Rapid adaptation of deep neural network for speech recognition

(深層学習に基づく音声認識システムの高速適応)

January 2019

Doctor of Philosophy (Engineering)

Hiroshi Seki
(関 博史)

Toyohashi University of Technology

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Automatic speech recognition (ASR) technologies have become a popularized technology for us. It is mainly due to a spread of interface, e.g., smart-phone and smart speaker, and also because of an openness of speech recognition API. In terms of technical aspect, deep learning has achieved a great success in many computer science competitions including automatic speech recognition [1, 2, 3, 4, 5, 6], and machine learning has become a big trend in the technology industry for these 10 years.

ASR systems are used across a wide range of situations. The difficulty of speech recognition task is decomposed to multiple factors, e.g., recognition target, speaker differences, environmental variability, and reverberation. Recognition of keyword and phrase with low environmental variability is applicable to control of home appliances, and recognition of spontaneous speech is applicable in scenarios such as meeting and call center. Extension to the recognition of spontaneous speech requires additional treatment in general, e.g., an introduction of grammatical information. One example of recognition of short utterance under high environmental variability is control of an automotive navigation system. Recognition under the noisy condition requires additional treatments, e.g., identification and filter out of noise signal, for robust execution. Recognition of spontaneous speech under noisy condition is applicable to a general purpose such as Application Programming Interface (API) for third parties.

ASR system models conversion from speech to text by using big data called training data. During operation, the ASR system recognizes input speech uttered by various speakers which are recorded in various conditions. However, since the system can not identify a test speaker and speech environments in advance, there is a problematic degradation in speech recognition performance owing to a mismatch between the input speech and the training data. Adaptation is one approach to alleviate this mismatch. Adaptation techniques can be roughly classified into three types: feature adaptation [7, 8, 9, 10], model adaptation [11, 12, 10, 13, 14, 15, 16], and addition of auxiliary features [17, 18, 19, 20, 21]. In the case of feature adaptation, the input acoustic feature is adapted and the adapted feature is fed into the ASR system. In the case of auxiliary feature based adaptation, information which degrades recognition performance, e.g., speaker and noise information, is extracted and the information is used as the auxiliary feature to take the noise and speaker information into consideration. In the case of model adaptation, one promising approach, parameters of the ASR system are re-updated using data called adaptation data. The adaption data are collected in the same condition as the recognition target. The distinction between model adaptation and feature adaptation is getting blurry. For example, re-update of a denoising auto-encoder can be regarded as both the feature adaptation and the model adaptation.

The adaptation techniques are employed in some existing applications. For example, Google voice search collects speech of “Ok google” for 3 times, and Cortana collects 6 utterances[*1], respectively. The existence of these data collection procedures shows the importance of adaptation to personalize the ASR system.

However, data collection depending on each speaker and each condition takes tedious time and leads to the poor user experience. In other words, users have to utter additional words before the recognition of intended words which leads to undesired high latency. Therefore, it is important to adapt the ASR system rapidly and robustly using a small amount of adaptation data. We call such low latency adaptation “rapid adaptation”. The size of adaptation data should be small because it saves time and effort of data collection. Robustness is also a key factor for adaptation. Adaptation is employed to improve recognition performance but not degradation. When the adaptation data is too small, it makes difficult to estimate the detailed auxiliary feature in the case of auxiliary feature based adaptation. In the case of model adaptation, the ASR system often overfits to the given adaptation data which leads to performance degradation.

[*1] October, 2018

1.2 Motivation

One approach for the development of robust Gaussian Mixture Model (GMM) based ASR system is composed of speech data clustering and following cluster-dependent acoustic modeling, that is, leading to multiple ASR systems. This method divides the training data into multiple clusters depending on acoustic feature similarity, and models multiple ASR systems using each cluster. In an inference stage, recognition is performed by selecting an ASR system which cluster is similar to the input speech. The cluster dependent modeling can reduce the diversity of speaker individuality and recognize the input speech using the ASR system trained by speech close to the test speaker [22, 23]. The similarity of speech is defined as the speaker's characteristics including vocal tract parameters [24], eigen voice [25], speaking rate [26], i-vector [27], and so on. Deep neural network (DNN) can take various input feature by concatenating different types of features and the feature extraction is conducted in a data-driven manner. This enables the reduction of feature engineering stage and eases the usage of auxiliary feature [18, 19, 20]. The i-vector, which represents speaker information, is used as speaker representation for the auxiliary feature based adaptation [19]. In the case of recognition of short time utterance, e.g., keyword and phrase, it is considered that a duration of the input speech is approximately within 0.5 second. However, the earlier works assume the availability of speech from several ten seconds to several minutes. Therefore, these methods are difficult to apply for the recognition of short time utterance. Tsujikawa, et al., proposed a method to estimate i-vector from a short time utterance [28]. However, they also reported that it is difficult to estimate robust speaker characteristics from a short time speech, i.e., 0.5 second. Lie, et al., investigated a relation between speech duration for i-vector estimation and recognition performance, and reported that more than 5.0 seconds are required for an improvement of speech recognition performance [29]. They also reported that the performance got worse when the acoustic characteristics in the training data do not cover that of the input speech (in the evaluation) [29]. Therefore, it is difficult to apply the conventional auxiliary feature based speaker adaptation techniques to the recognition of short time utterance.

One main advantage of DNNs is a hierarchical non-linear feature extraction under a simple objective function. Exploiting this property, some recent novel approaches focus on front-end learning based on DNNs that take low-level acoustic features [30, 31, 32, 33, 34, 35]. Sainath, et al., [34] and Sailor, et al., [33] proposed a DNN model that uses waveforms and performs a frequency analysis. These studies reported that some of the learned characteristics showed a similarity with human auditory characteristics and traditional refined hand-crafted feature extractors [33, 34]. In addition, Sailor, et al., [33] investigated the difference of center frequencies among models that were trained by both clean and noisy speech. They reported that the center frequency of learned filters do not show consistency between clean speech and noisy speech, suggesting that the optimal properties of filterbanks depend on the task and target environments. Zhu, et al., [31] also presented a model to learn features directly from waveforms and performed convolution operations with several types of window sizes and stride parameters to push past the inherent trade-off between temporal and frequency resolutions. These DNN-based systems eliminate the feature extraction stage and significantly improve the recognition performance. Earlier works reported the difference of filter characteristics caused by the condition of training data. However, since a system can not identify test speaker and test environment in advance, there is a mismatch between input test speech and learned model which causes a performance degradation. Therefore, adaptation remains a major challenge for DNN-based systems, which must alleviate the mismatch and recover recognition performance. In practical use, it is preferable for low-level feature extractor to track various test conditions. The model adaptation is a promising approach for the alleviation of this mismatch problem. Earlier works on model adaptation, which update sub-modules of neural network architecture, introduce various restriction methods [10, 13, 14, 15, 16, 36] and regularization methods [37] for prevention of an over-fitting problem. However, there is a trade-off between complexity and expressiveness. In other words, the update of a large number of parameters causes over-fitting problem or requirement of much adaptation data, and extreme restriction (low expressiveness) makes it difficult to adapt to given adaptation data. Therefore, it is important to maximize expressiveness while minimizing the number of free parameters to robustly update the system using a small size adaptation data.

1.3 Thesis summary

In this research, we focus on the rapid adaptation of the ASR system based on a small size of adaptation data and propose two adaptation methods.

Firstly, we propose an auxiliary feature based adaptation technique targeting recognition of a short time utterance which is suitable for current situations of recognition system, e.g., command recognition and voice search, by focusing on data-clustering and cluster-dependent modeling of ASR system. For this purpose, we cluster training data and train GMMs using each speech data cluster. Then, we define a speaker representation as a set of similarities between an input speech and the GMMs, and use it as an auxiliary feature. For an evaluation targeting the recognition of short time utterance, a duration of the speech for estimation of the proposed speaker representation is limited to a first 50 frames ($\sim$ 0.5 second) of the utterance. This method is categorized to the auxiliary feature based adaptation.

Secondly, we propose a new model adaptation technique, filterbank incorporated DNN, by incorporating a physiologically-motivated model into the deep neural network based ASR system. The introduced filterbank layer and the following neural networks of the proposed model are trained jointly by exploiting the advantages of the hierarchical feature extraction of DNN, while most current systems use pre-defined mel-scale filterbank features as its input. In addition, the introduced filterbank layer is parameterized to represent speaker characteristics while minimizing a number of parameters. Furthermore, introducing restrictions resulting from a physiologically motivated model protects the neural network (filterbank layer) module from extreme deterioration. The optimization of one type of parameters corresponds to Vocal Tract Length Normalization (VTLN) [38], and another type corresponds to feature-space Maximum Linear Likelihood Regression (fMLLR) [39] and feature-space Discriminative Linear Regression (fDLR) [10]. Therefore, it is considered that our method is advantageous in adaptation under limited adaptation data and it prevents extreme deterioration or overfitting problem. This method is categorized to both the feature adaptation and the model adaptation.

Our approach, rapid adaptation of ASR systems, further improve user experience in terms of data collection procedure and recognition performance, and will lead to further popularization of ASR systems under various surrounding environments.

1.4 Thesis organization

The thesis is organized as follows:

- **Chapter 2** describes an outline of speech recognition and introduces base of ASR system including acoustic feature, GMM-HMM hybrid systems, language model, and WFST decoder. We also introduce deep neural network based ASR systems including DNN-HMM hybrid system and attention-based encoder decoder networks.

- **Chapter 3** describes an auxiliary feature based adaptation technique and evaluate its effectiveness for recognition of short time utterance.
- **Chapter 4** presents a filterbank incorporated DNN and evaluates our method as data-driven feature extractor. Next, we apply this method to speaker adaptation and compare with other model adaptation methods. We also investigate relation between the learned filter shapes and physical characteristics of speakers by applying gender adaptation.
- **Chapter 5** extends our model adaptation technique targeting the filterbank layer to an end-to-end attention-based encoder decoder networks, and evaluate the proposed method for noise adaptation. We also compare the proposed method with other conventional adaptation methods.
- **Chapter 6** concludes the thesis and suggests future research directions.

Chapter 2

Outline of speech recognition

In this chapter, we first define the speech recognition problem in Section 2.1 and review the commonly used approaches including GMM-HMM, DNN-HMM, and end-to-end encoder decoder networks. Section 2.2 describes the acoustic features commonly used for the speech recognition system. Section 2.3 introduces an acoustic model, GMM-HMM hybrid system, and Section 2.4 introduces a language model, N-gram language model. Deep neural network and its extension to the hybrid system, DNN-HMM, are described in Section 2.5. Section 2.6 describes a WFST decoder, Section 2.7 describes an end-to-end speech recognition system, and Section 2.8 describes an evaluation metric for speech recognition. We conclude this chapter by reviewing freely available open source ASR system and projects in Section 2.9.

2.1 Definition of speech recognition problem

A speech recognition problem is defined as an estimation of the most probable label sequence $\hat{\mathbf{y}}$ given an input speech feature $\mathbf{x} = (x_1, ..., x_T)$ consists of T-frames, and is defined as:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \tag{2.1}$$

where $\mathbf{y} = (y_1, \cdots, y_l, \cdots, y_L)$ is the output target label sequence and $\mathbf{y}$ is the words in vocabulary $(y_n|n = (1, \cdots, L)) \in \mathcal{V}$. In case of hybrid system, $P(\mathbf{y}|\mathbf{x})$ in Eq. (2.1) is factorized by the Bayes' theorem as:

$$\begin{aligned} \hat{\mathbf{y}} &= \arg\max_{\mathbf{y}} \frac{P(\mathbf{x}|\mathbf{y})P(\mathbf{y})}{P(\mathbf{x})} \\ &\approx \arg\max_{\mathbf{y}} P(\mathbf{x}|\mathbf{y})P(\mathbf{y}). \end{aligned} \tag{2.2}$$

The term $P(\mathbf{x}|\mathbf{y})$ is called an acoustic model and $P(\mathbf{w})$ is called a language model. In general, a modeling unit of the acoustic model is phoneme and the modeling unit of the language model is a word. Let $\boldsymbol{\Psi}$ be the modeling unit of the acoustic model. Then, the Eq. (2.2) leads to:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \sum_{\boldsymbol{\Psi}} P(\mathbf{x}|\boldsymbol{\Psi})P(\boldsymbol{\Psi}|\mathbf{y})P(\mathbf{y})$$

In the following, Section 2.2 describes the acoustic feature $\mathbf{x}$, Sections 2.3 and 2.4 describe the acoustic model and the language model, respectively.

2.2 Speech analysis

Feature engineering is one of the main research topics and is crucial for the speech recognition performance. Acoustic feature for GMM-HMM includes Mel-Frequency Cepstrum Coefficients (MFCC) [40], Perceptual Linear Prediction (PLP) [41], and PNCC [42]. In deep learning era, neural network based feature extraction from low-level speech signal is mainstream and is trained to minimize ASR-related objective function [30, 31, 33, 34, 35]. [*1] Many deep learning based systems use filterbank feature as its input. In this section, we first describe filterbank feature in Section 2.2.1 followed by MFCC feature in Section 2.2.2.

2.2.1 Filterbank feature

Firstly, speech signal is converted to digital signal. In this paper, we use 16kHz sampling frequency and 16 sampling bit rate unless otherwise noted. Speech signal has a large amplitude at lower frequency region and small amplitude at higher frequency region. Therefore, pre-emphasis is applied to the signal by taking a first-order difference equation. Let $s_t \in (s_t|t = 1, \cdots T)$ be the speech signal at time t and s'_t be the speech signal after pre-emphasis. Then, the pre-emphasis is defined as:

$$s'_t = s_t - k^{\text{emp}} s_{t-1},$$

where k^{emp} is the pre-emphasis coefficients, and we used $k^{\text{emp}} = 0.97$ in this paper.

Then, the signal is divided into a sequence of frames by applying a window function which can be regarded as a stationary signal (quasi-stationary signal). An N-point

[*1] We describe the details in Chapter 4.

Hamming window, one commonly used window function, is defined as:

$$w_n = 0.54 - 0.46\cos(\frac{2\pi n}{N-1}), \quad n = 0, \cdots, N-1.$$

N-dimensional amplitude spectrum at frame t is obtained by applying fast Fourier transform (FFT) to the windowed N-point speech signal:

$$\begin{aligned} x_{t,k} &= \sum_{n=0}^{N-1} s'_{t+n} w_n e^{-i2\pi k \frac{n}{N}}, \quad k = 1, \cdots, N, \\ x_t &= (x_{t,k} | k = 1, \cdots, N) \end{aligned}$$

Finally, filterbank feature is obtained by applying weighted sum between the obtained amplitude spectrum and mel-scale triangular filterbank. The triangular filters are deployed according to the mel-scale defined as:

$$\mathrm{Mel}(f) = 1127.0 \ln(\frac{f}{700.0} + 1.0),$$

where f is the linear-scale frequency. The L-dimensional filterbank feature is calculated as:

$$m_{t,l} = \sum_{k=f^{\text{low}}}^{f^{\text{high}}} W_{k,l} |x_{t,k}|, \quad l \in (1, \cdots, L)$$

where $W_{k,l}$ is the k-th dimensional weight of l-th filter. The l-th filter takes positive values at certain region ranging from k_l^{low} to k_l^{high}:

$$W_{k,l} = \begin{cases} \frac{k - k_l^{\text{low}}}{k_l^c - k_l^{\text{low}}} & \{k_l^{\text{low}} \leq k \leq k_l^c\} \\ \frac{k_l^{high} - k}{k_l^{\text{high}} - k_l^c} & \{k_l^c \leq k \leq k_l^{\text{high}}\}, \end{cases}$$

where k_l^c is the center frequency of the l-th filter, k_l^{low} and k_l^{high} are the lower bound and upper bound spectum channel number, respectively. These values satisfy:

$$k_{l-1}^{high} = k_l^c = k_{l+1}^{low},$$

and center frequencies, $(k_l^c | l = 1, \cdots, L)$, are evenly spaced along the mel-scale.

2.2.2 MFCC

Mel-Frequency Cepstral Coefficients (MFCC) is obtained by further applying the Discrete Cosine Transform (DCT) to the filterbank feature:

$$c_{t,k} = \sqrt{\frac{2}{N}} \sum_{l=1}^{L} m_{t,l} \cos\Big(\frac{\pi i}{N}(l - 0.5)\Big)$$

Typically, the low-dimensional representations (e.g., 13 dimensions) are selected as a final representation to retain spectrum envelop, and the other high-dimensional parts are discarded. The (time-domain) values obtained by performing discrete cosine transform towards spectrum is called cepstrum.

2.2.3 Delta feature

Derivatives of the acoustic feature are added with the basic statistical feature, e.g., MFCC, targeting improvement of speech recognition performance. In the case of HMM Toolkit (HTK) [43], the dynamic feature called delta coefficients at time frame t is computed as:

$$d_{t,k} = \frac{\sum_{\theta=1}^{\Theta} \theta(c_{t+\theta,k} - c_{t-\theta,k})}{2\sum_{\theta=1}^{\Theta} \theta^2}$$

where Θ is the window size set to 2. Acceleration coefficients are obtained by applying same computation targeting the delta coefficients.

2.3 Acoustic model

2.3.1 Hidden Markov model

Definition of HMM

A hidden Markov model (HMM) is one approach to model time series data and is defined as Non-deterministic Finite Automaton (NFA). A structure of the HMM is depicted in Figure 2.1. A set of HMMs are used to model labels, e.g., phonemes and syllables, and each HMM consists of multiple components as follows:

- Set of finite number of states: $\{S_i|(i \in \{i \in \mathbb{N} : 1 \leq i \leq N\}\}$ where N is the number of states.
- Transition probability from an i-th state S_i to a j-th state S_j: a_{ij}
- Set of probability distributions of acoustic feature for each state: $\Big\{P(c|S_i) = b_i(c)\Big\}$ where c is the acoustic feature, e.g., MFCC.

An acoustic feature characteristics of phonemes change depending on preceding and succeeding phonemes but not only depending on a phoneme at current position. For the detailed modeling, typical HMM uses (context-dependent) triphone as its modeling unit which concatenating the preceding and succeeding phonemes. However,

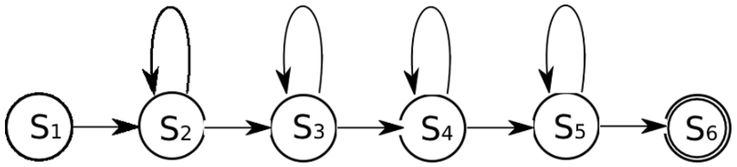


Fig. 2.1 Structure of HMM

the number of triphones is three power of the number of phonemes, and it leads to data-sparsity problem. Therefore, state tying is applied to reduce the number of triphone states [44, 45]. In the case of syllable-based HMM, a preceding vowel is concatenated to the syllable as a left-context [46]. A set of the unique HMM states is called senones.

