method.

always increases PCR upto a certain level because an In/En network always gives the DPF patterns with clear distinction between a DPF peak and dip (Fig. 4, curve(e)).

The variation of performance of the proposed method for different values of enhancement coefficient, C_1, is shown in Fig. 12. At all mixture components except 1, C_1=4.0 exhibits the highest PCR over the other investigated values of C_1 and hence, C_1 is set to 4.0 for our experiments.

4.3 Advantages of the Proposed Method

Figure 13 shows a comparison of the phoneme recognition performance of the proposed method with baseline (a) and the method (e) proposed by T. Fukuda, et al. [10] for the investigated mixture components. It should be noted that the proposed method outperformed the baseline at all mixture components. For example, at mixture component 16, the proposed method (83.33% PCR) improves the performance by 10.63% in comparison with the baseline (72.70% PCR). On the other hand, at mixture component 16, an improvement of 4.04% is achieved by the proposed method in comparison with the method proposed by T. Fukuda, et al. [10] (79.29% PCR). Moreover, the proposed method requires fewer mixture components in the HMMs.

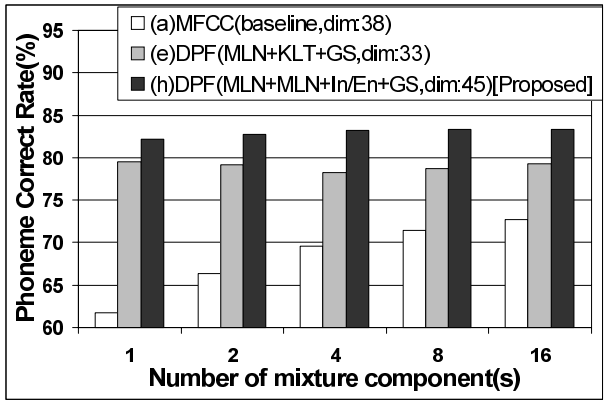

Fig. 13 Comparison among proposed and other existing methods for phoneme recognition rate.

It is claimed that our proposed method requires less computation time than the method proposed by T. Fukuda, et al. [10]. Here, we assume that the total number of frames is 1000 on an average in a speech file, and multiplication and division carry same meaning. The proposed method requires total 124,095,000 ($= 48,096,000 + 75,360,000 + 495,000 + 135,000 + 9,000$) multiplications, where the MLN_{LF-DPF}, MLN_{Dyn}, In/En network, GS orthogonalization and HMM-based classifier use 48,096,000 ($= 1000 \times (75 \times 256 + 256 \times 96 + 96 \times 45)$), 75,360,000 ($= 360,000 + 1000 \times (135 \times 300 + 300 \times 100 + 100 \times 45)$), 495,000 ($= 360,000 + (45 \times 1000 + 45 \times 1000) + 45 \times 1000$), 135,000 ($= 9 \times 15 \times 1000$), and 9,000 ($= 1 \times 3^2 \times 1000$) multiplications, respectively. In the proposed method, 75, 256, and 96 are dimensions of input, first hidden, and second hidden layers of the MLN_{LF-DPF}, respectively; 135 ($= 3 \times 45$), 300, and 100 indicate input, first hidden, and second hidden layer units of the MLN_{Dyn}, respectively; 45 is for output DPF dimensions of MLN_{LF-DPF}/MLN_{Dyn}; nine and 15 represent number of multiplications in GS procedure and dimensions of each context vector, respectively; 360,000 ($= 2 \times 45 \times 3 \times 1000 + 2 \times 45 \times 1000$) is for 45-dimensional ΔDPF and $\Delta\Delta DPF$ calculation in which two and three denote {delta, delta-delta} and three-points LR, respectively; one and three in the HMM-based classifier represent mixture component and number of states (three loops only), respectively. On the other hand, the total number of multiplications required by the method of T. Fukuda, et al. [10] is at least 3,186,280,080,150 ($= 48,096,000 + 3,186,231,840,150 + 135,000 + 9,000$) in which the MLN_{LF-DPF}, KLT procedure, GS orthogonalization, and HMM-based classifier take 48,096,000 (described in proposed), 3,186,231,840,150 (required multiplications to evaluate determinant of 15×15 matrix for calculating orthogonal basis), 135,000 (described in proposed), and 9,000 (described in proposed) multiplications, respectively. Since our proposed method and the method proposed by T. Fukuda, et al. [10] require 124.1 M (=124,095,000) and 3.2 T (=3,186,280,080,150) multiplications, respectively, our proposed method extracts DPFs with low computation cost.

On the other hand, conventional MFCC-based method requires higher mixture components (more than 16 for Fig. 13) to achieve comparable recognition performance. For each mixture component, a Viterbi search needs $O(S^2T)$ computation time, where S and T represent the number of states and the number of observation sequences, respectively. Therefore, for phoneme recognition, the MFCC-based method requires at least 144,000 ($= 16 \times 3^2 \times 1000$) multiplications at the HMM classifier stage to obtain its highest performance, while each of our proposed method and the method proposed by T. Fukuda, et al. [10] needs 9,000 ($= 1 \times 3^2 \times 1000$) multiplications using one mixture component for achieving better performance than the MFCC-based method.

In this paper, the proposed method inserts dynamic parameters, ΔDPF and $\Delta\Delta DPF$, into the second MLN, MLN_{Dyn}, to solve coarticulation problems (Fig. 6) and adopts monophone models, which require a small number of speech parameters, in the HMM classifier.