Parameter estimation of HMM

The Baum-Welch algorithm [47] updates the HMM parameters using the Expectation-Maximization (EM) algorithm to find the maximum likelihood. Let M^{HMM} be a set of HMM parameters. Then the objective is defined as:

$$\underset{M^{\text{HMM}}}{\arg\max} P(c|M^{\text{HMM}}) = \underset{M^{\text{HMM}}}{\arg\max} \frac{P(z, c|M^{\text{HMM}})}{P(z|c; M^{\text{HMM}})},$$

where z is the latent variable representing the state transitions. Since there are multiple combinations of state transitions for a given observation sequence, the log likelihood is estimated by calculating the expected values for all transition state paths. The EM algorithm repeats update of the HMM parameters. Let $M^{\text{HMM}'}$ be the HMM parameters before update and M^{HMM} be the model parameters after the update. Then, the log-likelihood is defined as:

$$\log P(c|M^{\text{HMM}}) = \log \sum_z \Big\{ P(z|c, M^{\text{HMM}'}) \frac{P(z, c|M^{\text{HMM}})}{P(z|c, M^{\text{HMM}})} \Big\},$$

and the Jensen's inequality gives:

$$\begin{aligned} \log P(c|M^{\text{HMM}}) &\geq \sum_z \Big[P(z|c, M^{\text{HMM}'}) \log \frac{P(c, z|M^{\text{HMM}})}{P(z|c, M^{\text{HMM}})} \Big] \\ &= Q(M^{\text{HMM}}, M^{\text{HMM}'}) - \sum_z \{ P(z|c, M^{\text{HMM}'}) \log P(z|c, M^{\text{HMM}}) \} \end{aligned} \tag{2.3}$$

$$\text{where} \quad Q(M^{\text{HMM}}, M^{\text{HMM}'}) = \sum_z P(z|c, M^{\text{HMM}'}) \log P(c, z|M^{\text{HMM}}). \tag{2.4}$$

Eq. (2.3) is further re-formulated as:

$$\begin{aligned}\log P(c|M^{\mathrm{HMM}}) - \log P(c|M^{\mathrm{HMM}'}) &= Q(M^{\mathrm{HMM}}, M^{\mathrm{HMM}'}) - Q(M^{\mathrm{HMM}'}, M^{\mathrm{HMM}'}) \\ &\quad + \mathrm{KL}(P(z|c, M^{\mathrm{HMM}'}), P(z|c, M^{\mathrm{HMM}})) \\ &\geq Q(M^{\mathrm{HMM}}, M^{\mathrm{HMM}'}) - Q(M^{\mathrm{HMM}'}, M^{\mathrm{HMM}'}),\end{aligned}$$

where $\mathrm{KL}(\cdot, \cdot)$ is the Kullback-Leibler (KL) divergence defined as:

$$\mathrm{KL}(p, q) = \sum_i p_i \log \frac{p_i}{q_i} \geq 0.$$

When we assume the parameter M^{HMM} maximizes the Q-function in Eq. (2.4), it leads to:

$$\begin{aligned}Q(M^{\mathrm{HMM}}, M^{\mathrm{HMM}'}) &\geq Q(M^{\mathrm{HMM}'}, M^{\mathrm{HMM}'}), \\ P(c|M^{\mathrm{HMM}}) &\geq P(c|M^{\mathrm{HMM}'}).\end{aligned}$$

Therefore, the maximization of $P(c|M^{\mathrm{HMM}})$ can be replaced as the maximization of $Q(M^{\mathrm{HMM}}, M^{\mathrm{HMM}'})$. The Baum-Welch algorithm repeats the following two steps, Expectation-step and Maximization-step, to estimate the HMM parameters:

E-step Calculation of expected log-likelihood: $Q(M^{\mathrm{HMM}}, M^{\mathrm{HMM}'})$.

M-step Calculation of M^{HMM} that maximizes $Q(M^{\mathrm{HMM}}, M^{\mathrm{HMM}'})$.

Computation of acoustic score

Forward algorithm [48] calculates the most probable symbol sequence path with its probability as Algorithm 1. Let $\alpha(i, t)$ be the probability to generate observations c up to time t at state i:

$$\alpha(i, t) = P(c_1, c_2, \cdots, c_t, q_t = i|M^{\mathrm{HMM}}),$$

where q_t is the state at time t. Then, $\alpha(\cdot, \cdot)$ can be defined recursively as:

$$\alpha(j, t+1) = \sum_i^N \alpha(i, t) a_{i,j} b_j(o_{t+1}). \tag{2.5}$$

Finally, the probability to generate observation sequence c under the HMM M^{HMM} is represented as:

$$P(o|M^{\mathrm{HMM}}) = \sum_i^N \alpha(i, T).$$

The forward algorithm updates the accumulated probability α by considering all the states at the previous time step. Replacement of the summation operation $\sum$ in Eq. (2.5) to a max operation leads to the Viterbi algorithm [49, 48]. Algorithm 2 shows the Viterbi algorithm. The most probable path is estimated by saving the computation history and back-tracking the candidate states which corresponds to lines (10, 14-15) in the Algorithm 2.

Algorithm 1 Forward algorithm

```
$N \leftarrow$ number of states
$\pi_i \leftarrow$ initial state
$T \leftarrow$ length of observation sequence
for $i = 1$ to $N$ do
    $\alpha(i, 0) = \pi_i$
end for
for $t = 1$ to $T$ do
    for $j = 1$ to $N$ do
        $\alpha(j, t+1) = \sum_{i \in (1, \cdots N)} \alpha(i, t) a_{i,j} b_j(c_{t+1})$
    end for
end for
$P(c|M^{\mathrm{HMM}}) = \sum_{i \in (1, \cdots N)} \alpha(i, T)$
```

2.3.2 GMM-HMM

The GMM-HMM hybrid system models the state emission probability of HMM states, $b_i(c) = P(c|S_i)$, using Gaussian Mixture Model (GMM). Let n^{mix} be the number of mixtures, $\mu_{i,n}$ and $\Sigma_{i,n}$ be the n-th dimensional mean vector and covariance matrix of the state i, respectively. Then, the state emission probability of state i is represented as:

$$b_i(c) = \sum_{n=1}^{n^{\mathrm{mix}}} \alpha_n^{\mathrm{GMM}} \mathcal{N}(c; \mu_{i,n}, \Sigma_{i,n}), \tag{2.6}$$

$$\sum_{n=1}^{n^{\mathrm{mix}}} \alpha_n^{\mathrm{GMM}} = 1.0$$

$$\mathcal{N}(c; \mu_{i,n}, \Sigma_{i,n}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_{i,n}|^{\frac{1}{2}}} \exp\Big(-\frac{(c - \mu_{i,n})^T \Sigma_{i,n}^{-1} (c - \mu_{i,n})}{2} \Big),$$

Algorithm 2 Viterbi algorithm

```
$N \leftarrow$ number of states
$\pi_i \leftarrow$ initial state
$T \leftarrow$ length of observation sequence
for $i = 1$ to $N$ do
    $\alpha(i, 0) = \pi_i$
end for
for $t = 1$ to $T$ do
    for $j = 1$ to $N$ do
        $\alpha(j, t+1) = \max_{i \in (1, \cdots N)} \alpha(i, t) a_{i,j} b_j(c_{t+1})$
        $\zeta(j, t+1) = \underset{i \in (1, \cdots N)}{\arg\max}\, \alpha(i, t) a_{i,j} b_j(c_{t+1})$
    end for
end for
$P(o|M^{\mathrm{HMM}}) = \sum_i^N \alpha(i, T)$
$q_T^* = \underset{i \in (1, \cdots N)}{\arg\max}\, \zeta(i, T)$
$q_T^* = \zeta(q_{t+1}^*, t+1)$
```

where α_n^{GMM} is the weight of the n-th Gaussian component, and d is the number of dimension of the observation acoustic feature c. Figure 2.2 depicts an example of the GMM-HMM hybrid system. The combination with the HMM enables modeling of the observation acoustic feature sequence as a sequence of Gaussian mixture distribution by switching the states of HMM.

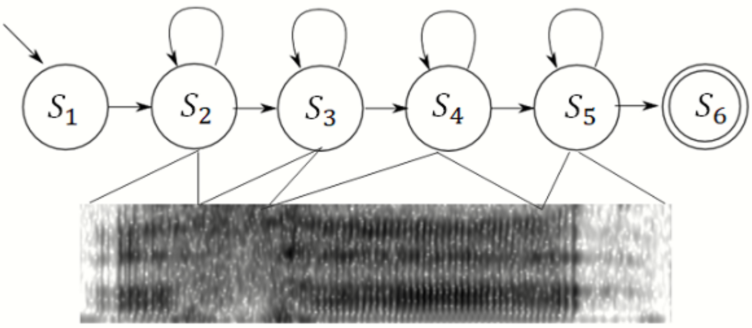


Fig. 2.2 Modeling of time series data.

2.4 Language model

2.4.1 Definition of N-gram language model

N-gram language model (LM) [50] models a probability to generate a word sequence $w = (w_1, \cdots, w_i, \cdots, w_K)$ consists of K words by assuming $(N-1)$-order Markov chain:

$$\begin{aligned} p(w) = p(w_1, \cdots, w_i, \cdots, w_K) &= \prod_{i=1}^{K} P(w_i | w_{i-N+1}, \cdots, w_{i-2}, w_{i-1}) \\ &= \prod_{i=1}^{K} p(w_i | w_{i-N+1}^{i-1}), \\ P(w_i | w_{i-N+1}, \cdots, w_{i-1}) &= \frac{c(w_{i-N+1}, \cdots, w_i)}{c(w_{i-N+1}, \cdots, w_{i-1})}, \end{aligned}$$

where $c(w_{i-N+1}, \cdots, w_i)$ is the frequency that the word sequence $(w_{i-N+1}, \cdots, w_i)$ is appeared in the training text data. When the number N is large, it is considered that the N-length long word sequences do not appear in the training data. In this situation, these probabilities should not be set as 0 and is necessary to give an appropriate probability mass. There are several techniques including interpolation and back-off smoothing. In the case of interpolation, the probability is approximated by using low-order N-grams:

$$p(w_i | w_{i-N+1}^{i-1}) = \sum_{k=1}^{K} \lambda_k^{\mathrm{LM}} p(w_i | w_{i-k+1}^{i-1}),$$

where λ_k^{LM} is the interpolation parameters. The back-off smoothing technique approximates probabilities for word sequences where $c(\cdot) = 0$. In the case of Witten-bell smoothing [51, 52], the N-gram probabilities are represented as:

$$P(w_i | w_{i-N+1}^{i-1}) = \begin{cases} \frac{c(w_{i-N+1}^{i})}{c(w_{i-N+1}^{i-1}) + r(w_{i-N+1}^{i-1})} & \text{if } \ c(w_{i-N+1}^{i}) > 0, \\ \frac{r_{i-N+1}^{i-1}}{c(w_{i-N+1}^{i-1}) + r(w_{i-N+1}^{i-1})} \alpha P(w_i | w_{i-N+2}^{i-1}) & \text{if } \ c(w_{i-N+1}^{i-1}) > 0, \\ P(w_i | w_{i-N+2}^{i-1}) & \text{otherwise,} \end{cases}$$

where r_{i-N+1}^{i-1} is the number of different vocabulary words which appear next to the word w_{i-N+1}^{i-1}.

2.4.2 Evaluation of N-gram language model

Perplexity (PP) [53] is one method to evaluate the language models which represents a number of candidate words in a probabilistic form given a word history. Let $w_1^N = (w_1, w_2, \cdots, w_n, \cdots, w_N)$ be the word sequence consists of N words. In the case of 3-gram language model, the perplexity is defined as:

$$\begin{aligned}\mathrm{PP} &= P(w_1, \cdots, w_N)^{-\frac{1}{N}} \\ &= \Big[\prod_{n=1}^{N} P(w_n|w_{n-2}^{n-1})\Big]^{-\frac{1}{N}}\end{aligned}$$

and the log-domain perplexity is:

$$\begin{aligned}\log_2 PP &= -\frac{1}{N}\log_2 P(w_1, \cdots, w_N) \\ &= -\frac{1}{N}\sum_{n=1}^{N}\log_2 P(w_n|w_{n-2}^{n-1})\end{aligned}$$

The vocabulary of the generated recognition result is restricted by a vocabulary of collected text data. Words outside the vocabulary are called out of vocabulary (OOV) words. The OOV words are replaced to a unique symbol and modeled as $P(\text{“UNK''}|w_{n-2}^{n-1})$. For comparison of N-gram LMs of different vocabulary size, adjusted PP (APP) [54] discounts the score of “UNK” by a number of unknown vocabulary size as:

$$\begin{aligned}\log_2 \mathrm{APP} &= -\frac{1}{N}\Big\{\log_2 P(w_1, \cdots, w_N) - o\log_2 m\Big\} \\ &= -\frac{1}{N}\Big\{\sum_{n=1}^{N}\log_2 P(w_n|w_{n-2}^{n-1}) - o\log_2 m\Big\}\end{aligned}$$

where o is the observed number of OOV words and m is the number of different OOV words.

2.5 Neural networks for hybrid-base speech recognition system

2.5.1 Outline of neural network

The deep neural network (DNN) composed of multiple hidden layers. Figure 2.3 shows an example of the DNN with 2 hidden layers. Let $x \in (x_1, \cdots, x_t, \cdots, x_T)$

be the D-dimensional input feature of length T and $r \in (r_1, \cdots, r_t, \cdots, r_T)$ be the T-length V-dimensional output label sequence. Then, the j-th hidden unit at l-th hidden layer, h_j^l, is calculated as:

$$\begin{aligned} o_j^l &= \sum_i w_{j,i}^l h_i^{l-1} + b_j^l, \\ h_j^l &= f(o_j^l), \end{aligned}$$

where $w_{j,i}^l$ is the connection weight of the i-th unit at $(l-1)$-th layer and the j-th unit at l-th layer, and b_j^l is the j-th bias at l-th layer. In the following sub-sections, we discard time index t for simplicity. The function f is an activation function, including sigmoid function:

$$f(x) = \frac{1}{1+e^{-x}},$$

rectified linear unit (ReLU) [55, 56]:

$$f(x) = \max(0, x),$$

hyperbolic tangent function, and maxout function [57, 58]. In the case of DNN for the hybrid system, posterior probabilities of senones are estimated at the last layer. Let L be the number of layers of the DNN. Then, the posterior probability of the k-th label is calculated as

$$\begin{aligned} p(y_k|x) &= \text{softmax}(h^{L-1}) \\ &= \frac{\exp(h_k^{L-1})}{\sum_i^V \exp(h_i^{L-1})}. \end{aligned} \tag{2.7}$$

In summary, the network is trained to output hypotheses as:

$$y = \text{DNN}(x; \theta^{\text{DNN}})$$

where x and y are the input and output feature and θ^{DNN} is the set of parameters of the DNN. Then a *loss* value is calculated by comparing with a corresponding correct reference r, and the parameters of the DNN are optimized to minimize the loss in a supervised manner as:

$$\begin{aligned} l &= \text{Loss}(y, r), \\ \theta^{\text{DNN}} &\leftarrow \underset{\theta^{\text{DNN}}}{\arg\min}\, l. \end{aligned}$$

In the early deep learning era, unsupervised pre-training methods for the DNN are studied to give good initial values which bring performance improvement at the

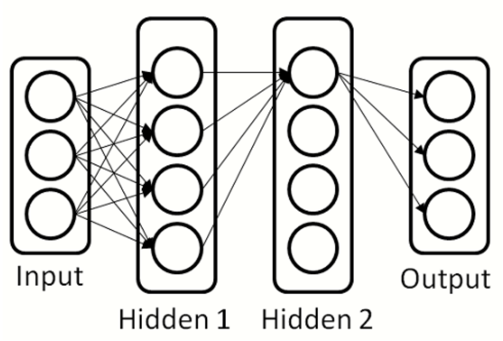


Fig. 2.3 Structure of multi-layer fully-connected neural network.

following supervised training stage [59]. In the following section, we first describe one pre-training method, contrastive divergence, in Section 2.5.2. We describe the supervised training method in Section 2.5.3 followed by the DNN-based hybrid system DNN-HMM in Section 2.5.4.

2.5.2 Unsupervised training of deep neural network

The Contrastive Divergence (CD) [60] algorithm first estimates parameters of bottom two layers by regarding the sub-network of the DNN as Restricted Boltzmann Machine (RBM). The parameters are connection weights between visible nodes v (at the first layer) and hidden nodes h (at the second layer), and biases of both visible and hidden nodes. There is no connection within the visible units and the hidden units, respectively as shown in Figure 2.4. After the estimation of bottom first and second layers, parameters of the second and the third layers are estimated in the same manner and repeats the parameter estimation.

As the acoustic features take real values, the bottom RBM consisting of the first and the second layer is trained as Gaussian-Bernoulli RBM. The following RBMs are trained as Bernoulli-Bernoulli RBM. Let $v^{\mathrm{rbm}} \in \mathbb{R}^{|V|}$ $(V = \{1, \cdots, |V|\})$ be the

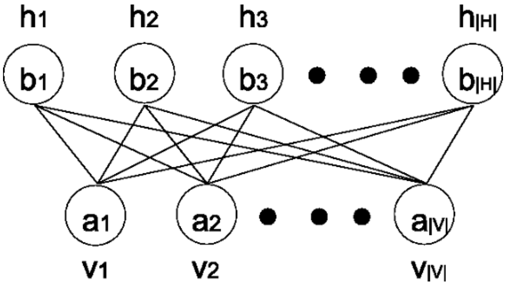


Fig. 2.4 Structure of Restricted Botlzmann Machine.

$|V|$-dimensional acoustic feature vector, and $h^{\text{rbm}} \in \mathbb{R}^{|H|}$ $(H = \{1, \cdots, |H|\})$ be the $|H|$-dimensional hidden vector.

Then an energy function of the Gaussian-Bernoulli RBM is defined as:

$$E(v^{\text{rbm}}, h^{\text{rbm}}) = \sum_{i \in V} \frac{(v_i^{\text{rbm}} - a_i^{\text{rbm}})^2}{2} - \sum_{j \in H} b_j^{\text{rbm}} h_j^{\text{rbm}} - \sum_{i \in V, j \in H} v_i^{\text{rbm}} h_j^{\text{rbm}} w_{i,j}^{\text{rbm}},$$

and the energy function of the Bernoulli-Bernoulli RBM is defined as:

$$E(v^{\text{rbm}}, h^{\text{rbm}}) = -\sum_{i \in V} a_i^{\text{rbm}} v_i^{\text{rbm}} - \sum_{j \in H} b_j^{\text{rbm}} h_j^{\text{rbm}} - \sum_{i \in V, j \in H} v_i^{\text{rbm}} h_j^{\text{rbm}} w_{i,j}^{\text{rbm}}.$$

Conditional probabilities of the visible node and the hidden node for the Gaussian-Bernoulli RBM are calculated as:

$$\begin{aligned} p(v_i^{\text{rbm}} = v | h^{\text{rbm}}) &= N(v | a_i^{\text{rbm}} + \sum_j h_j^{\text{rbm}} w_{i,j}^{\text{rbm}}, 1), \\ p(h_j^{\text{rbm}} = 1 | v) &= \sigma(b_j^{\text{rbm}} + \sum_i v_i^{\text{rbm}} w_{i,j}^{\text{rbm}}), \end{aligned}$$

where $\sigma(x)$ is the sigmoid function defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$

In case of the Bernoulli-Bernoulli RBM, the conditional probabilities are defined as:

$$\begin{aligned} p(v_i^{\text{rbm}} = 1 | h^{\text{rbm}}) &= \sigma(a_i^{\text{rbm}} + \sum_j h_j^{\text{rbm}} w_{i,j}^{\text{rbm}}), \\ p(h_j^{\text{rbm}} = 1 | v^{\text{rbm}}) &= \sigma(b_j^{\text{rbm}} + \sum_i v_i^{\text{rbm}} w_{i,j}^{\text{rbm}}). \end{aligned}$$

Given the set of RBM parameters: $\theta = (w^{\text{rbm}}, a^{\text{rbm}}, b^{\text{rbm}})$, a log-likelihood of the RBM is then defined as:

$$\begin{aligned} \ln p(v^{\text{rbm}} | \theta) &= \ln \frac{1}{Z} \sum_{j \in H} e^{-E(v^{\text{rbm}}, h^{\text{rbm}})} \\ &= \ln \sum_{j \in H} e^{-E(v^{\text{rbm}}, h^{\text{rbm}})} - \ln \sum_{i \in V, j \in H} e^{-E(v^{\text{rbm}}, h^{\text{rbm}})}, \end{aligned}$$

and the partial differentiation of the log-likelihood with regard to the parameter θ_k is represented as:

$$-\frac{\partial p(v^{\text{rbm}} | \theta)}{\partial \theta_k} = < \frac{\partial E}{\partial \theta_k} >_{\texttt{data}} - < \frac{\partial E}{\partial \theta_k} >_{\texttt{model}} .$$

In the case of weight $w_{i,j}^{\text{rbm}}$, it leads to:

$$-\frac{\partial p(v^{\text{rbm}}|\theta)}{\partial w_{i,j}^{\text{rbm}}} =< v_i^{\text{rbm}} h_j^{\text{rbm}} >_{\texttt{data}} - < v_i^{\text{rbm}} h_j^{\text{rbm}} >_{\texttt{model}} .$$

The first term $< v_i^{\text{rbm}} h_j^{\text{rbm}} >_{\texttt{data}}$ is estimated by the input acoustic feature and the conditional probability $p(h_j^{\text{rbm}} = 1|v)$. Since the computation of $< v_i^{\text{rbm}} h_j^{\text{rbm}} >_{\text{model}}$ is intractable, it is commonly approximated using sampling algorithm, e.g., Gibbs sampling [61].

After the parameter estimation of the bottom two layers, the parameters of the bottom layers are fixed and the above two layers, i.e., the second and third layers, are trained as the RBM. The input to the second layer is $p(h^{\text{rbm}} = 1|v^{\text{rbm}})$ which is calculated at the previously estimated RBM(s).

2.5.3 Supervised training of deep neural network

Backpropagation (BP) algorithm [62, 63] is one supervised learning method for training of the DNN. The algorithm computes the prediction y given an input speech sample x and updates parameters to correctly output reference label r. The parameters of DNN are optimized to minimize an *error* defined by a loss function. A typical ASR task is categorized to a classification task, and a cross entropy loss is employed in many cases. The cross entropy loss is defined as:

$$E^{\text{CE}} = -\sum_{k}^{|V|} \Big\{ r_k \log(p(y_k|x)) \Big\}.$$

A derivative of the cross entropy loss E^{CE} with regard to the parameter $w_{j,i}^l$ is:

$$\begin{aligned} \frac{\partial E^{\text{CE}}}{\partial w_{j,i}^l} &= \frac{\partial E^{\text{CE}}}{\partial h_j^l} \frac{\partial h_j^l}{\partial w_{j,i}^l} \\ &= \delta_j^l o_i^{l-1} \quad \text{where} \begin{cases} \delta_j^l = h_j^l - r_j & \text{if } l = L, \\ \delta_j^l = o_j^l(1 - o_j^l) \sum_k (\delta_k^{l+1} w_{k,j}^{l+1}) & \text{otherwise.} \end{cases} \end{aligned} \tag{2.8}$$

The parameters of DNN are updated to decrease the error as:

$$w_{j,i}^l \leftarrow w_{j,i}^l - \eta \delta_j^l o_i^{l-1},$$

where η is the learning rate which controls contribution of the calculated derivative. More sophisticated optimization methods are proposed, e.g., AdaGrad [64], AdaDelta [65], and Adam [66].

Glorot, et al., [55] and Tóth [56] reported that an usage of a rectified linear unit (ReLU) enables elimination of the pre-training procedure without degradation of speech recognition performance. In the case of DNN-based hybrid system, we followed these research and used the ReLU activation function without the pre-training procedure for the training of DNN-HMM hybrid system to reduce training time. The usage of ReLU function replaces Eq. (2.8) to:

$$\delta_j^l o_i^{l-1} \quad \text{where} \begin{cases} \delta_j^l = h_j^l - r_j & \text{if } l = L, \\ \delta_j^l = \max(0, \sum_k (\delta_k^{l+1} w_{k,j}^{l+1})) & \text{otherwise.} \end{cases}$$

2.5.4 DNN-HMM hybrid system

The acoustic model computes the posterior probability $P(\mathbf{x}|\boldsymbol{\Psi})$ as described in Section 2.1. This is factorized as:

$$\begin{aligned} P(\mathbf{x}|\boldsymbol{\Psi}) &= \frac{P(\boldsymbol{\Psi}|\mathbf{x})P(\mathbf{x})}{P(\boldsymbol{\Psi})} \\ &\approx \frac{P(\boldsymbol{\Psi}|\mathbf{x})}{P(\boldsymbol{\Psi})} \end{aligned}$$

The probability $P(\mathbf{x})$ is discarded because of independence from the state transition. The $P(\boldsymbol{\Psi})$ is computed by calculating frequencies of labels of the training data. In the framework of hybrid system, the DNN is trained to predict the posterior probability of states $\boldsymbol{\Psi}$ given the input acoustic feature $\mathbf{x}$. The probability estimated by the GMM as in Eq. (2.6) is replaced by the DNN as in Eq. (2.7).

2.6 WFST Decoder

Weighted Finite State Transducer (WFST) [67] based decoding is one framework to search hypotheses by integrating the acoustic model and the language model. Finite State Transducer (FST) consists of a finite input symbol set, a finite output symbol set, a finite state set, a state transition function, and an initial and a final state set. A transition state is determined by the current state and its input. WFST can define a weight with regard to the state transition in FST and can compute cost along the state transitions. Hybrid systems, i.e., GMM-HMM and DNN-HMM, use the HMM, pronunciation dictionary, and the language model. These models are represented as WFSTs, and are combined to one WFST using a composition operation. When the HMM is modeled as a context-dependent form, an additional WFST C is introduced for a conversion to the context-dependent form. When we define a WFST for a conversion from characters to a word as L, a WFST for a conversion from a word to word sequence for N-gram LM as G, and a WFST from a conversion from the acoustic feature sequence to the modeling unit of the HMM as H, the WFSTs are combined to one WFST as:

$$H \circ C \circ L \circ G,$$

where $\circ$ is the composition operation. In addition to the composition operation, the WFST supports operations including determinization operation (for an elimination of ambiguity) and minimization operation (for minimization of state number) for efficient hypothesis search.

2.7 End-to-end ASR system

2.7.1 Attention-based encoder decoder networks

Attention-based encoder decoder networks [68] consist of three modules, an encoder network, a decoder network, and an attention network. Let $Y = (y_1, \cdots, y_N)$ be the label sequence generated by the encoder decoder networks, $R = (r_1, \cdots, r_N)$ be the reference label sequence, and $O = (o_t \in \mathbb{R}^D | t = 1, \cdots, T)$ be the T-frame sequence of D-dimensional input feature vector. The encoder network takes the input feature vector and converts to an L-frame sequence of C-dimensional high-level representation $H = (h_l \in \mathbb{R}^C | l = 1, \cdots, L)$ as:

$$H = \mathrm{Encoder}(O).$$

Typical encoder networks consist of a stack of bi-directional long short-term memory (BLSTM) [69, 70]. The length of the input feature vector and the hidden vector is different when the encoder network takes sequence length reduction techniques, including convolutional operation [6], max-pooling [71], and sub-sampling [68, 72].

The decoder network repeats computation of posterior probabilities by taking a context vector c, and a previous label y_{n-1} generated by the decoder network. The posterior probability of y_n at decoding time step n is defined as:

$$p^{\mathrm{att}}(y_n | O, y_{1:n-1}) = \mathrm{Decoder}(c_n, y_{n-1}),$$

where the context vector is calculated by the attention network which takes the hidden representation H, an attention weight a, and a hidden state e_n:

$$\begin{aligned} c_n, a_n &= \text{Attention}(a_{n-1}, e_n, H), \\ e_n &= \text{Update}(e_{n-1}, c_{n-1}, L_{n-1}). \end{aligned}$$

The posterior probability for generation of the label sequence is defined as:

$$\begin{aligned} p^{\text{att}}(Y|O) &= \prod_{n=1}^{N} p^{\text{att}}(y_n|O, y_{1:n-1}) \\ &= \prod_{n=1}^{N} \text{Decoder}(c_n, y_{n-1}) \end{aligned}$$

At training stage, the previous label history $y_{1:n-1}$ is replaced to the reference label history $r_{1:n-1}$ in a teacher-forcing fashion for efficient training.

For the models including Decoder($\cdot$), Attention($\cdot$), and Update($\cdot$), many researchers proposed various types of network architectures. In this section, we describe detailed modules which are employed in an open source project ESPnet [73] and are also used in this thesis. The context vector is calculated as a weighted sum between the hidden representation and the attention weight as:

$$c_n = \sum_{l}^{L} a_{n,l} h_l,$$

where n is the index of label sequence and l is the index of the hidden representation. The attention weight is calculated as:

$$\begin{aligned} a_{n,l} &= \frac{\exp(\alpha k_{n,l})}{\sum_{l=1}^{L} \exp(\alpha k_{n,l})}, \\ k_{n,l} &= w^T \tanh(V^E e_{n-1} + V^H h_l + V^F f_{n,l} + b), \\ f_n &= F * a_{n-1}, \end{aligned}$$

where w, V^E, V^H, V^F, b, F are the weight parameters, and α is a constant value. The operation $*$ is a convolution operation. The hidden state e is calculated as:

$$\begin{aligned} e_n &= \text{Update}(e_{n-1}, c_{n-1}, y_{n-1}) \\ &= \text{LSTM}(\text{Lin}(e_{n-1}) + \text{Lin}(c_{n-1}) + \text{Emb}(y_{n-1})), \end{aligned}$$

where LSTM($\cdot$) is the LSTM module, Lin($\cdot$) is the linear layer, and Emb($\cdot$) is the embedding layer which converts a label index to a fixed size dense vector.

2.7.2 Sequence-to-sequence frameworks

The attention-based encoder decoder networks directly model the relation between the input acoustic feature and the label sequence in a sequence-to-sequence manner without a conditional independence assumption. In addition to the encoder decoder networks, many studies proposed end-to-end architectures, e.g., Connectionist Temporal Classification (CTC) [74, 75, 5], recurrent neural network transducer [76], and transformer [77]. Hori, et al., proposed a method to train the attention-based encoder decoder networks described in Section 2.7 and CTC jointly as multi-task learning to encourage the attention network to generate monotonic alignment [6]. This method shares the encoder network for the extraction of high-level representation, and the CTC module is added for an encouragement of monotonic alignment. When we define the loss of encoder decoder networks and CTC as $\mathcal{L}_{\text{att}}$ and $\mathcal{L}_{\text{CTC}}$, the final loss is defined as

$$\mathcal{L}_{\text{mtl}} = \lambda_{\mathcal{L}}\mathcal{L}_{\text{att}} + (1 - \lambda_{\mathcal{L}})\mathcal{L}_{\text{CTC}}$$

where $\lambda_{\mathcal{L}}$ is the interpolation factor.

2.8 Evaluation of speech recognition

Word error rate (WER) is one metric to measure the performance of the ASR systems by taking the generated hypothesis word sequence and ground-truth reference word sequence. The WER is calculated based on the Levenshtein distance because the length of the reference word sequence and the hypothesis word sequence is different. Let $\#\mathrm{w}^{\mathrm{err}}$ be the number of error words consisting of substitution errors ($\#\mathrm{w}_{\mathrm{S}}^{\mathrm{err}}$), deletion errors ($\#\mathrm{w}_{\mathrm{D}}^{\mathrm{err}}$), and insertion errors ($\#\mathrm{w}_{\mathrm{I}}^{\mathrm{err}}$), and let $\#\mathrm{w}_{\mathrm{corr}}$ be the number of correct words and $\#\mathrm{w}$ be the number of words in the reference. Then, the WER is defined as:

$$\begin{aligned}\mathrm{WER} &= \frac{\#\mathrm{w}^{\mathrm{err}}}{\#\mathrm{w}} \\ &= \frac{\#\mathrm{w}_{\mathrm{S}}^{\mathrm{err}} + \#\mathrm{w}_{\mathrm{D}}^{\mathrm{err}} + \#\mathrm{w}_{\mathrm{I}}^{\mathrm{err}}}{\#\mathrm{w}_{\mathrm{S}}^{\mathrm{err}} + \#\mathrm{w}_{\mathrm{D}}^{\mathrm{err}} + \#\mathrm{w}_{\mathrm{C}}^{\mathrm{err}}}.\end{aligned} \tag{2.9}$$

The error rate is defined using other modeling unit other than the word. CER (Character Error Rate) and PER (Phoneme Error Rate) are alternative metrics for the measurement of ASR performance. The CER and PER are defined by using Eq. (2.9) with the change of counting units to character and phoneme, respectively.

2.9 Summary

This chapter described the definition of speech recognition problem and general approaches to building a speech recognition system. There are freely available toolkits to build automatic speech recognition systems. The Hidden Markov Model Toolkit (HTK) [43] supports the training of GMM-HMM hybrid systems. The Kaldi toolkit supports the training of GMM-HMM and DNN-HMM hybrid systems and WFST based decoding. ESPnet (End-to-End Speech Processing Toolkit) [73] and EESEN [78] are active projects developing the end-to-end speech recognition system.

Chapter 3

Rapid speaker class adaptation using speaker information

3.1 Introduction

Difference of acoustic condition among training data and test data, e.g., environment and recording interface, is one crucial factor for degradation of speech recognition performance in real environments. Several approaches have been proposed to address this problem. Normalization of the acoustic feature is one of the methods for a suppression of acoustic difference in transfer characteristics [79, 23]. Another approach is class-dependent modeling of acoustic feature which aims at modeling multiple acoustic models depending on several conditions by splitting data into some clusters [22]. The training data is split into multiple clusters by measuring a similarity of speech, and is defined as speaker's characteristics including vocal tract parameters [24], eigen voice [25], speaking rate [26], i-vector [27], and so on. Class-dependent modeling is deeply studied by targeting the GMM-HMM hybrid system in the above research.

The trained multiple ASR systems are also used as an ensemble for improvement of generalization ability [80, 81, 82]. Tan, et al., proposed a method to construct speaker-dependent DNN by computing a weighted sum of multiple DNN model parameters as an extension of cluster adaptive training (CAT) [81]. Kosaka, et al., incorporated multiple DNN-HMM systems at a level of posterior probability generated by the DNNs. These systems control contribution of each model using a specific metric which can measure a similarity of an input speech (for evaluation) and the clusters (for training). This similarity is defined as a distance between speaker characteristics. Some studies proposed methods to add these speaker representations with conventional acoustic features [18, 19, 20]. Hamid, et al., proposed an usage of speaker-code [18], and Huang, et al., proposed an usage of bottleneck feature [20] as an auxiliary feature. However, the earlier works assume the availability of speech from several tens seconds to several minutes, and are difficult to apply for the recognition of short time utterance. Tsujikawa, et al., proposed a method to estimate i-vector from a short time utterance. However, they also reported that it is difficult to estimate robust speaker characteristics from a short time speech, i.e., 0.5 second [28]. Lie, et al., investigated a relation between speech duration for i-vector estimation and recognition performance, and reported that more than 5.0 seconds are required for the improvement of speech recognition performance [29]. They also reported that the recognition performance got worse when the acoustic characteristics in the training data do not cover that of the input speech (in the evaluation) [29]. Therefore, it is difficult to apply the conventional auxiliary feature based speaker adaptation technique to the recognition of short time utterance.

In this chapter, we first apply cluster-dependent modeling technique to the DNN-HMM hybrid system, which was proposed by our group for a GMM-HMM hybrid system [83] and investigate its effectiveness. Next, we investigate a method to integrate the similarity measure used for data-clustering with the conventional acoustic feature to handle class-dependent clustering information within the DNN model. The proposed system is evaluated as an ASR system for the recognition of short time utterance targeting voice command recognition and voice search system. For this purpose, we restrict an available time period for the estimation of speaker information to 0.5 second and regard it as an auxiliary feature.

This chapter is organized as follows: We first review conventional feature normalization methods in Section 3.2. Section 3.3 describes the proposed method, clustering of speech data and its cluster-dependent modeling. Section 3.4 describes experimental setup and results.

3.2 Normalization of acoustic feature

Let $\mathbf{c} = (c(1), c(2), \cdots, c(t), \cdots, c(T))$ be a sequence of cepstrum feature consisting of T frames, e.g., MFCC, and $c(t) = (c_1(t), c_2(t), \cdots, c_D(t))$ be a set of dimensions of cepstrum feature at time frame t. These methods first computes mean, $\mu = (\mu_1, \mu_2, \cdots, \mu_D)$, and variance, $\sigma^2 = (\sigma_1^2, \sigma_2^2, \cdots, \sigma_D^2)$, of the cepstrum feature using training data per each dimension:

$$\mu_i = \frac{1}{T}\sum_{t=1}^{T} c_i(t) \tag{3.1}$$

$$\sigma_i^2 = \frac{1}{T}\sum_{t=1}^{T}(c_i(t) - \mu_i)^2 \tag{3.2}$$

Cepstral mean normalization (CMN) transforms the i-th dimensional cepstrum feature at frame t as:

$$\hat{c}_i(t) = c_i(t) - \mu_i, \tag{3.3}$$

and Cepstral variance normalization (CVN) transforms the i-th dimensional cepstrum feature at frame t as:

$$\hat{c}_i(t) = \frac{c_i(t)}{\sqrt{\sigma_i^2}}. \tag{3.4}$$

Combination of Eqs. (3.3) and (3.4) results in Cepstrum Mean and Variance Normalization (CMVN) [79] as:

$$\hat{c}_i(t) = \frac{c_i(t) - \mu_i}{\sqrt{\sigma_i^2}}. \tag{3.5}$$

These normalization methods are employed to suppress the mismatch of acoustic features, and they are also applied to suppress the mismatch among speakers. However, these normalization methods need to compute statistical values, Eqs. (3.1) and (3.2). In other words, when the system waits for an end of the utterance to calculate the statistical values, this constraint leads to a delay in the speech recognition process. In contrast, a restriction of usable frames leads to an inaccurate statistical estimation because of unbalanced appearance of phonemes. Pujol, et. al., [84] proposed an online CMVN that updates the statistical values successively along the input streaming speech.

$$\begin{aligned}
\mu_i(t) &= \beta\mu_i(t-1) + (1-\beta)c_i(t) \\
\sigma_i^2(t) &= \beta\sigma_i^2(t-1) + (1-\beta)(c_i(t) - \mu_i(t))^2, \\
\beta &= 0.992.
\end{aligned}$$

Nakano, et al., [85] proposed a class dependent CMVN. This method first splits training data into multiple clusters and statistical values are calculated per each class. They modeled a distribution of the cepstral features by using the GMM with respect to each class. Then, a test utterance (for evaluation) is classified to the nearest class using the first short (e.g., 50) frames and the set of GMMs, and the CMVN is applied using the mean and variance of the selected class.

3.3 Reduction of acoustic feature variation

3.3.1 Acoustic feature clustering

This section describes an algorithm to cluster speech data. The purpose of data clustering is a suppression of acoustic diversity within clusters. At an initialization step, a number of classes M is defined and the training data is randomly split to M classes. Then, initial GMMs are trained by using the speech data of each class. At a clustering step, we employed a soft-clustering which allows an assignment of a single utterance into multiple classes to prevent a decrease of data size of each class as shown in Algorithm 3.

Let u_i be an i-th utterance in the training data $\mathcal{U} = (u_1, \cdots, u_i, \cdots, u_{|\mathcal{U}|})$, and n

Algorithm 3 Clustering of speech data

```
for utterance $u_i \in \mathcal{U}$ do
    $n = 1$
    Calculate likelihood $sc_{i,m}$ between $u_i$ and $m \in (1, \cdots, M)$ for each model
    Sort classes in descending order with regard to scores $(sc_{i,m} | m \in (1, \cdots, M))$
      $(1, \cdots, m_{min}, \cdots, m_{max}, \cdots, M)$
    for $j = 2$ to $M$ do
        if $sc_{(1)} - sc_{(j)} < rs$ then
            $n = n + 1$
        end if
    end for
    if $n < m_{min}$ then
        Assign $u_i$ to $1, \cdots, m_{min}$
    else if $n > m_{max}$ then
        Assign $u_i$ to $1, \cdots, m_{max}$
    else if $m_{min} \leq n \leq m_{max}$ then
        Assign $u_i$ to $n$ classes.
    end if
end for
```

be the number of classes for assignment. The algorithm calculates the likelihood sc between the utterance u_i and a set of GMMs:

$$\lambda = (\lambda_m | m \in (1, \cdots, M))$$

as:

$$\begin{aligned} sc_{i,m} &= \text{Likelihood}(u_i, \lambda_m) \\ &= \log p(u_i|\lambda_m) \\ &= \sum_{t=1}^{T} \log p(u_i(t)|\lambda_m) \end{aligned}$$

where m is the index of M classes and T is the length of input acoustic feature. Then, the M classes are sorted in descending order with regard to the corresponding likelihoods, and it is used to decide the number of classes for assignment (n) at lines 5-9 in the algorithm. This algorithm assigns the utterance u_i to class j when a difference between the likelihood of most probable class and that of j-th class is smaller than a threshold value rs. In other words, the training data size is controlled by rs, and we set rs with the constraint that the data size in each class exceeds that of the initial class. Constant values m_{min} and m_{max} are also introduced at lines 10-16 to restrict a range of n as $m_{min} <= n <= m_{max}$ to prevent extreme assignment. After the execution of clustering algorithm, the GMMs are retrained by using the generated clusters, and a ratio of fluctuated utterances is calculated for each class. This procedure is repeated until the ratio is saturated. The ratio of fluctuation is defined by the number of the fluctuated utterances divided by the total number of utterances in the training data.

3.3.2 Class dependent modeling

In the case of class dependent modeling of ASR systems, the clusters generated by the algorithm described in Section 3.3.1 are used. In the evaluation stage, the most probable model is selected by calculating the set of likelihoods between the input speech and the M GMMs. The likelihood is defined as:

$$\begin{aligned} sc_{i,m} &= \log p(o|\lambda_m) \\ &= \sum_{t=1}^{T} \log p(o_t|\lambda_m), \end{aligned} \tag{3.6}$$

where o_t is the acoustic feature at time frame t and T is the available time frame. The available time frame T is set to 50 in the experiment.

3.3.3 Class dependent feature normalization

In the case of class dependent feature normalization [85], the statistics are calculated using each cluster. Eq. (3.5) is replaced as:

$$\hat{c}_i(t) = \frac{c_i(t) - \mu_{i,m}}{\sqrt{\sigma^2_{i,m}}}$$

where $m \in (1, \cdots, M)$ is the m-th cluster. As class dependent modeling in Section 3.3.2, the statistics are selected based on Eq. (3.6).