5. Conclusion

This paper presents a DPF-based feature extraction method using two multilayer neural networks and an Inhibition/Enhancement network for accurate phoneme recognition. The findings of our study include the following

(a) The DPF extractor based on two MLNs shows a higher DPF detection performance and exhibits a higher phoneme correct rate than the DPF extractor implemented with a single MLN.
(b) The proposed method, $DPF(MLN + MLN + In/En + GS, dim : 45)$, provides a higher phoneme correct rate than the existing methods [8], [10] and MFCC-based method for all mixture components investigated.
(c) A higher phoneme recognition performance can be obtained by incorporating an In/En network with the DPF extraction methods based on MLN (s)+GS.

In future work, we would like to evaluate noise-corrupted speech data at different acoustic environments for different signal-to-noise ratios (SNRs) using the proposed DPF extractor. Moreover, we intend to do some experiments for evaluating the word accuracy by incorporating the language model in near future. Since this paper is limited to monophone models, further evaluation based on DPF and MFCC using triphone models will be done in future as a continuation of our current research.

Acknowledgments

This work was supported in part by Global COE program "Frontiers of Intelligent Sensing," MEXT, Japan.

References

[1] I. Bazzi and J.R. Glass, "Modeling OOV words for ASR," Proc. ICSLP, pp.401–404, Beijing, China, 2000.
[2] O. Scharenborg and S. Seneff, "A two-pass for strategy handling OOVs in a large vocabulary recognition task," Proc. Interspeech, 2005.
[3] A. Waibel, et al., "Phoneme recognition using time-delay neural networks," IEEE Trans. Acoust. Speech Signal Process., vol.37, pp.328–339, March 1989.
[4] S.J. Young, "The general use of tying in phoneme-based HMM speech recognizers," IEEE Int. Conf. Acoust., Speech Signal Process., ICASSP'92, vol.1, pp.569–572, March 1992.
[5] K. Takuya, et al., "Recurrent neural networks for phoneme recognition," IEICE Technical Report., Speech, vol.94, no.42 (19940519), pp.1–8, May 1994.
[6] H. Suzuki, et al., "Continuous speech recognition based on general factor dependent acoustic models," IEICE Trans. Inf. & Syst., vol.E88-D, no.3, pp.410–417, March 2005.
[7] K. Kirchhoff, et al., "Combining acoustic and articulatory feature information for robust speech recognition," Speech Commun., vol.37, pp.303–319, 2002.
[8] T. Fukuda, W. Yamamoto, and T. Nitta, "Distinctive phonetic feature extraction for robust speech recognition," Proc. ICASSP'03, vol.II, pp.25–28, 2003.
[9] S. Sivadas and H. Hermansky, "Hierarchical tandem feature extraction," ICASSP-2002, vol.1, pp.809–812, May 2002.
[10] T. Fukuda and T. Nitta, "Orthogonalized distinctive phonetic feature extraction for noise-robust automatic speech recognition," IEICE Trans. Inf. & Syst, vol.E87-D, no.5, pp.1110–1118, May 2004.
[11] S. King and P. Taylor, "Detection of phonological features in continuous speech using neural networks," Comput. Speech Lang., vol.14, no.4, pp.333–345, 2000.
[12] E. Eide, "Distinctive features for use in an automatic speech recognition system," Proc. Eurospeech 2001, vol.III, pp.1613–1616, 2001.
[13] T. Nitta, "Feature extraction for speech recognition based on orthogonal acoustic-feature planes and LDA," Proc. ICASSP99, pp.421–424, 1999.
[14] T. Kobayashi, et al., "ASJ continuous speech corpus for research," Acoustic Society of Japan Trans., vol.48, no.12, pp.888–893, 1992.
[15] JNAS: Japanese Newspaper Article Sentences. http://www.milab.is.tsukuba.ac.jp/jnas/instruct.html

Mohammad Nurul Huda was born in 1973. He received his B. Sc and M. Sc. in Computer Science and Engineering degrees from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 1997 and 2004, respectively. Now, he is a Ph.D. student of Toyohashi University of Technology, Japan. His research field includes Automatic Speech Recognition. He is a student member of the Acoustic Society of Japan (ASJ) and the International Speech Communication Association (ISCA).

Hiroaki Kawashima was born in 1985. He received his B.E. degree in Knowledge-based Information Engineering from Toyohashi University of Technology, Japan, in 2007. He is currently a Master course student in the same university. His research interest is in the area of automatic speech recognition. He is a student member of the ASJ.

Tsuneo Nitta was born in 1946. He received his B.E.E. degree in 1969 and his Dr. Eng. Degree in 1988, both from Tohoku University, Japan. After engaging in research and development at the R&D Center of Toshiba Corporation and Multimedia Engineering Laboratory, where he was a chief Research Scientist, since 1998, he has been a Professor at the Graduate School of Engineering, Toyohashi University of Technology. His current research interests include speech recognition, multi-modal interaction, and acquisition of language and concepts. He received the Best Paper Award from the Institute of Electronics, Information and communications Engineers (IEICE), Japan, in 1988. He is a member of the Information Processing Society of Japan (IPSJ), the ASJ, the Japanese Society for Artificial Intelligence, and the IEEE.