3.3.4 Addition of auxiliary speaker class information

The above section focuses of the selection of the most probable class for the employment of class dependent statistics and class dependent acoustic models. However, it is considered that the DNN has an ability to model the complex distribution of acoustic features because of its expressive power, and class-specific transformation is modeled within the DNN automatically. Therefore, we investigate methods to input speaker information with the conventional acoustic feature, and further compare various input types of speaker information.

Figure 3.1 shows an overview of our proposed method. In the training stage, we first cluster the training data on the basis of acoustic feature similarity as described in Section 3.3.1. After the clustering, a set of GMM likelihoods are calculated between

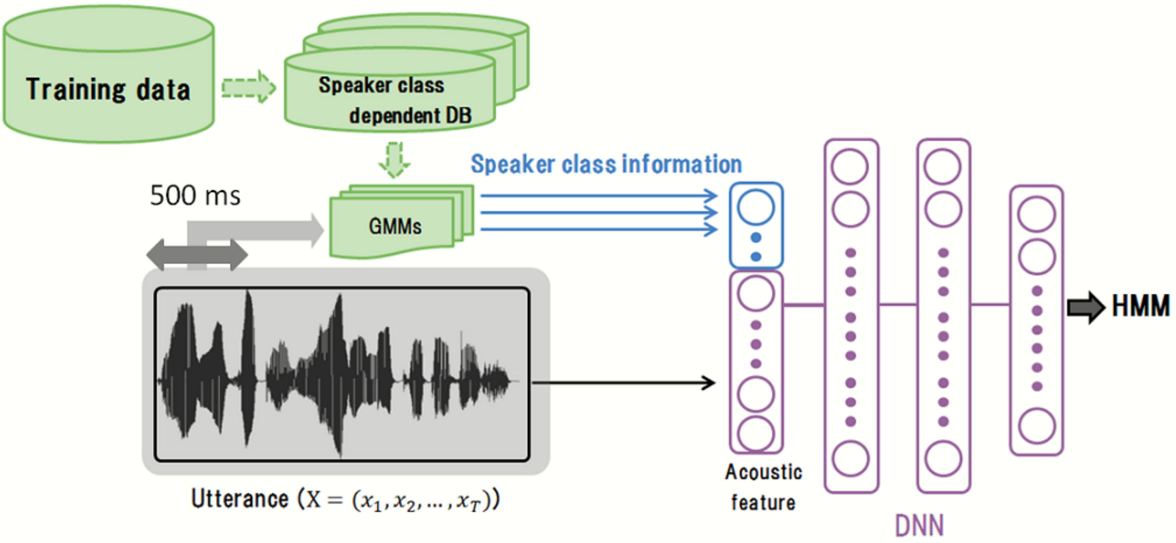

Training data
Speaker class dependent DB
Speaker class information
500 ms
GMMs
HMM
Utterance (X = $(x_1, x_2, \ldots, x_T)$)
Acoustic feature
DNN

Fig. 3.1 Overview of speaker class incorporation.

the GMMs and the input utterance. We regarded each cluster as speaker-class, and defined a set of likelihoods as “speaker-class information” which represents speaker characteristics. This speaker class information is fed into the DNN as a speaker information. In other words, a number of units added to the DNN corresponds to the number of clusters. The speaker class information of i-th utterance sc_i is defined as:

$$\begin{aligned} sc_i &= (cs_{i,m} | m \in (1, \cdots, M)) \\ sc_{i,m} &= \log p(o|\lambda_m) \\ &= \sum_{t=1}^{T} \log p(o_t|\lambda_m). \end{aligned}$$

The available time period is set to the length of input speech in the training stage, and it is set to 50, i.e., 0.5 second, in the evaluation stage.

In addition to the input of likelihoods, we investigated various types of speaker class information as follows:

a. Input of speaker class information as an auxiliary feature
   (1) Log likelihood: A set of likelihoods between the input utterance and the class-dependent GMMs are calculated and they are fed into the DNN with the acoustic feature.
   (2) Estimated class: In order to investigate whether the function of CMVN can be performed inside the DNN, we used a one-hot vector by assigning 1 to the most probable class and zeroing out the other $M-1$ speaker classes as a one-of-M representation.
   (3) Posterior probability distribution: Log-likelihood varies on the utterance with high variance and there is a possibility for unstable speaker class modeling. This method applies a softmax operation on the likelihoods for re-scaling of the range of likelihoods.

b. Speaker class dependent CMVN: Most probable speaker class is calculated by computing the likelihoods between the input speech and the class-dependent GMMs. The CMVN is applied by the selected class-dependent statistics and the normalized feature is input to the DNN-HMM. For the purpose of rapid selection, 50 frames of the utterance are used for the selection of speaker class.

c. Combination of speaker class information and class dependent CMVN: This method combines both the input of speaker class information and the normalization of acoustic feature based on the class dependent CMVN.

3.3.5 Stepwise training

As reported in the following experiment, when training the DNN using the high-dimensional acoustic feature and the low-dimensional speaker class information, the DNN mainly focuses on the optimization with regard to the acoustic feature, and makes it difficult to take the low-dimensional speaker class information into account for the DNN. Therefore, we trained the DNN using two-step procedure aiming at a stable training of the DNN, and additional parameter search based on the auxiliary speaker class information. First, the DNN is trained by using only the acoustic feature while the speaker class information is set to 0.0. Then the DNN is retrained by using both the acoustic feature and the speaker class information. We called it stepwise training.

3.4 Experimental work

3.4.1 Corpus

To ensure an age- and gender-independent speech recognition system, we used three types of corpora consisting of adult speakers, elder speakers, and child speakers, summarized in Table 3.1. The database used for an adult class is ASJ+JNAS [86, 87] database consisting of 133 male and 164 female speakers aged 18 to 59. This corpus consists of 20,337 sentences (≈33 hours) and 25,056 sentences (≈44 hours) uttered by males and females, respectively. The database for an elder class is S-JNAS [88] database consisting of 151 male and 150 female speakers aged 60 to 90. This corpus consists of 24,081 sentences (≈53 hours) and 24,061 sentences (≈53 hours) uttered by males and females, respectively. The database for a child class is CIAIR-VCV [89] database consisting of 140 male and 138 female speakers aged 6 to 12. This corpus consists of 7,538 sentences and 3,993 words (≈11 hours) and 7,744 sentences and 3,910 words (≈11 hours) uttered by males and females, respectively. In the CIAIR-VCV corpus, the child class was mainly composed of speech obtained from a reading of fairy tales. However, a language model we used in the experiment was trained by newspapers. As a result, the child class's out-of-vocabulary rate (OOV) was high as 13.8 and 13.6% for male and female respectively, while the rates for the elder class and the adult class were 0.5% and 2.1%, respectively. In the following experiment, we refer to these initial speaker class as "6-class-init". Each corpus contains male and female speech data. Therefore we divided the data into six (initial) classes: speakers of adult male speakers (SAM), speakers of adult female speakers (SAF), speakers of elder male speakers (SEM), speakers of elder female speakers (SEF), speakers of child male speakers (SCM), speakers of child female speakers (SCF).

Test data for each class was 100 sentences. The number of speakers in the test data for SAM and SAF were 23 speakers. The number of speakers for SEM and SEF were 10 speakers, and SCM and SCF were 7 and 8 speakers, respectively.

The average number of frames per one utterance was 540 frames. Although our aim is to recognize a short utterance, there was no appropriate test set to evaluate speaker adaptation/recognition on the short utterance. Therefore, we restrict the available frames to the first 50 frames of the utterance and calculate the speaker class information and the most probable class from the trimmed data.

Table. 3.1 Dataset for rapid speaker class adaptation.

AS+JNAS		
Gender	Male (SAM)	Female (SAF)
Age	18-59	18-59
# speakers	133	164
# utterances	20,337($\approx$ 33h)	25,056($\approx$ 44h)
OOV	0.45%	0.45%
S-JNAS		
Gender	Male (SEM)	Female (SEF)
Age	60-90	60-90
# speakers	151	150
# utterances	24,081($\approx$ 53h)	24,061($\approx$ 53h)
OOV	2.07%	2.05%
CIAIR-VCV		
Gender	Male (SCM)	Female (SCF)
Age	6-12	6-12
# speakers	140	138
# utterances	7,538(+3993) $\approx$ 11h	7,744(+3910) $\approx$ 11h
OOV	13.81%	13.64%

3.4.2 Acoustic feature and acoustic model

(i) Acoustic feature

The speech was analyzed using a 25-ms Hamming window with a pre-emphasis coefficient of 0.97 and shifted with a 10-ms frame advance. All acoustic models, GMM-HMM and DNN-HMM, and GMMs for speaker classification are trained by using 12 MFCCs along with their first and second derivatives and the first and second derivatives of the logarithm power (38-dims.) extracted by Hidden Markov Model Toolkit (HTK) [43]. In the case of DNN-HMM, the input MFCCs of ±5 frames are stacked to take context information into account.

(ii) Syllable-based modeling

The basic unit of Japanese is syllable and there are 116 context-independent syllables in total. In this study, we used left context (vowels and pause: a, i, u, e, o, N, qs, silence), which leads to 928 left context-dependent syllables in total [46]. Each HMM has four states, and the number of output units increases to 3,712 (=928 × 4). To reduce the number of output units, we used tied 3 state syllables (TC3), which tied the latter three states of the syllable.

Figure 3.2 shows an example of the left-context dependent syllable and TC3. In this example, the left context is "a" and the syllable is "ka". In the case of TC3, the last three states are tied and the the syllable (without preceding left context) share same symbol, i.e., TC. If the latter three states are tied, only the first state is a left context-dependent syllable (or states) and the others consist of the context-independent syllables (or state). Therefore, the number of output

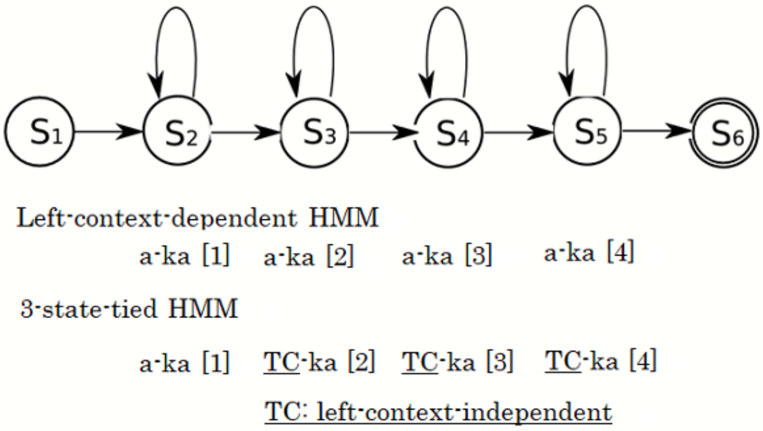


Fig. 3.2 Left context dependent syllable based HMM.

units becomes 1,276 ($= 1 \times 928 + 3 \times 116$).

(iii) GMM-HMM

The 928 context-dependent syllable-based HMMs were trained using the EM estimation algorithm. In the case of speaker class dependent GMM-HMM hybrid system, each HMM has four states and each distribution was represented with 32 mixture diagonal Gaussians. In the case of speaker class independent monolithic GMM-HMM hybrid system, the distribution of each HMM was set to 128 mixture diagonal Gaussians. The GMM-HMMs were trained by using HTK [43].

(iv) DNN-HMM

The acoustic features were normalized to zero mean and unit variance using the training data except for the speaker class dependent CMVN. The training targets were obtained by applying force alignment using the corresponding tied 3 state context-dependent syllable-based GMM-HMM. To reduce the training time of the DNNs, the DNNs were fine-tuned by using a rectified linear unit as an activation function [55, 56] and skipped the pre-training procedure [60]. As a preliminary experiment, we trained the DNN without pre-training with the usage of the rectified linear unit and the DNN with contrastive-divergence based pre-training with the usage of sigmoid function, and these two models showed comparable results. The DNN has 3 hidden layers with 2,048 units and an output layer with 1,276 ($= 928 + 116 \times 3$) units. The initial parameters of DNN were set sampled from the following uniform distribution [90]:

$$w \sim U\Big[-\frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}\Big],$$

where n_j and n_{j+1} are the number of units of j-th layer and $(j+1)$-th layer.

(v) Language model and WFST decoder

As a language model, a tri-gram word-based language model was trained on the Mainichi newspaper corpus collected from January 1991 to September 1994 and from January 1995 to June 1997 (11,533,739 words, vocabulary of 20,000 words) [91]. The cut-off was set to 1, and Witten-Bell method was used for the computation of back-off. The perplexities of 6 classes were 125.7 (SAM), 125.7 (SAF), 129.4 (SEM), 129.4 (SEF), 293.0 (SCM), and 301.2 (SCF). As a decoder, we used the SPOJUS++ (SPOken Japanese Understanding System) WFST version [92].

(vi) GMM for speaker classification

The initial 6 GMMs (SAM, SAF, SEM, SEF, SCM, SCF) for speaker class classification were trained by using 10,000 utterances and the number of mixtures was set to 8. The GMMs were retrained by repeating the clustering algorithm and EM algorithm. We set the number of mixtures of the final GMMs to 64, and called it "6 class soft". In the experiment, the number of clusters was further increased to 12, and it was called "12 class soft". It was executed by randomly halving "6 class soft" and repeating clustering algorithm and GMM training. The threshold rs was set to 0.6, and m_{min} and m_{max} were set to 1 and 3, respectively.

3.4.3 Baseline class independent model

Table 3.2 shows WER of the baseline class-independent models which were trained using all training data. GMM-HMM showed an average WER of 13.0% and DNN-HMM showed an average WER of 11.2%. Our result indicates the advantage of DNN as with earlier works.

3.4.4 Class dependent models

Table 3.3 shows WERs obtained by training 6 GMM-HMMs and 6 DNN-HMMs using 6-class-init which corresponds to the original corpora in Table 3.1. In this experiment, one model among the 6 models has to be selected for the evaluation of each utterance. This table shows the result assuming that the correct class is known. When we focused on the GMM-HMM, class-dependent GMM-HMM obtained an average WER of 13.0%, and showed better performance than the monolithic GMM-HMM system in Table 3.2. Especially, child classes, i.e., SCM and SCF, obtained significant performance improvement. It is considered that there is an acoustic difference between the child class and other classes, and the reduction of acoustic variation brought the curated GMM-HMMs for child classes.

Table. 3.2 WER (%) of the baseline GMM-HMM and DNN-HMM hybrid systems.

Acoustic Model	SAM	SAF	SEM	SEF	SCM	SCF	Ave.
GMM-HMM	9.2	8.3	10.4	8.3	32.3	24.2	15.4
DNN-HMM	5.5	4.5	7.1	6.2	23.5	20.0	**11.2**

Table. 3.3 WER (%) of the class dependent models trained by 6-class-init (class-known).

Acoustic Model	SAM	SAF	SEM	SEF	SCM	SCF	Ave.
GMM-HMM	6.5	6.4	10.4	6.6	26.8	21.1	13.0
DNN-HMM	6.1	4.9	7.1	5.3	22.8	21.1	**11.2**

When we trained 6 DNN-HMMs, an average WER was 11.2%. Different from the GMM-HMM, there is no significant difference between the monolithic DNN-HMM in Table 3.2, and it is considered that the DNN-HMM is robust against the diversity of speakers. However, it should be noted that the 6-class-init (corpus) dependent modeling leads to a decrease in training data. This trade-off between the number of models and the available training data affects a generalization ability of the DNN.

In the case of GMM-HMM, the split of training data brought performance improvement. Therefore, we investigated further split of the training data and evaluated its performance. We clustered the training data to 12 clusters. The classification result of training data and test data are described in Table 3.4 and Table 3.5, respectively. Labels 1 to 12 represents each cluster name. As initial training data for 12 clusters, we randomly halved the "SAM" of 6 class initial and assigned to labels 1 and 2, and repeated same procedures for all other classes, SAF, SEM, SCM, and SCF. The labels 1, 2, 5, and 6 mainly consists of SAM and SEM, and the labels 3, 4, 7, and 8 mainly consists of SAF and SEF. The labels 9, 11, and 12 mainly consists of SCM and SCF, and the label 10 is the cluster that equivalently contains the utterances of all classes. We can see that there is an acoustical similarity between SAM and SEM, SAF and SEF (adult and elder speakers), and SCM and SCF (child speakers).

Table 3.6 shows the speech recognition performance. In this experiment, we assumed that the correct speaker class is unknown, and the speaker class for speech recognition is selected by calculating the likelihoods between the input speech and the GMMs for speaker classification. The likelihoods were calculated by using both the first 50 frames of an utterance (50 frames) and all the frames of an utterance (all frames).

In the case of 6-class-init, when the class is unknown and the available time period was 50 frames, the WER was 16.1% and it was worse than the condition where the corresponding class was known (15.4%). However, when all frames of an utterance were available, the WER was recovered to 14.1% and it was better than the class-known condition (15.4%). The selected classes were changed by restricting the

Table. 3.4 Result of 12-class clustering targeting training data using all frames of utterance (%).

	label 1	label 2	label 3	label 4	label 5	label 6	label 7	label 8	label 9	label 10	label 11	label 12
SAM	44.2	54.0	0.4	0.2	43.0	24.0	0.7	0.3	0.1	11.2	0.0	0.1
SAF	0.2	2.6	64.3	46.7	0.7	1.3	31.0	32.1	1.3	44.0	0.3	6.0
SEM	54.4	42.8	1.0	0.7	52.2	60.8	1.9	1.3	0.0	14.3	0.0	0.0
SEF	1.1	0.3	33.8	52.1	4.0	13.8	66.1	65.9	0.0	19.6	0.0	0.1
SCM	0.1	0.4	0.3	0.2	0.1	0.1	0.3	0.2	46.5	7.8	45.7	44.1
SCM	0.0	0.0	0.2	0.2	0.0	0.0	0.1	0.2	52.1	3.1	53.9	49.7
Sum.	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

Table. 3.5 Result of 12-class clustering targeting test data using all/50- frames of utterance (# utterances).

	label 1	label 2	label 3	label 4	label 5	label 6	label 7	label 8	label 9	label 10	label 11	label 12	Sum.
SAM	23/15	58/46	0/0	0/0	8/6	7/5	0/1	0/0	0/1	4/26	0/0	0/0	100/100
SAF	0/0	0/0	31/27	32/27	0/1	0/0	13/7	14/7	0/1	8/27	0/0	2/3	100/100
SEM	34/20	6/8	0/1	0/1	16/29	39/26	0/4	0/0	0/0	5/11	0/0	0/0	100/100
SEF	0/1	0/0	22/20	13/24	0/0	6/5	23/28	24/18	0/0	12/2	0/0	0/2	100/100
SCM	0/0	0/0	0/5	0/0	0/0	0/0	0/0	0/0	32/22	2/18	10/22	56/33	100/100
SCM	0/0	0/0	0/5	0/1	0/0	0/1	0/0	0/0	5/16	0/7	74/36	21/34	100/100

Table. 3.6 Increase of speaker-class and changes in WER (class-unknown).

Acoustic Model	Training data	Class estimation: all frames						
		SAM	SAF	SEM	SEF	SCM	SCF	Ave.
	6 class init (6 GMMs)	6.8	7.5	11.2	9.1	28.0	22.3	14.1
GMM-HMM	12 class soft (12 GMMs)	7.1	5.9	8.8	7.0	28.4	20.4	**12.9**
		Class estimation: 50 frames						
		SAM	SAF	SEM	SEF	SCM	SCF	Ave.
GMM-HMM	6 class init (6 GMMs)	8.3	10.2	17.2	8.2	28.9	23.9	16.1
	12 class soft (12 GMMs)	8.7	6.0	11.3	8.1	28.9	23.7	**14.4**

available time frames. Therefore, it is considered that the decrease in accuracy of the speaker class selection had an influence on the decrease of recognition performance. By increasing the speaker class to 12 clusters, the GMM-HMM systems obtained the average WER of 14.4% when 50 frames of an utterance were used for the speaker class selection, and the relative improvement of 6.5% was obtained from the baseline monolithic GMM-HMM (15.4%).

From these results, clustering of speech data based on the acoustic feature and speaker class dependent training of acoustic models are effective for the GMM-HMM hybrid system. However, in the case of DNN-HMM hybrid systems, the performance of class depending modeling and monolithic modeling was comparable. In the following section, we further evaluate robust DNN-HMM hybrid systems against the diversity of speakers by applying cepstrum normalization and auxiliary feature based model adaptation in Section 3.4.5 and Section 3.4.6, respectively.

3.4.5 Cepstrum normalization

(i) Speaker class dependent CMVN

Table 3.7 shows WERs of the DNN-HMM systems with speaker class dependent CMVN. The speaker class for CMVN was selected by using all frames or 50 frames of an utterance. By applying CMVN using 6 initial class, the WERs were 10.8% on all frames and 10.9% on 50 frames. We applied data clustering and generated 6 class soft and 12 class soft. When the speaker class was estimated by using all frames of an utterance, the average WERs were 11.4% on 6 class soft and 11.2% on 12 class soft. The best performance was obtained by applying 6 class init dependent CMVN, and the same result was obtained in the case of 50 frames. These results indicated that the increase of speakers class leads to performance degradation for DNN-HMM while it brings performance improvement for GMM-HMM.

(ii) Unit of CMVN

We conducted several comparisons by changing unit of CMVN as corpus, speaker, and utterance. Table 3.8 shows the average WER obtained under various CMVN unit conditions by assuming that the oracle classes (corpus, speaker, and utterance) are known. The corpus dependent CMVN showed an average WER of 11.1% and 6 class init based CMVN showed an average WER of 11.2%. These results were the same as the baseline DNN-HMM hybrid system in Table 3.2.

When we applied CMVN per utterance consuming all frames (≈ 540 frames) of an utterance, the average WER was 10.2%. However, the CMVN using 50 frames degraded the WER to 20.6%. We can see that it is difficult to estimate

Table. 3.7 WER (%) using speaker class dependent CMVN (class-unknown)

Acoustic Model	Training data	Class estimation: all frames						
		SAM	SAF	SEM	SEF	SCM	SCF	Ave.
DNN-HMM	1 class (Table 3.2)	5.5	4.5	7.1	6.2	23.5	20.0	11.2
	6 class init	5.7	4.4	6.5	5.5	23.3	19.5	10.8
	6 class soft	5.7	4.6	7.2	5.5	24.9	20.5	11.4
	12 class soft	5.3	4.7	7.5	4.9	24.6	20.3	11.2
		Class estimation: 50 frames						
		SAM	SAF	SEM	SEF	SCM	SCF	Ave.
DNN-HMM	1 class (Table 3.2)	5.5	4.5	7.1	6.2	23.5	20.0	11.2
	6 class init	5.5	4.3	6.5	5.3	23.4	20.2	**10.9**
	6 class soft	5.7	4.6	7.2	5.5	24.9	20.5	11.4
	12 class soft	5.5	4.7	6.8	4.9	24.4	20.9	11.2

Table. 3.8 Comparison of CMVN unit (class-known)

CMVN unit	# normalization unit	Ave. WER (%)
corpus	3	11.1
6 class init	6	11.2
speaker	81	10.6
utterance (all frames)	6×100	10.2
utterance (50 frames)	6×100	20.6

robust statistics for CMVN, and not applicable to the recognition of short time utterance.

(iii) Online CMVN

Table 3.9 shows the WER of online CMVN targeting the DNN-HMM hybrid system as described in Section 3.2. We employed two methods "Online" and "Batch" in the training stage. In the case of "Online", the statistical values, mean and variance for CMVN, were updated sequentially in an online manner by taking the statistical value of the previous frame. In the case of "Batch", CMVN was applied by assuming the input utterance is known and the statistical values are calculated before CMVN as an offline manner.

The WER was 10.7% in the case of "Online" and it was 12.3% in the case of "Batch". In general, online CMVN is applied only for the test data and it corresponds to the "Batch". This is due to a requirement of certain steps of statistical update from the statistics estimated by training data to the appropriate statistics for each specific input utterance. During this certain steps, CMVN was applied by using the inappropriate statistics. In the case of this experiment, "online" showed better performance than the baseline monolithic DNN-HMM system in Table 3.2.

We considered that the difference of corpora normalized by CMVN is limited, and a degree on how to normalize speaker characteristics per utterance appeared as a difference in recognition performance. We further investigate the evaluation of short-time utterance in Section 3.4.8.

3.4.6 Input of speaker information

This section evaluated the recognition performance of the auxiliary feature based adaptation technique by adding the speaker class information. The speaker class information was extracted by using all frames or 50 frames of an utterance and the extracted feature was fed into the DNN with the acoustic feature. Results without

Table. 3.9 Utterance level online cepstral normalization on DNN-HMM hybrid system.

Training	Test	Ave. WER (%)
Online	Online	10.7
Batch	Online	12.3

the stepwise training is shown in Table 3.10 and results with the stepwise (two steps) training is shown in Table 3.11.

(i) Without stepwise (two steps) training
When we focus on the 6-class-init with the usage of 50 frames, the average WER of the speaker-class-dependent CMVN was 10.9% as in Table 3.7, and the addition of speaker class information, input of likelihoods, was 10.8%. The

Table. 3.10 WER (%) using speaker class information without stepwise training (class-unknown)

Training method	Training data	Class estimation: all frames						
		SAM	SAF	SEM	SEF	SCM	SCF	Ave.
Baseline	1 class (Table 2)	-	-	-	-	-	-	-
w/o stepwise training	6 class init	5.3	4.3	6.7	5.2	23.1	19.9	10.8
	6 class soft	5.8	4.4	6.5	5.0	23.1	20.1	10.8
	12 class soft	5.1	4.7	7.3	6.2	23.0	19.1	10.9
Training method	Training data	Class estimation: 50 frames						
		SAM	SAF	SEM	SEF	SCM	SCF	Ave.
Baseline	1class (Table 2)	5.5	4.5	7.1	6.2	23.5	20.0	11.2
w/o stepwise training	6 class init	5.3	4.3	6.7	5.2	23.1	19.9	**10.8**
	6 class soft	5.8	4.4	6.5	5.0	23.1	20.1	**10.8**
	12 class soft	5.7	4.7	7.1	6.5	23.3	18.7	11.0

Table. 3.11 WER (%) using speaker class information with stepwise training (class-unknown)

Training method	Training data	Class estimation: all frames						
		SAM	SAF	SEM	SEF	SCM	SCF	Ave.
Baseline	1 class (Table 2)	-	-	-	-	-	-	-
w/ stepwise training	6 class init	5.5	4.1	6.2	4.9	21.9	20.6	10.6
	6 class soft	5.7	4.6	6.3	5.1	22.2	19.6	10.6
	12 class soft	5.1	4.4	6.4	4.6	22.0	20.0	10.4
Training method	Training data	Class estimation: 50 frames						
		SAM	SAF	SEM	SEF	SCM	SCF	Ave.
Baseline	1class (Table 2)	5.5	4.5	7.1	6.2	23.5	20.0	11.2
w/ stepwise training	6 class init	5.6	4.1	6.0	4.9	22.2	20.7	10.6
	6 class soft	5.6	4.1	5.9	4.6	22.2	20.2	**10.4**
	12 class soft	5.1	4.4	6.4	4.6	22.0	20.0	**10.4**

same performance was also obtained by using all frames of an utterance for the estimation of speaker class information, and these results were better than the baseline monolithic DNN-HMM hybrid system of 11.2% as in Table 3.2. These results showed that the extraction and incorporation of speaker class information provided better results than the monolithic 1-class DNN-HMM (11.2%) even if the available time period was only 50 frames. Additionally, we conducted a significance test between the monolithic DNN-HMM system without the use of speaker class information and the DNN-HMM with the addition of speaker class information defined by 6 class init. The significance test showed that our method was statistically significant at the 10% level ($p = 0.084$). When we combined the two methods, the speaker class dependent CMVN and the usage of speaker class information, the average WER was not improved. These results indicate that the combination of the two approaches does not provide a complementary function.

We also conducted experiments using 12 class soft to investigate whether the further increase of speaker class could lead the recognition performance improvement as same as the GMM-HMM system. The recognition performance of the DNN-HMM with the usage of 12 class soft based speaker class information showed almost the same WER as the 6 class init based DNN-HMM system, unlike the GMM-HMM system. The increase of speaker class could represent more detailed speaker information. On the other hand, it is considered that the increase of speaker class suffered the lack of training data to achieve better generalization.

(ii) With stepwise (two steps) training

We evaluated the effectiveness of stepwise training as shown in Table 3.11. When the available time period was 50 frames, the DNN-HMM hybrid system with the usage of 6 class soft based speaker class information showed the best performance and the average WER was 10.4%. This improvement was significant at the 1.0% level compared with the DNN-HMM with speaker class dependent CMVN in Table 3.7 (10.4 vs. 10.9%).

Next, we investigated the relation between the recognition performance and the updated layer during a second step of the stepwise training. DNN has the characteristic of hierarchical non-linear transformation, and it is considered that neural networks at the low-level layers are responsible for feature extraction. Based on this assumption, we retrained the first, second, or first and second layers during the second step of stepwise training. The results were described in Table 3.12. The update of the first layer and second layer obtained the average WERs of 11.0% and the update of the first and second layer obtained the average WER of 10.7%. However, these results were worse than the model with full layer update, and it showed the requirement of full layer update as a classifier but not feature extractor.

3.4.7 Comparison of speaker representation

This section evaluates various representation methods of speaker class information as described in Section 3.3.4. The speaker representations were estimated by using the first 50 frames of an utterance, and a set of likelihoods were calculated by using the GMMs of 6 class soft.

a. Input of speaker class information as an auxiliary feature
   (1) Log likelihood: The use of likelihood showed the average WER of 10.8% without the stepwise training and the use of stepwise training improved the average WER to 10.4%.
   (2) Estimated class: The use of estimated speaker class as one-hot vector representation showed the average WER of 12.2% without the stepwise training, and the use of stepwise training improved the average WER to 10.7%.
   (3) Posterior probability distribution: The use of posterior probabilities showed the average WER of 11.6% without the stepwise training, and the use of stepwise training improved the average WER to 10.6%.
b. Speaker class dependent CMVN: The average WER was 11.4%.
c. Input of speaker class information and CMVN: Simultaneous use of speaker

Table. 3.12 Relation between the updated layer at the second step and WER (%) (class-unknown).

Retrained layer	Class estimation: 50 frames						
	SAM	SAF	SEM	SEF	SCM	SCF	Ave.
all	5.6	4.1	5.9	4.6	22.2	20.2	10.4
1	6.2	4.3	6.2	5.7	23.2	20.6	11.0
2	5.7	4.7	6.5	4.8	23.7	20.5	11.0
1 and 2	6.3	4.4	6.2	4.8	22.7	19.6	10.7

class dependent CMVN and addition of speaker class information (log likelihood) obtained the average WER of 11.2% without the stepwise training and the stepwise training further decreased the average WER to 10.7%. The combinational use did not show performance improvement, and it can be seen that there is no complementary relation in both two methods.

These results supported the effectiveness of stepwise training, and there is no difference between the input of likelihoods and posterior probabilities in the case of stepwise training.

3.4.8 Evaluation using first one word

The purpose of this study is the recognition of short-time utterance. However, there is no speech database consisting of a few words uttered by various speakers including child speakers, adult speakers, and elder speakers. Therefore, we calculated WER by collecting the first words of the recognition results that we reported in the above section. Out-of-vocabulary ratio (OOV) of the first words was 0.0%, and the average duration of the first words was 0.88 second, which was measured by the GMM-HMM hybrid system's alignment. The results were shown in Table 3.13.

We can see that the input of speaker class information improved the recognition performance compared with the baseline DNN-HMM system. On the contrary, the WER of the online CMVN was worse than the baseline DNN because it is difficult to estimate robust statistics from 0.5 second which corresponds to approximately 2∼3 syllables. These results were obtained from only 6 classes × 100 utterances (total of 600 words). Therefore, there was no significant difference between the baseline and the proposed method, that is, the input of speaker class information with stepwise training, although 4 classes out of 6 classes showed performance improvement and other classes showed comparable performance to the baseline DNN-HMM system.

Table. 3.13 WER (%) calculated by first words of sentences.

method	SAM	SAF	SEM	SEF	SCM	SCF	Ave.
baseline (1class, 1DNN)	95	92	85	88	85	89	89.0
stepwise training (6 class soft, 50 frames)	95	92	87	91	87	90	90.3
online CMVN (training: online, recognition: online)	93	94	86	88	84	88	88.8
online CMVN (training: batch, recognition: online)	94	90	84	88	76	88	86.7

Chapter 4

Rapid speaker adaptation by neural network based feature extraction

4.1 Introduction

Deep neural networks (DNNs) have achieved significant success in the field of automatic speech recognition. One main advantage of DNNs is automatic feature extraction under a simple objective function without human intervention. Exploiting this property, some recent novel approaches focus on front-end learning based on DNNs that take low-level acoustic features [30, 31, 32, 33, 34, 35]. Chen, et al., [30] trained a neural network based on a multi-task training by integrating a speech enhancement module and a speech recognition module. Variani, et al., [32] proposed Complex Linear Projection (CLP) which can take complex spectrum into account.

Sainath, et al., [34] and Sailor, et al., [33] proposed an end-to-end model that uses raw waveforms and performs a frequency analysis. These studies reported that some of the learned characteristics showed a similarity with human auditory characteristics and traditional refined hand-crafted feature extractors [33, 34]. In addition, Sailor, et al., [33] investigated the difference of center frequencies among models that were trained by both clean and noisy speech. They reported that the center frequencies of the learned filters did not show consistency between the clean speech and the noisy speech, suggesting that the optimal properties of the filterbanks depend on the task and target environments. Zhu, et al., [31] also presented a model to learn features directly from waveforms and performed convolution operations with several types of window sizes and stride parameters to push past the inherent trade-off between temporal and frequency resolutions. These DNN-based systems eliminate the feature extraction stage and significantly improve the recognition performance.

Earlier works reported the difference of the filter characteristics caused by the condition of training data. However, since a system can not identify varying test speaker and test environment in advance, there are mismatches between the input speech (for evaluation) and the learned model which causes the recognition performance degradation. Therefore, adaptation remains a major challenge for the DNN-based systems, which must alleviate the mismatch and recover speech recognition performance. In practical use, it is preferable for a low-level feature extractor to track various test conditions rapidly using a small size adaptation data. In the case of model adaptation, parameters of the DNN are re-estimated by using the adaptation data. In this scenario, the trade-off between the size of the adaptation data and the number of parameters becomes a critical problem. In other words, too many parameters cause poor generalization and overfitting to the given data if the available adaptation data are limited.

In contrast to the DNNs, physiologically motivated models are composed of a small number of parameters. Therefore, the physiologically motivated model is advantageous in the model adaptation under limited adaptation data. Furthermore, introducing restrictions resulting from the physiologically motivated model explicitly protects the introduced filterbank layer from extreme deterioration. In this chapter, we propose a filterbank-incorporated DNN that incorporates a filterbank layer that presents the filter shape/center frequency and the DNN-based acoustic model. The filterbank layer and the following neural networks of the proposed model are trained jointly by exploiting the advantages of the hierarchical feature extraction, while most systems use pre-defined triangular mel-scale filterbank features as its input. Filters in the filterbank layer are parameterized to represent speaker characteristics while minimizing the number of free parameters. Since the filterbank layer consists of just a few parameters, it is advantageous in the adaptation under limited available adaptation data. We evaluate the advantage of filterbank-incorporated DNNs in speaker/gender adaptations as the model adaptation. The followings are the contributions of this work:

(i) proposed a filterbank-incorporated DNN and evaluated it as a speaker independent model;

(ii) evaluated our proposed model for speaker/gender adaptation and compared various filter types;

(iii) discussed the relation between the physical characteristics of speakers' vocal tracts and optimal filterbanks from an engineering viewpoint.

This chapter is organized as follows: We first describe earlier model adaptation techniques in Section 4.2. Section 4.3 describes the proposed neural network based filterbank layer. Section 4.4 summarizes the advantages of the proposed neural network filterbank layer and its expected behavior under adaptation of filter shapes. Experimental setup and results are described in Section 4.5.

4.2 Earlier works on model adaptation

In this section, we first review model adaptation techniques. We also discuss introduced constraints for some adaptation techniques from a viewpoint of matrix multiplication. The model adaptation is a promising method to adapt the DNN that updates its parameters given the adaptation data. For the model adaptation, structural changes and parameter restrictions are introduced to robustly learn a speaker specific transformation without overfitting using a small speech data sampled from the recognition target speaker. Neto, et al., and Bo, et al., presented a Linear Input Network (LIN) that restricts the adapting layer to the input layer [11, 12]. The same idea is also applicable to the other layers: Linear Hidden Network (LHN) and Linear Output Network (LON). The computation of each layer consists of a matrix multiplication and a non-linear transformation.

- SVD
  Singular value decomposition (SVD) replaces the weight matrix to the product of two low-ranked matrices. The SVD-based parameter reduction showed effective adaptation [13]. The SVD is applied to the weight matrix of l-th hidden layer as:

$$\begin{aligned}\mathbf{w}^l_{m,n} &= U^l_{m,n}\Sigma^l_{n,n}(V^l_{n,n})^T,\\ &\approx U^l_{m,k}\Sigma^l_{k,k}(V^T_{n,k})^T,\end{aligned}$$

  where Σ is the diagonal matrix of singular values, and subscript is the size of weight matrix. The adaptation of the decomposed diagonal matrix and a further selection of k singular values decrease the number of free parameters. In our compared experiment, the singular values in the diagonal matrix, $\Sigma^l_{k,k}$, are updated for each target speaker.
- LHUC

Swietojanski, et al., [14] presented an approach that adapted the hidden units called learning hidden unit contributions (LHUC), which directly re-scaled amplitudes of the hidden units as following equation:

$$h_j^l = 2\sigma(r_j^{l,s}) \cdot \psi(\mathbf{w}_j^l \mathbf{h}^{l-1} + b_j^l),$$

where j the the index of the hidden units, $\mathbf{w}_j^l$ is the weight vector, and $\psi(\cdot^l)$ computes the l-th hidden units. The hidden units are re-scaled by applying element-wise multiplication with $\sigma(\cdot)$ ranging from 0.0 to 1.0. Variables $r_j^{l,s}$ are optimized for each target speaker s.

- fDLR

  In the particular case of feedforward DNNs, the neighboring frames of the acoustic features are concatenated to take the context information into account that contributes to the senone classification. Focusing on this stacked frames, Seide, et al., [10] inserted a linear layer that is tied across neighboring frames (fDLR; feature-space discriminative linear regression).

Zhao, et al., [93] presented an adaptation method to adapt the node activation function. LHUC and the adaptation of the node activation function also resemble a matrix multiplication of a diagonal matrix. During the model adaptation, a large number of parameters must be trained without causing any overfitting. Yu, et al., [94] presented a Kullback-Leibler divergence-based regularization to address this concern. Such model adaptation techniques only focus on the reduction of free parameters. However, we must consider expressiveness against the total number of free parameters for the adaptation under limited available data.

Several model adaptation techniques can be regarded as matrix multiplications. Figure 4.1 summarizes the relation among adaptation methods and the introduced restrictions to the matrix. LIN inserts a matrix at the bottom of the DNN without any restrictions that resembles a fully connected layer. fDLR introduces a restriction where the matrix is a block-diagonal type and the block is shared across the diagonal. Therefore, the frame-based transformation is carried out for each frame. Likewise, VTLN is regarded as a transformation by a tri-diagonal matrix, even though it is not adapted under the backpropagation framework [95]. LHUC is also regarded as a matrix multiplication by introducing a restriction under which the matrix takes a diagonal matrix. From these results, LHUC's expressiveness is included in the fDLR, which is again included in the LIN. The categorization of the feature transformations, which are based on the considerations of the Spectro-Temporal domain, was previously discussed in [96].

Some studies reported acoustic models based on convolutional neural networks (CNNs) [97]. A convolution operation focuses on small localized regions of the input speech, unlike the fully connected layer. In addition, weight sharing significantly reduces the number of free parameters. Kaneyama, et al., [98] proposed a method to apply convolutional filters that follows a Gabor function for an image texture classification task. CNN's success shows that the introduction of structural restrictions is critical to capture locally invariant features and further improves the recognition performance even though fully connected neural networks include CNN capability.

Other studies reported methods that represent speaker characteristics as a combination of components of *bases*. Cluster adaptive training (CAT) combines multiple weight matrices using an interpolation vector to form one final DNN layer [15]. In the adaptation stage, the interpolation vector is updated while maintaining the weight matrices. The Factorized Hidden Layer (FHL) approach resembles CAT [16]. In FHL,

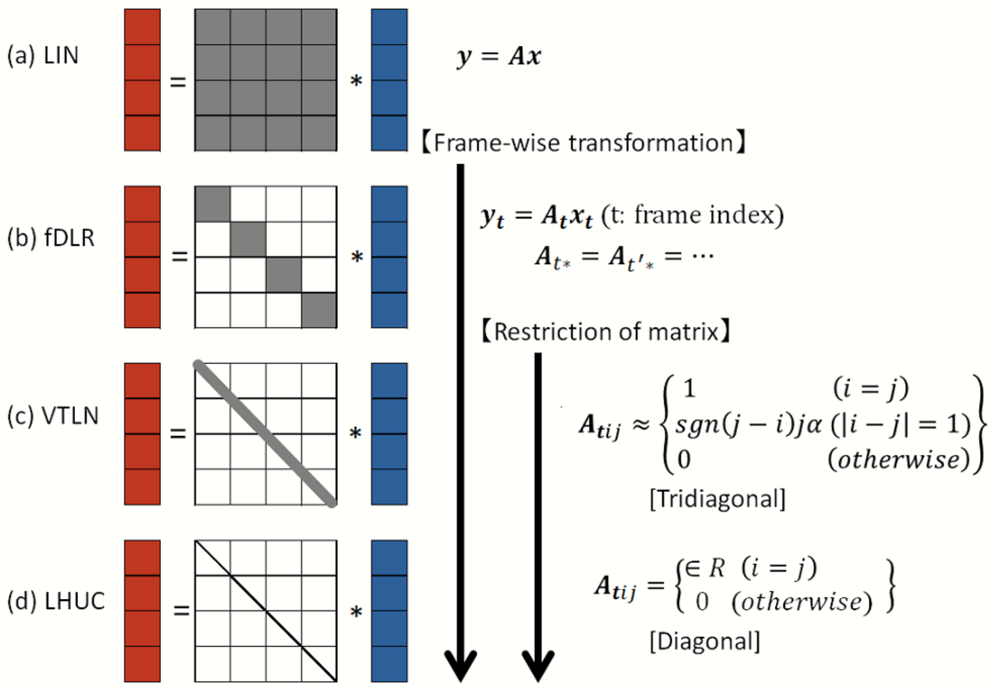

(a) LIN
=
*
$y = Ax$
【Frame-wise transformation】
(b) fDLR
=
*
$y_t = A_t x_t$ (t: frame index)
$A_{t*} = A_{t'*} = \cdots$
【Restriction of matrix】
(c) VTLN
=
*
$A_{tij} \approx \begin{cases} 1 & (i = j) \\ sgn(j-i)j\alpha & (|i-j| = 1) \\ 0 & (otherwise) \end{cases}$
[Tridiagonal]
(d) LHUC
=
*
$A_{tij} = \begin{cases} \in R & (i = j) \\ 0 & (otherwise) \end{cases}$
[Diagonal]

Fig. 4.1 Relations among adaptation methods and introduced restrictions: LIN inserts a matrix $\mathbf{A}$ without adaptation. Here, the input is composed of four frames. fDLR introduces a restriction with a block-diagonal matrix and the block is shared across the diagonal ($\mathbf{A}_t = \mathbf{A}_{t*} = ...$). VTLN and LHUC are regarded as a matrix multiplication with tri-diagonal and diagonal matrices, respectively.

the interpolation vector is shared among the several layers and initialized by the i-vector. By introducing the interpolation vector, these studies separate speaker and phone spaces for efficient adaptations. CAT and FHL only adapt the interpolation vector, and the robust adaptation is guaranteed only within the range that covered by the training data.

4.3 Neural network based filterbank layer

4.3.1 Hand-crafted triangular filterbank

Filterbank feature is calculated by weighing spectra of speech waveform using triangular filterbank. The vertex of triangles is configured according to the mel-scale which models non-linear sensitivity of human perception. The mel-scale is defined as follows:

$$\mathrm{Mel}(f) = 1127.0 \ln(\frac{f}{700.0} + 1.0)$$

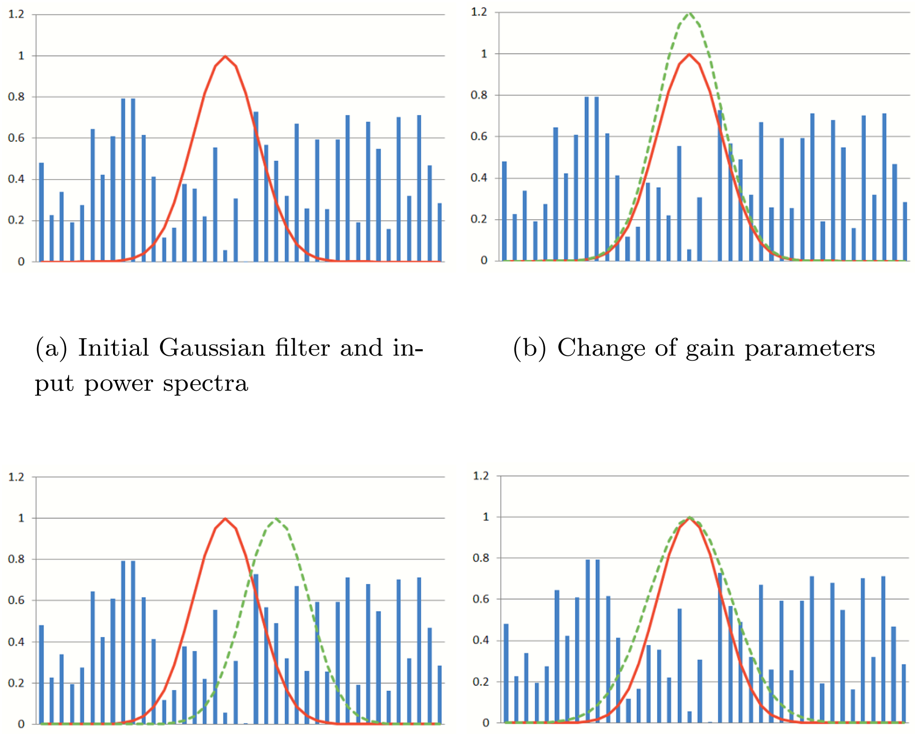
where f is the linear frequency and $\mathrm{Mel}(f)$ is the mel-scale frequency. The pre-defined configuration of filterbank is unchanged at all times.

4.3.2 Gaussian filterbank

The hand-crafted triangular filters are used as filter shapes in general. However, this triangular filter is not differentiable and cannot be incorporated into a scheme of the backpropagation algorithm. To parametarize the filter, Biem, et al., modeled its shape as a Gaussian function [99]:

$$\theta_n(f) = \varphi_n \exp\left\{-\beta_n(\mathrm{Mel}(\gamma_n) - \mathrm{Mel}(f))^2\right\}, \tag{4.1}$$

where $\theta_n(f)$ is the n-th filter at frequency f. φ_n is the gain parameter, β_n is the bandwidth parameter, and γ_n is the center frequency parameter, respectively. A function $\mathrm{Mel}(\cdot)$ maps linear frequency f to the mel-scale. Three trainable parameters, φ_n, β_n, and γ_n, control the filter shape. Figure 4.2 visualizes the role of three parameters. A change of the gain parameter scales the magnitude of the filterbank feature. This function is also realized by the adjusted weight in the following layer. A change of the center frequency parameter shifts the region of the power spectra on which the filter is focused. A change of the bandwidth parameter enlarges the power spectra region on which the filter is focused. A set of Gaussian filters can be regarded as a neural network layer that maintains a function of frequency domain smoothing.



(a) Initial Gaussian filter and input power spectra

(b) Change of gain parameters

(c) Change of center frequency

(d) Change of bandwidth

Fig. 4.2 Roles of parameter changes, gain, center frequency, and bandwidth. The x- and y-axis of each subfigure are power spectra and corresponding amplitude. The red line represents an initial filter shape, and the green dotted line represents the filter shape after re-tuning.

Figure 4.3 shows an overview of the Gaussian filterbank-incorporated DNN. Power spectra at frame t, $x_t(f)$, are concatenated from several consecutive frames and fed into the filterbank layer. These features are multiplied by the corresponding filter gain given by Eq. (4.1) and summed across the frequency bin. Then applying a log-compression gives the following neural network based log mel-scale filterbank features:

$$h_{t,n} = \log(\sum_f \theta_n(f) x_t(f))$$

$$\mathbf{h}_t = [h_{t,1}, h_{t,2}, ..., h_{t,n}, ..., h_{t,N}]$$

where N is the number of filters and t is the frame index. For the training of the following DNN, $\mathbf{h}_t$ with consecutive $\pm c$ frame features, $[\mathbf{h}_{t-c}, ..., \mathbf{h}_t, ..., \mathbf{h}_{t+c}]$, are fed into the following layer to compute the posterior probability of the triphone states. We call this architecture Gaussian filterbank incorporated DNN (GFDNN).

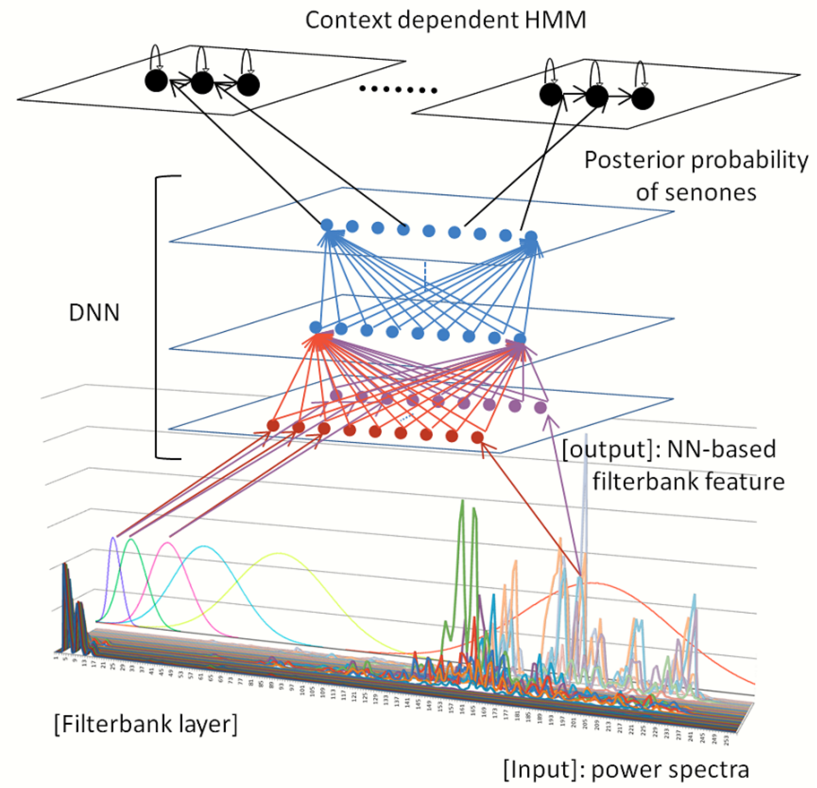

Context dependent HMM
Posterior probability of senones
DNN
[output]: NN-based filterbank feature
[Filterbank layer]
[Input]: power spectra

Fig. 4.3 Overview of Gaussian filterbank incorporated DNN: Filterbank weighting is performed at DNN's bottom. Horizontal axis is for the frequency bin, and vertical axis is for the power spectrum. In the experiment, input power spectra are concatenated from several consecutive frames (depth).

4.3.3 Gammatone filterbank

Under this framework, arbitrary differentiable filter functions can be used as filter shapes. To compare the recognition performance among filter types, we also used a Gammatone filter, which is a widely used model as an auditory filter [100]. A Gammatone filter is modeled as:

$$g_n(t) = c_n t^{a-1} \exp(-2\pi b_n t) \cos(2\pi f_0(n)t + \zeta_n), \tag{4.2}$$

where c_n is the constant value, ζ_n is the phase, a is the order, b_n is the temporal decay, and $f_0(n)$ is the center frequency, respectively. A equation (4.3) is obtained by applying the Fourier transform to Eq. (4.2):

$$\begin{aligned} H_n(f) = &\frac{c_n}{2}(a-1)!(2\pi b_n)^{-a} \\ &\left\{ e^{(i\zeta_n)} \left[1 + \frac{i(f - f_0(n))}{b_n}\right]^{-a} + e^{(-i\zeta_n)} \left[1 + \frac{i(f + f_0(n))}{b_n}\right]^{-a} \right\}, \end{aligned}$$

$$\begin{aligned}\theta_n(f) = |H_n(f)|^2 \sim k_n^2 \\ \left\{ \left[1 + \frac{(f - f_0(n))^2}{b_n^2} \right]^{-a} + \left[1 + \frac{(f + f_0(n))^2}{b_n^2} \right]^{-a} \right\},\end{aligned} \tag{4.3}$$

where

$$\begin{aligned} a &= 4, \\ k_n &= \frac{c_n}{2}(a-1)!(2\pi b_n)^{-a}, \\ \zeta_n(f) &= \tan^{-1}\left\{ \frac{-2f_0(n)b_n}{b_n^2 + (f^2 - f_0(n)^2)} \right\}. \end{aligned}$$

The followings are trainable parameters of the Gammatone filter: k_n (gain), $f_0(n)$ (center frequency), and b_n (temporal decay). In the experiment, the initial values of $f_0(n)$ and b_n are set [101, 102]:

$$f_0(n) = -\eta + (f_{\max} + \eta)\exp\left\{ \frac{n \log \frac{f_{\min}+\eta}{f_{\max}+\eta}}{N} \right\} \tag{4.4}$$

$$b_n = 1.019 \times 24.7 \times (f_0(n) \times \frac{4.37}{100} + 1) \tag{4.5}$$

$$\eta = 228.83, \tag{4.6}$$

where n is the index of filters, N is the total number of filters, and $f_{\min}$ (in Hz) and $f_{\max}$ (in Hz) are the lowest and highest cutoff frequencies of the filterbank, respectively. As seen in Eq. (4.3), the Gammatone filter takes a line asymmetric curve. Mitra, et al, reported the effectiveness of Gammatone filterbank features for DNN acoustic model in noisy condition (Note that our experimental condition is clean environment) [103]. The Gammatone filterbank incorporated by DNN (GtFDNN) without the update of filterbank corresponds to the DNN with Gammatone filterbank features.

4.4 Discriminative learning of filterbank

4.4.1 Training of filterbank layer

The filterbank layer parameters are trained by backpropagation. The update rule of φ_n, for example, is as follows:

$$\begin{aligned} \varphi_n^{\text{new}} &= \varphi_n^{\text{old}} - \eta \frac{\partial L}{\partial \varphi_n} \\ &= \varphi_n^{\text{old}} - \eta \frac{\partial L}{\partial h_n} \frac{\partial h_n}{\partial \varphi_n}, \end{aligned}$$

where L is the objective function and η is the learning rate. The other parameters, β_n, γ_n (for GFDNN), k, f_0, and b (for GtFDNN), are updated in the same manner. In the experiment, the filterbank incorporated models are trained in two stages. First, except for the filterbank layer, the DNN was fine-tuned until a convergence criterion is met. Hereinafter, we refer to this model as a fixed model. Then the filterbank layer and the following DNN are trained jointly with the same initial learning rate. Hereinafter, we refer to this model as a trained model.

4.4.2 Adaptation of filterbank layer

During the adaptation stage, the filterbank layer is adapted for specific speaker while the other network parameters are fixed. The target speakers for adaptation have different vocal tract shape and vocal tract length. This difference of vocal tract length corresponds to the shift of power spectra in the acoustic feature space domain.

We considered whether there is a relation between the learned center frequencies and the vocal tract length. A vocal tract's average length depends on gender and age. The average length of the vocal tracts of Japanese adult males and females is 17.0 cm and 15.5 cm, respectively. Theoretically, the spectra of female speakers shift to an approximately 9.7% (17.0/15.5) higher frequency domain from that of male speakers due to the differences of vocal tract length. Therefore, we assume that the center frequencies of the filterbank layer shift to an 9.7% higher frequency domain by adapting the filterbank layer of the male-specific DNN using the female speech data. It should be noted that the shift (warping) of frequencies in VLTN is also accomplished by the adjustment of channel gains. The function of VTLN is executed by both the shift of center frequencies and the scale of filter gains.

4.4.3 Advantage of filterbank learning

The filterbank incorporated DNNs have some advantages compared with earlier studies.

- The proposed method can compute the neural network based log mel scale filterbank features.
- The shapes of filters are adapted in a discriminative manner using backpropagation.
- Unlike the fully connected layer, the proposed system performs a framewise

transformation. Each filter takes a certain portion of the input power spectra. The initial center frequency and bandwidth values are described in Section ii.

- An adjustment of the gain parameters corresponds to fMLLR [39] and fDLR [10]. An adjustment of the center frequency parameters corresponds to VTLN [38] by regarding the frequency shift as frequency warping. In summary, our proposed system has fMLLR and VTLN capability while minimizing the number of free parameters.
- The filterbank layer, which consists of a small number of parameters, is effective for the adaptation under limited available data while fully neural network based architecture suffers from the overfitting problem (e.g. time-domain convolution layer in [34] has 16,000 parameters).

4.4.4 Earlier works on filterbank learning

Finding an optimal filterbank is an important topic not only for speech recognition but also for speaker recognition, dialization, and event detection. Several studies proposed methods based on heuristic search algorithms. Pinheiro, et al., [104] proposed a scheme to find the best filterbank configuration using an Artificial Bee Colony (ABC) algorithm for speaker verification. Charbuillet, et al., [105] proposed a method to search for optimal center frequency and bandwidth based on genetic algorithms. These heuristic search algorithms independently repeat both the *selection* and *evaluation* stages. Several studies proposed methods that introduce objective functions. Kobayashi, et al., [106] and Burget, et al., [107] proposed methods based on a dimensionality reduction technique, and Suh, et al., [108] proposed a method that measures filterbank properties derived from the Kullback-Leibler (KL)-divergence among filters. Recently, hierarchical feature extraction based on deep neural networks has become a topic of interest in classification tasks [35]. Sainath, et al., [34] presented a method to apply convolution over a raw time-domain waveform. Sailor, et al., [33] also proposed a method based on a convolutional Restricted Boltzmann Machine that uses a raw time-domain waveform. Tokozume, et al., [109] presented an end-to-end convolutional neural network for environmental sound classification. Su, et al., [110] further introduced an event-specific Gaussian filterbank layer to handle different temporal properties of audio events. In this paper, we propose a novel approach to train and adapt a filterbank based on DNN.

- ExpFDNN (DNN with exponential filter)

  Sainath, et al., [111] proposed a method to jointly train a filterbank layer and the following networks under a restriction where the elements of the filters take positive values by introducing the exponential of weights (Exponential filterbank incorporated DNN; ExpFDNN):

$$h_{t,n} = \exp\left(\mathbf{w}_n\right)\mathbf{x}_t = \sum_f \exp\left(w_n(f)\right)x_t(f),$$

where n is the filter index, f is the frequency bin, $\mathbf{w}_n$ is the weight vector of n, and $x(f)$ are the input power spectra. However, this weak restriction does not explicitly give a frequency-domain smoothing function, which is the original purpose of the hand-crafted triangular filterbank. In other words, the parameters of the filterbank layer overfit to the given data and the shape of the filters leads to multiple peaks. Figure 4.4 shows an example of the actual filter shapes that were fine-tuned in the experiment. This ExpFDNN characteristic could become a disadvantage in the adaptation. Therefore, we also trained this model for comparison.

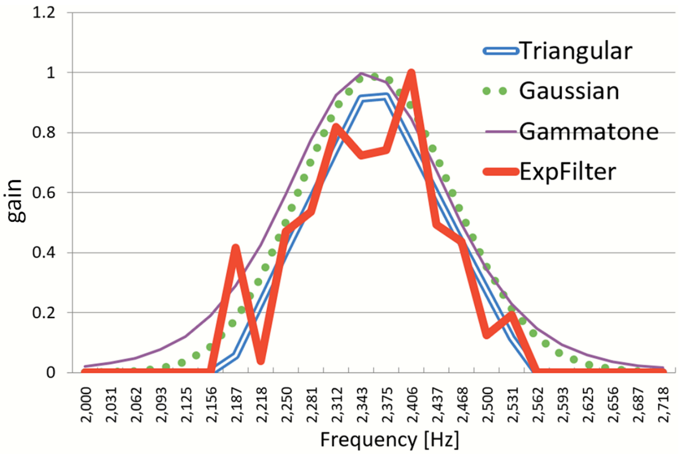


Fig. 4.4 Example of actual filter shapes that were fine-tuned in the experiment. Blue double line shows conventional triangular filter. Green dotted line is a Gaussian filter, and red bold line is an exponential filter.

4.5 Experimental work

4.5.1 Experimental setup

(i) Corpus

We used the Corpus of Spontaneous Japanese (CSJ) [112] for the validation. The details of the corpus is shown in Table 4.1[*1]. It consists of 186.0 hours of speech of male speakers (SM) and 42.0 hours of speech of female speakers (SF). We used an officially attached evaluation set-2 for the evaluation that consists of five male speakers and five female speakers. We used all utterances of the evaluation set-2 as the test data for speaker-independent experiment and gender adaptation experiment. In the case of speaker-independent experiment, we trained SM-specific, SF-specific, and gender independent models and tested the models using the gender-matched test data. In the case of gender adaptation experiment, we adapted the SM-specific models using the training data of female speakers, and tested the models using female speakers in the evaluation set-2. In the case of speaker adaptation experiment, we assigned 20 utterances to the adaptation data and 40 utterances to the test data. The SM specific models were trained by male speakers and they were adapted/tested by 5 male speakers in the evaluation set-2. The OOVs are 0.0% and 0.0%, and the perplexities are 73.5 and 73.2 for male and female speakers, respectively.

The speech was analyzed using a 25-ms Hamming window with a pre-emphasis coefficient of 0.97 and shifted with a 10-ms frame advance.

(ii) Acoustic model

We built hybrid DNN-HMM systems. For the experiment of speaker and gender

Table. 4.1 Details of CSJ corpus.

	Gender	Male (SM)	Female (SF)
Train	Lectures	787	166
	Data	186 hours	42 hours
(Test Evaluation set-2)	Lectures	5	5
	Data	1.0 hours	0.9 hours

[*1] We first evaluated the effectiveness of our proposed method using a small size corpus. The results are described in Appendix A.

adaptations, we implemented some conventional model adaptation techniques. The following is the experimental setup of GMM-HMM and DNN:

GMM-HMM

To obtain the training target labels for DNNs, GMM-HMMs are trained using a corpus of SM, SF, and SM plus SF (mixed). The models were trained on the standard MFCC features. The senones of SM, SF, and SM plus SF were 4783, 4860, and 5023, respectively. Corresponding GMM-HMMs were used for forced alignment.

Baseline DNN (Triangle filterbank)

As a baseline system, we trained a fully connected DNN, which has five hidden layers with 2,048 rectified linear units [55]. Its input was 11 consecutive frames of 40-dimensional log mel-scale triangular filterbank features extracted using the Hidden Markov Model Toolkit (HTK) [43]. The features were normalized to zero mean and unit variance. Due to the fixed and undifferentiable shape of the triangular filters, the filter shapes are unchanged all the times.

Gaussian filterbank incorporated DNN (GFDNN)

The Gaussian filterbank layer in Eq. (4.2) was inserted to the bottom of the baseline DNN as in Fig. 4.3. Its input was 11 consecutive frames of 256-dimensional power spectra. The number of filters was set to 40, which is the same as the baseline system ($n = 1, 2, ..., 40$). The initial values of the gain parameter were set to 1.0. The center frequencies were spaced equally along the mel-scale. The bandwidths were set so that the two-sigma range of Gaussian filter, i.e., $(\mu - 2\sigma, \mu + 2\sigma)$, equals the corresponding bandwidth of the mel-scale filterbank. At the Gaussian filterbank layer, 120 parameters, which consist of φ (gains), γ (center frequencies), and β (bandwidths), were updated using the backpropagation.

Gammatone filterbank incorporated DNN (GtFDNN)

The Gammatone filterbank layer in Eq. (4.3) was inserted to the bottom of the baseline DNN. The initial values were set according to Eqs. (4.4), (4.5), and (4.6). The other setup was the same as the GFDNN. At the Gammatone filterbank layer, 120 parameters were updated using the backpropagation.

ExpFDNN

A DNN with the exponential filterbank layer [111] was trained for the comparison. The initial values of the filterbank layer were set similar to the triangular filterbank. At the filterbank layer, 10,240 parameters (256 frequency bins of 40 filters) were updated using backpropagation. The other setup was identical to the GFDNN.

feature-space discriminative linear regression (fDLR)

A linear layer was inserted to the bottom of the baseline DNN, and it was inserted after the filterbank layer for the GtFDNN. The size of identity matrix was 40 by 40.

Learning hidden unit contribution (LHUC)

To adapt the models, we re-scaled the hidden units using LHUC. In the experiment, the hidden units of the third layer were adapted which showed the best performance in the preliminary experiment. The number of parameters was 2,048.

Singular value decomposition (SVD)

We applied SVD on the 1st fully connected layer and kept top 420 singular values. These values were decided from the best performance.

Table 4.2 summarizes the number of updated parameters for each adaptation method. Comparative adaptation methods, fDLR, LHUC, and SVD, are applicable to GtFDNN since the training of filterbank layer and the adaptation of hidden layer is independent. Therefore, we applied comparative speaker adaptation methods targeting the baseline (triangular shape filter) DNN and GtFDNN. The filterbank layer was un-adapted when the comparative methods were applied to GtFDNN.

We used Chainer [113] for training the DNNs. The models were trained using Adam [66] with batch normalization [114]. The 1% of training data were used for the model selection. We followed the existing Kaldi recipe [115] for the training of GMM-HMMs and decoding.

Table. 4.2 Number of parameters updated in adaptation stage.

Target of adaptation	Parameters
GFDNN	120 (40filters × 3)
GtFDNN	120 (40filters × 3)
ExpFDNN	10,240 (256bins × 40filters)
fDLR	1,600 (40dims. × 40dims.)
LHUC	2,048 (hidden units)
SVD	420 (hidden units)

4.5.2 Speaker independent model

The performance of the speaker-independent models is shown in Table 4.3. The baseline gender independent DNN, which takes triangular filterbanks, achieved an average WER of 13.4%. When we focused on the baseline models and the *fixed* (untrained) models, the latter outperformed the baseline models in all cases, even though the filter shapes were the only difference between the two models. This difference changes the coverage of the frequency bin. The Gaussian and Gammatone filters focus on all the frequency bins while the baseline triangular filter zeroes out the frequency bins outside a certain bin distance. These results comparing fixed and baseline models showed the importance of refined acoustic features. Mitra, et al., also investigated the effectiveness of robust features for DNN including Gammatone filterbank [103]. The performance improvement of the fixed models corresponds with their results. The Gammatone filter is widely used as an auditory filter. However, the difference between filter types did not show any performance gain.

In comparison with the fixed models, the trained models did not show performance improvement. We considered that the difference in optimal center frequencies between male and female speakers made it difficult to learn universal center frequencies for both male and female speakers. In the following experiment, we only present the results of the trained models.

Table. 4.3 WERs (%) of baseline DNN and filterbank incorporated DNNs.

System	WER (%)			
	SM	SF	Ave.	SM+SF
Baseline (Triangle)	12.4	20.4	16.4	13.4
GFDNN (fixed)	12.4	18.8	15.6	12.5
GFDNN (trained)	12.5	19.0	15.8	12.9
GtFDNN (fixed)	12.1	16.3	14.2	12.6
GtFDNN (trained)	12.1	15.9	14.0	12.6
ExpFDNN	12.3	17.0	14.7	12.9

Table. 4.4 WERs (%) of gender adaptation from adult male speakers to adult female speakers. Bold is the best performance among models.

Adaptation data	# speakers	GFDNN	GtFDNN			
		filterbank	filterbank	fDLR	LHUC	SVD
0.0 h	0	26.8	**24.4**	**24.4**	**24.4**	28.2
0.02 h (72 seconds)	20	**22.3**	22.7	23.3	22.9	28.2
0.03 h (108 seconds)	30	**18.9**	20.1	22.0	20.9	27.2
0.1 h (360 seconds)	51	**16.4**	16.5	17.8	20.3	22.3
0.5 h (1800 seconds)	166	15.7	14.9	17.0	**13.9**	18.2
1.0 h	166	15.4	15.4	16.5	**13.8**	14.6
10.0 h	166	14.9	15.6	16.1	**14.2**	14.3
30.0 h	166	15.0	15.2	16.1	**13.6**	14.0

Adaptation data	# speakers	Baseline DNN (Triangle)			ExpFDNN	
		fDLR	LHUC	SVD	filterbank	
0.0 h	0	26.5	26.5	26.5	25.2	
0.02 h (72 seconds)	20	26.0	32.8	28.6	23.6	
0.03 h (108 seconds)	30	27.1	32.2	28.2	20.4	
0.1 h (360 seconds)	51	19.1	31.8	23.7	17.2	
0.5 h (1800 seconds)	166	17.4	31.2	18.2	15.7	
1.0 h	166	19.4	31.6	16.2	14.1	
10.0 h	166	16.6	31.2	14.3	14.6	
30.0 h	166	16.6	31.4	14.1	14.7	

4.5.3 Gender adaptation

Next, we performed gender adaptation from SM to SF as shown in Table 4.4 to confirm the presence of the filters' shift to alleviate the difference of vocal tract length. The first column is a duration of SF speech data for adaptation. The row of 0 utterance is WERs of the models without the adaptation. The WERs of the SM-specific GFDNN and GtFDNN were worse at 26.8% and 24.4%, due to the gender mismatched condition. For the evaluations of 10 and 20 utterances, 60 utterances in Table 4.4 were split into six or three folds, and averaged to alleviate any selection bias. In the scenario of limited adaptation data, the best performance was obtained when we adapted the filterbank layer of GFDNN. We considered that the focus on filter adaptation worked on the alleviation of gender mismatch effectively while discarding

Table. 4.5 Shift of center frequencies [Hz] of 1-st to 20-th filters caused by gender adaptation from SM to SF speakers using 10 hours of data. SM → SF shows center frequencies of unadapted and adapted models. SF column shows center frequencies of SF-specific model trained using SF speech data.

	SM → SF			SF
n	Before Adaptation	After Adaptation	Difference	-
1	28.1	74.4	46.4 (165.2 %)	31.5
2	86.1	95.7	9.6 (11.2 %)	87.0
3	156.1	213.3	57.3 (36.7 %)	153.5
4	192.6	207.6	15.0 (7.8 %)	187.9
5	251.2	299.5	48.3 (19.2 %)	249.2
6	315.8	369.7	54.1 (17.2 %)	310.5
7	382.6	438.1	55.5 (14.5 %)	374.3
8	449.4	508.1	58.7 (13.1 %)	439.3
9	530.6	592.0	61.4 (11.6 %)	524.3
10	597.2	669.0	71.8 (12.0 %)	589.5
11	701.1	741.1	39.9 (5.7 %)	687.1
12	783.9	912.9	129.1 (16.5 %)	783.7
13	866.1	976.1	110.0 (12.7 %)	874.9
14	953.6	1033.9	80.4 (8.4 %)	965.4
15	1055.1	1103.9	48.7 (4.6 %)	1061.5
16	1183.2	1219.9	36.7 (3.1 %)	1186.8
17	1307.8	1374.9	67.1 (5.1 %)	1316.3
18	1433.6	1501.3	67.7 (4.7 %)	1439.3
19	1553.3	1604.6	51.4 (3.3 %)	1561.3
20	1734.1	1805.8	71.7 (4.1 %)	1724.7

other mismatched conditions that were difficult to adapt under limited data. When the adaptation data increased to 0.5 hour and more, the best model was replaced by LHUC which has larger free parameters.

By adapting the GFDNN from SM to SF speakers, we considered that the frequency shift of the filters was caused by the differences of the vocal tract lengths. Table 4.5 and 4.6 show the relation among the center frequencies of SM-dependent GFDNN, adapted GFDNN from SM to SF using 10 hours of data, and SF-dependent GFDNN. Theoretically, an ideal frequency shift is approximately 9.7%, as described in Sec-

Table. 4.6 Shift of center frequencies [Hz] of 21-th to 40-th filters caused by gender adaptation from SM to SF speakers using 10 hours of data. SM → SF shows center frequencies of unadapted and adapted models. SF column shows center frequencies of SF-specific model trained using SF speech data.

	SM → SF			SF
n	Before Adaptation	After Adaptation	Difference	-
21	1879.2	1963.4	84.2 (4.5 %)	1878.0
22	2032.8	2151.0	118.3 (5.8 %)	2040.1
23	2169.0	2180.0	11.0 (0.5 %)	2188.4
24	2363.0	2425.7	62.7 (2.7 %)	2373.9
25	2558.3	2587.2	28.9 (1.1 %)	2564.3
26	2774.0	2785.9	11.9 (0.4 %)	2780.7
27	2990.7	2991.2	0.5 (0.0 %)	3000.2
28	3240.0	3250.1	10.1 (0.3 %)	3246.6
29	3492.0	3514.4	22.5 (0.6 %)	3497.1
30	3757.5	3798.2	40.7 (1.1 %)	3752.5
31	4004.2	4098.8	94.6 (2.4 %)	4026.6
32	4323.6	4355.2	31.6 (0.7 %)	4340.0
33	4637.1	4619.7	-17.4 (-0.4 %)	4654.4
34	4993.6	5063.9	70.3 (1.4 %)	4999.3
35	5373.9	5481.7	107.8 (2.0 %)	5376.7
36	5740.0	5843.0	103.0 (1.8 %)	5747.4
37	6148.8	6189.1	40.4 (0.7 %)	6155.0
38	6596.5	6719.6	123.1 (1.9 %)	6595.5
39	7067.0	7200.9	133.9 (1.9 %)	7067.7
40	7535.7	7657.5	121.7 (1.6 %)	7537.1

tion 4.4.2. The column of difference shows that the actual shift was approximately 4.6% to 17.2%, which resembles the theoretical value at low- and middle-frequency region ($300Hz \sim 1000Hz$). These results show that the optimization of the filterbank layer causes a shift of the center frequencies to discriminatively perform frequency warping. This characteristic corresponds to the VTLN function.

The last column of the Table 4.5 and 4.6, SF, showed the center frequencies of the SF-dependent GFDNN. When we focused on the SM- and SF-dependent GFDNNs, the relation between the two models cannot be observed in the experiment. Instead,

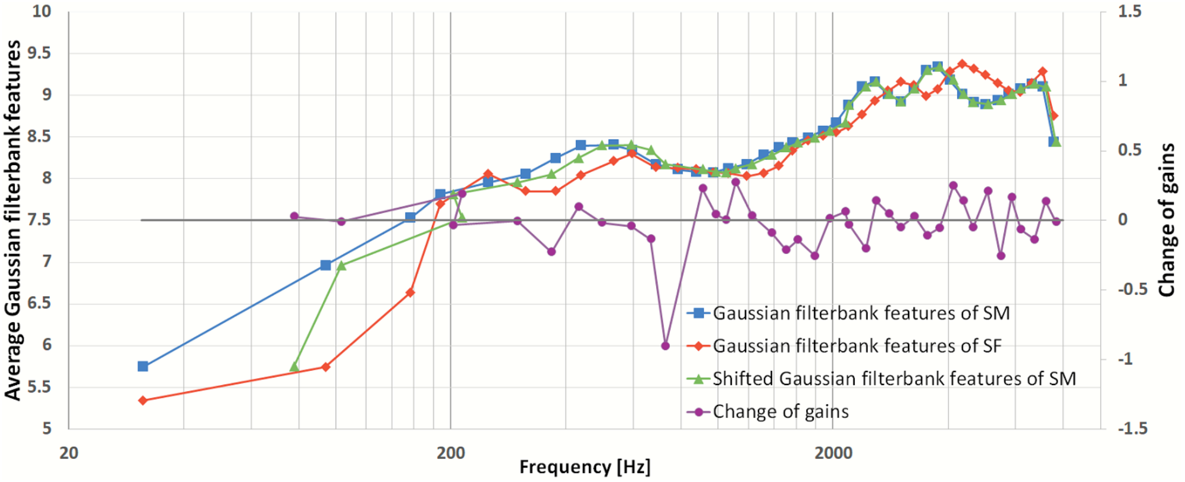


Fig. 4.5 Changes of gain parameters from SM-specific model (SM) to SF-adapted model and averaged Gaussian filterbank features of SM- and SF-speakers.

the learned center frequencies based on SF speakers showed lower frequencies than those of the SM speakers at $n = 6 \sim 12$ because the optimal position of the filters in the training stage depends on the condition of the following DNN. However, in the adaptation stage, the filterbank layer was updated, and the parameters of the following DNN were fixed. In this situation, the filterbank layer can be handled independently of the following DNN to perform frequency warping.

Figure 4.5 shows the gender-dependent Gaussian filterbank features and the change of gain parameters. The square markers and diamond markers show the Gaussian filterbank features of SM and SF speakers, respectively. The triangle markers show the SM-dependent features with horizontal shifting according to the change of center frequencies caused by gender adaptation to SF-speakers from SM-speakers. We can see that the features of SM at low-frequency region shifted toward the ones of SF speakers. Next, the change of gain parameters was depicted at the right vertical axis. To emphasize the conspicuous change of gains, their relative changes were plotted with circle markers by computing $\log(gains\_of\_SF/gains\_of\_SM)$. In the low-frequency region, the change of gain was relatively small while the shift of center frequency was remarkable. Conversely, the shift of center frequency was relatively small at a high-frequency region ($\sim 2000Hz$). The variances of the filterbank features are relatively large enough to overlap the SM- and SF-speakers. Therefore, the optimization of the gain parameters was a secondarily important factor in gender adaptation. In contrast to the above two parameters, no discrimination of the change of the bandwidth parameters was observed in gender adaptation.

4.5.4 Speaker adaptation

Table 4.7 shows the supervised adaptation result. The models trained by SM in Table 4.3 were used as the source models. The 0 utterance row shows the WERs, which were recognized using the model without adaptation. By adapting the filterbank layer of GFDNN using 5 utterances, the WER was improved from 12.5% to 12.0%, and a word error reduction rate (WERR) of 4.0% was obtained. By adapting the filterbank layer of GtFDNN using 15 utterances, the WER was improved from 12.1% to 11.2%, and a word error reduction rate (WERR) of 7.4% was obtained. This WERR was better than the unadapted GtFDNN at a significance level of 0.005 under a statistical sign test. These results showed that the adjustment of the filter shapes can handle the diversity of speakers.

Performance gains were observed when adaptation was applied for GFDNN with 5 utterances ($p < 0.03$) and GtFDNN with 10 utterances ($p < 0.005$). These results were better than the other adaptation methods, although the baseline DNN with LHUC and ExpFDNN also showed performance improvements when more than 15 or 10 utterances were available for each method. Table 4.7 also shows the adaptation result using fDLR, LHUC, and SVD under the same GtFDNN. The adaptation of the filterbank layer obtained the best performance on all conditions of adaptation utterances among other adaptation methods.

Finally, we depicted the relation between adaptation utterances and WERs per

Table. 4.7 WERs (%) of the triphone level supervised speaker adaptation. Bold is the best performance among models.

#utt	GFDNN	GtFDNN				Baseline DNN (triangle)			ExpFDNN
	filterbank	filterbank	fDLR	LHUC	SVD	fDLR	LHUC	SVD	filterbank
0	12.5	**12.1**	**12.1**	**12.1**	13.2	12.4	12.4	13.4	12.3
1	12.3	**12.2**	13.6	13.3	13.1	36.0	13.2	13.3	12.4
2	**12.2**	12.9	13.6	14.1	13.1	15.4	12.8	13.4	12.7
3	12.5	12.8	13.6	14.4	13.2	12.7	**12.4**	13.4	13.1
4	**12.4**	12.6	13.2	14.1	13.1	13.1	12.5	13.3	13.0
5	**12.0**	12.3	13.4	13.9	13.0	13.0	12.3	13.2	12.8
10	**11.4**	**11.4**	13.4	12.9	12.8	13.0	12.4	12.6	11.7
15	**11.2**	**11.2**	13.4	12.5	12.6	12.2	12.0	12.7	11.4
20	11.4	11.3	13.3	12.1	12.5	12.7	11.9	12.5	**11.2**

speaker targeting GtFDNN in Fig. 4.6. The WERs of almost all the speakers decreased linearly over the adaptation utterances. However, Speaker 1 showed unexpected behavior when the value of the horizontal axis was 2 to 5.

In addition, the proposed filterbank layer is a simple neural network module and can be combined with other modules, e.g. CNNs (Convolutional Neural Networks), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) [116]. In the next chapter, we extend the filterbank layer to the end-to-end attention-based encoder decoder networks and investigate its noise adaptation.

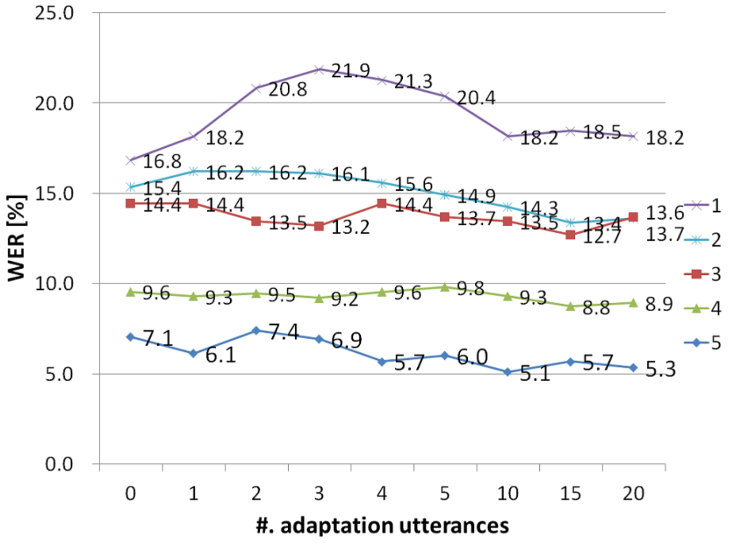


Fig. 4.6 Relation between number of adaptation utterances and WERs per speaker (Speaker 1 to Speaker 5). GtFDNN in Table 4.7 was used for error analysis.

Chapter 5

Rapid noise adaptation by neural network based feature extraction

5.1 Introduction

Noise robust automatic speech recognition system is a widely studied problem for practical use in a real-world environment. Many methods are proposed in the past and Li, et al., summarized the noise robust techniques into five categories [117]: 1) feature-domain vs. model-domain processing, 2) the use of prior knowledge about the acoustic environment distortion, 3) the use of explicit environment-distortion models, 4) deterministic vs. uncertainty processing, 5) the use of acoustic models trained jointly with the same feature enhancement or model adaptation process used in the testing stage. Zhang, et al., [118] summarized recent techniques based on deep learning for robust automatic speech recognition.

Feature space approaches compensate the acoustic features as a pre-processing. Relative spectral processing (RASTA)- Perceptual Linear Predictive (PLP) [119], Power-Normalized Cepstral Coefficients (PNCC) [120], and Gabor filterbank feature [121] are widely used as noise-robust acoustic features. TempoRAL Pattern (TRAP) uses narrow band spectrum with long temporal context as its input to take temporal trajectory into account. Normalization of acoustic feature is also employed for noise suppression. CMN, CMVN, and histogram equalization (HEQ) [122] normalize statistical moment(s) of the acoustic feature. Feature compensation approaches aim at eliminating unnecessary noise information from noisy speech. These methods include Ideal Binary/Ratio Mask (IBM/IRM) [123], Spectral Subtraction (SS) [124] and Wiener filter [125].

Deep neural network is employed for noise robust automatic speech recognition. Multi-condition training uses acoustic features of all conditions for training of one DNN. This approach is simple but it brings competitive performance than other DNN based approaches when a large amount of speech with various noise types are available. Denoising auto-encoder aims at estimating clean speech signal given noisy speech [126, 127] and it is also used for noise estimation [128]. The estimated noise feature is used as an auxiliary feature. Noise aware training takes noisy speech and the estimated noise information as its input [128, 129]. The estimation of clean speech and the label classification are combined as a multi-task training [130, 131]. Modeling of each separation/classification module based on the deep neural network and its combination can lead to direct optimization of speech separation network based on ASR loss [130]. Adversarial training is one promising approach and is studied to train noise-independent feature extractor [132, 133].

In the previous chapter, we proposed the filterbank layer adaptation and showed its effectiveness for speaker adaptation. Experimental results showed the filters' shift to match the characteristics of target evaluation speakers. In this chapter, we integrate the filterbank layer with an end-to-end attention-based encoder decoder networks framework and conduct noise adaptation. Our proposed method has relation to IBM and IRM by regarding the change of filter gains as mask values for noise suppression. Therefore, we mainly compare the filterbank adaptation with IBM/IRM.

This chapter is organized as follows: We first review earlier works on noise suppression in Section 5.2. We integrate the neural network filterbank with the end-to-end architecture in Section 5.3. Experimental setup and results are described in Section 5.4.

5.2 Earlier works on noise suppression

5.2.1 Ideal binary/ratio mask (IBM/IRM)

Ideal binary mask (IBM) and ideal ratio mask (IRM) are one of typical approaches for noise suppression [123]. The IBM is defined as:

$$\mathrm{IBM}(t,f) = \begin{cases} 1 & \text{if } \mathrm{SNR}(t,f) > \mathrm{thr} \\ 0 & \text{otherwise} \end{cases}$$

where thr is a threshold to decide mask or not, and $\mathrm{SNR(t,f)}$ is signal-to-noise ratio (SNR) at f-th dimensional feature at frame t. The estimated mask is multiplied with the corresponding acoustic feature. Therefore, the (time-frame independent) IBM can be regarded as a control of the gain parameters. In the experiment, our method is compared with the IBM based noise suppression method and thr was set 0.0.

The IRM is defined as:

$$\mathrm{IRM}(t, f) = 10 \log_{10} \frac{|X_{t,f}|^2}{|N_{t,f}|^2},$$

where $|X_{t,f}|^2$ and $|X_{t,f}|^2$ are power spectra of clean speech and noise signal, and t and f are time and frequency indices.

5.2.2 Spectral subtraction

Spectral subtraction (SS) estimates spectral feature of noise using silent frames and subtracts the estimated noise spectral feature from the noise-overlapped spectral feature to estimate the power spectrum of clean speech [124]. Let $|X_{t,f}|^2$ be the f-dimensional noise-overlapped power spectram at time frame t, and $|N_f|^2$ be the estimated power spectram of noise. Then, the estimated power spectram of clean speech S is defined as:

$$|\hat{S}_{t,f}|^2 = \begin{cases} |X_{t,f}|^2 - \alpha |N_f|^2 & \text{if } \ |X_{t,f}|^2 - \alpha |N_t|^2 \geq 0.0 \\ \beta |X_{t,f}|^2 & \text{otherwise,} \end{cases}$$

where α is a subtraction factor and β is a flooring factor. In the experiment, we set α to 2.0 and β to 0.0, and applied SS targeting filterbank feature. We followed the noise estimation method as:

$$|N_f|^2 = \frac{1}{M} \sum_{m=1}^{M} |X_{t,f}|^2,$$

where M is the available frames and set to 30 in this experiment. In addition to the spectral subtraction, many noise suppression methods are proposed including Wiener filtering [134] and minimum mean square error short-term spectral amplitude (MMSE-STSA) [135].

5.3 Filterbank learning within an end-to-end framework

Let $x = (x_t \in \mathbb{R}^F | t = 1, \cdots, T)$ be a T-frame utterance of F-dimensional power spectra, and $x_{t,f}$ be an f-th dimensional power spectra at frame t. The power spectra are multiplied by the filter and summed across the frequency bin, and a following log-compression gives the neural network based filterbank feature:

$$h_{t,n} = \log(\sum_{f=1}^{F} \theta_n(f) x_{t,f}) \quad \text{for } n = 1, \cdots, N,$$

where N is the number of filters. A sequence of the neural network based filterbank features is calculated by concatenating all frames of features.

$$\begin{aligned} h_t &= (h_{t,n} | n = 1, \cdots, N), \\ h &= (h_t | t = 1, \cdots, T). \end{aligned}$$

In our preliminary experiment, we found it is important for convergence to normalize feature at filterbank level. For this purpose, global mean (μ) and variance (σ^2) are calculated by using the training data, and used for feature normalization:

$$\begin{aligned} \mu &= (\mu_n | n = 1, \cdots, N), && (5.1) \\ \sigma^2 &= (\sigma_n^2 | n = 1, \cdots, N), && (5.2) \\ h_{t,n} &\leftarrow \frac{h_{t,n} - \mu_n}{\sqrt{\sigma_n^2}}. \end{aligned}$$

The acoustic feature of test data is also normalized by the global mean and variance in Eqs. (5.1) and (5.2). The normalization is also applied in the evaluation stage using the same statistics.

The calculated neural network based filterbank feature is further fed into the encoder network. A decoder network generates posterior probability of a set of labels L at decoding time step j as $p(L_j)$ by taking a previous label L_{j-1}, a context vector c_j, and hidden states in the recurrent connection s_{j-1}:

$$p(L_j) = \text{Decoder}(L_{j-1}, c_j, s_{j-1}).$$

The context vector c is obtained by multiplying an attention weight α to the hidden vector e generated by the encoder network:

$$\begin{aligned} c_j &= \sum_t \alpha_{j,t} e_t, \\ \alpha_{j,t} &= \text{Attention}(\alpha_{j-1,t}, s_{j-1}, e), \\ e &= \text{Encoder}(h), \end{aligned}$$

where t is time step for the encoder network [6].

During the training stage, the loss is backpropagated to the bottom of the network, and the parameters of the filterbank layer are updated under the framework of backpropagation as a multi-condition training [136] with known noisy speech. During the adaptation stage, the filterbank layer is adapted for specific noises (and speakers) while other network parameters including the encoder network, decoder network, and attention network are fixed. Update of the filterbank layer can work as noise suppression by changing the filter gains depending on the specific spectral pattern of each noise. The same as in the speaker adaptation experiment, all parameters, i.e., gains, center frequencies, and temporal decays, contribute to the noise adaptation.

5.4 Experimental work

5.4.1 Experimental setup

(i) Corpus

We used ASJ+JNAS [87] corpus (#.speaker: 133) and further added noises from NOISEX-92 database [137] to a quarter of the speech of male speakers in ASJ+JNAS while varying the signal-to-noise ratio (SNR). Noise types of speech, car, F16, and Lynx (N1-N4) with 10 dB, 15 dB, and 20 dB SNRs are used to deteriorate the speech. The duration of the generated data was 134.0 hours. We conducted noise adaptation and noise-and-speaker joint adaptation. In the case of noise adaptation, 10 speakers are assigned as the noise adaptation data, and the other 13 speakers are assigned as the evaluation set. In total, the adaptation data consists of 4,160 utterances (10 speakers × 52 utterances × 8 noise types) and the test data consists of 2,080 utterances (13 speakers × 20 utterances × 8 noise types). The speakers of training data, adaptation data, evaluation data are selected exclusively as an open set. In the case of noise and speaker joint adaptation, the adaptation data and the test data consist of 13 speakers which were selected exclusively from the training data. As with the training data, we added the noise of speech, car, F16, and Lynx as a closed set. We trained the models as a multi-condition training. We also prepared an open set by adding 4 noises: machine gun, STITEL, factory, and operation room (N5-N8). As the evaluation data, we added the noises (N1-N8) to the clean speech at various SNRs: 5, 10, 15, and 20dB. In total, the adaptation data consists of 3,120 utterances (13 speakers × 30 utterances × 8 noise types) and the test data consists of 2,080 utterances (13 speakers × 20 utterances × 8 noise types). The speakers of adaptation and test data are the same in the case of noise and speaker joint adaptation.

(ii) Network architecture

We trained joint CTC/attention-based encoder decoder networks [6] using Chainer [113] and ESPnet [73]. We used a 6 layer bi-directional long short-term memory (BLSTM) as the encoder network. The 2nd and 3rd bottom layers of the encoder network sub-sample hidden vector by the factor of 2 [68]. Each BLSTM layer has 320 cells in each direction, and is followed by a linear projection layer with 320 units to combine the forward and backward LSTM outputs. The decoder network has a 1-layer LSTM with 300 cells. The number of labels was set to 2,247 including Japanese Kanji/Hiragana/Katakana characters and special tokens.

We trained an additional network for the prediction of IRM mask. The network has 3 stacked bi-directional LSTM with projection layer followed by softmax layer. Each LSTM and projection layer had 320 units. The IRM network was trained to predict the IRM mask for each time and frequency bin. In the case of IRM based ASR system, the Gammatone filterbank incorporated encoder decoder networks trained as multi-condition training was further retrained by using the masked noisy/clean speech.

In the case of spectral subtraction, the Gammaatone filterbank incorporated encoder decoder networks trained as multi-condition training was further retrained by using the noise suppressed feature based on spectral subtraction.

5.4.2 Speaker-independent model

We first trained multiple speaker- and noise-independent models using multi-condition training. Table 5.1 shows the CER of the baseline model which takes the triangular filterbank feature. The results were summarized with regard to the noise types and SNRs. In the case of baseline system, the average CER of known noises was 11.5% and that of unknown noises was 52.1%. At the training stage of Gammatone filterbank incorporated models, we took two-step training procedure which first trains the encoder decoder network without filterbank layer followed by full training of both the filterbank layer and the encoder decoder networks. The results of two models correspond to *fixed-* and *trained-*Gammatone fitlerbank and were shown in Table 5.2 and Table 5.3, respectively. The CER of the fixed model

Table. 5.1 CER (%) of the baseline encoder decoder networks (Input acoustic feature: triangular filterbank feature).

	dB				
	5	10	15	20	Avg.
N1	20.5	11.9	11.2	10.4	13.5
N2	10.8	10.5	12.3	12.1	11.4
N3	20.4	12.9	12.0	10.5	14.0
N4	18.3	13.0	13.5	8.9	13.4
Avg. (Known noise)	17.5	12.1	12.2	10.5	13.1
N5	60.2	34.3	25.2	19.7	34.9
N6	129.7	73.3	40.1	17.0	65.0
N7	45.6	30.8	20.1	14.1	27.7
N8	97.6	67.4	33.5	21.2	54.9
Avg. (Unknown noise)	83.3	51.5	29.7	18.0	45.6
Avg.	50.4	31.8	21.0	14.2	29.3

Table. 5.2 CER (%) of the *fixed* Gammatone-filterbank-incorporated encoder decoder networks (Input acoustic feature: power spectra).

	dB				
	5	10	15	20	Avg.
N1	19.6	12.1	11.7	10.0	13.3
N2	11.0	9.7	12.1	11.5	11.1
N3	20.9	13.8	12.0	10.3	14.2
N4	18.5	13.1	13.1	9.7	13.6
Avg. (Known noise)	17.5	12.2	12.2	10.4	13.1
N5	70.3	47.9	23.8	18.2	40.0
N6	92.1	58.2	33.1	15.2	49.6
N7	47.4	37.1	18.8	12.2	28.9
N8	98.8	67.8	51.9	15.2	58.4
Avg. (Unknown noise)	77.2	52.7	31.9	15.2	44.2
Avg.	47.3	32.4	22.0	12.8	28.6

Table. 5.3 CER (%) of the *trained* Gammatone-filterbank-incorporated encoder decoder networks (Input acoustic feature: power spectra).

	dB				
	5	10	15	20	Avg.
N1	18.3	10.7	9.6	9.3	12.0
N2	11.1	8.1	10.3	10.6	10.1
N3	18.1	12.8	9.9	8.9	12.4
N4	16.1	12.7	11.8	9.0	12.4
Avg. (Known noise)	15.9	11.1	10.4	9.5	11.7
N5	73.6	52.4	30.5	17.1	43.4
N6	107.5	59.0	41.3	15.6	55.9
N7	50.6	31.1	16.1	11.2	27.2
N8	141.9	119.4	54.5	14.7	82.6
Avg. (Unknown noise)	93.4	65.5	35.6	14.6	52.3
Avg.	54.7	38.3	23.0	12.0	32.0

averaged over the known noises was 13.1%, and it was 11.7% in the case of the trained model. The trained model showed better performance than the fixed model and the baseline model because the filterbank layer was optimized as a data-driven filterbank targeting the known noises included in the training data. In contrast, the CERs of the fixed model averaged over the unknown noises was 44.2%, and it was 52.3% in the case of the trained model. We can see that the canonical filter shape is useful for recognition of unknown noisy speech than the optimized filterbank targeting different other noises. In actual use, the trained model should be used because the training data are collected to cover speech in real conditions. In the case of known noises, the average CER of the fixed model was 20.2% and it was decreased to 17.9% on the trained model and obtained 11.3% relative improvement. In the following adaptation experiment, we only report the results obtained by adapting the trained model.

5.4.3 Noise adaptation

Figures 5.1 and 5.2 show the CERs of noise adaptation targeting the known and unknown noises, respectively. The models were adapted to specific noise type N1-N8 consisting of multiple 10 speakers. In the case of known noises, the average CER before adaptation was 17.9%, and it was improved to 15.2% when the filterbank layer was adapted using 500 utterances. The CERs were decreased linearly over the increase of the adaptation data size.

In the case of unknown noises, the average CER was 76.6% and it was improved to 25.9% when the filterbank layer was adapted using 100 utterances. These results indicate that our method can adapt to the unknown noisy speech by updating the filterbank layer. However, further increase of the adaptation data did not achieve performance improvement. The average CER of unknown noises was approximately 21.0%, and there is a room for further performance improvement compared with the results of known noises in Figure 5.1. It is considered that the filterbank layer can rapidly adapt to the noise characteristics that are easy to adapt. It also means a lack of expressiveness in a scenario where the system can use a large number of adaptation data. One simple approach for performance improvement is an increase of free parameters at the adaptation stage.

Our research focus is the rapid adaptation of the filterbank layer. Therefore, we further decrease the adaptation data size by making the adaptation data using one single speaker, and adapt the filterbank layer to specific noise and speaker jointly.

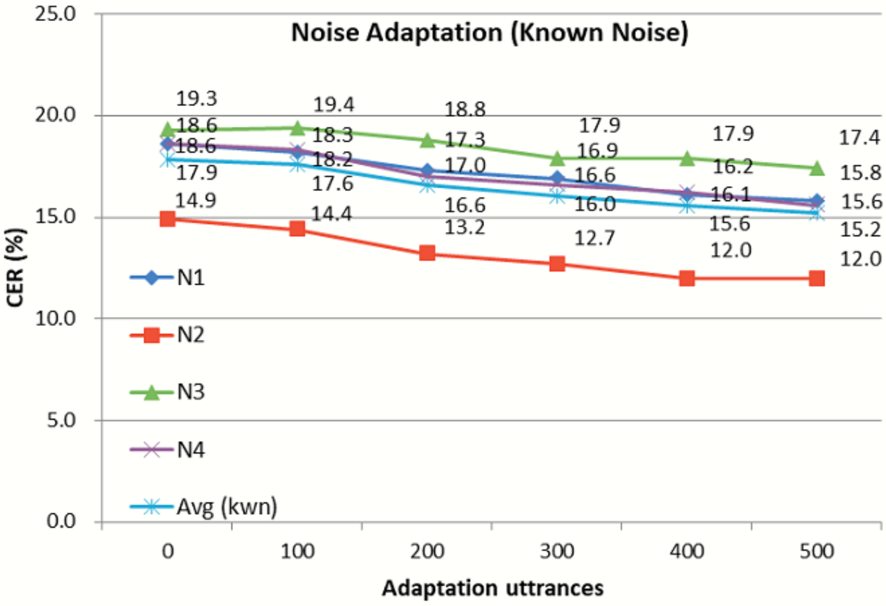


Fig. 5.1 Adaptation to known noises.

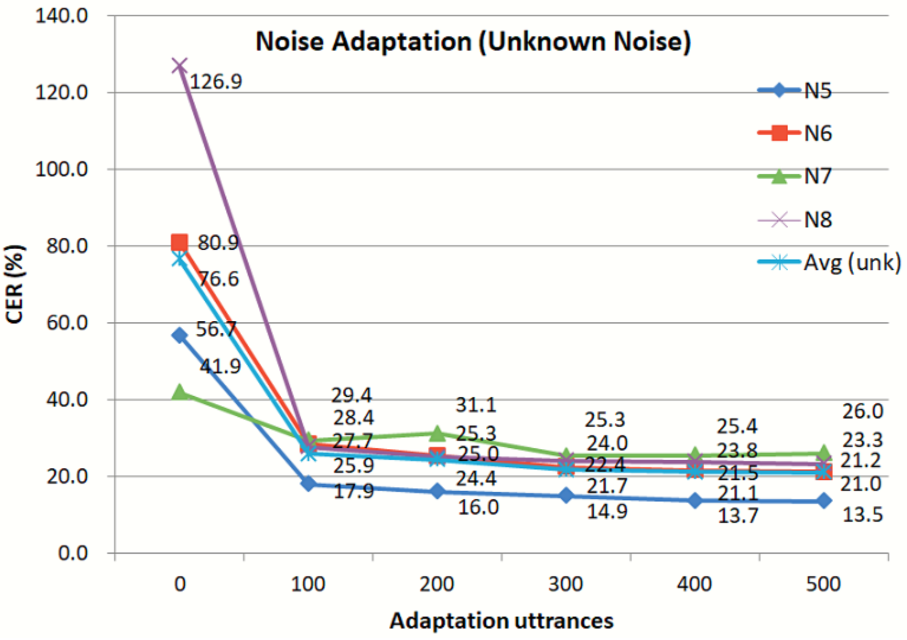


Fig. 5.2 Adaptation to unknown noises.

Table. 5.4 Adaptation of the filterbank layer using 10 utterances (CER %).

	dB				
	5	10	15	20	Avg.
N1	14.3	7.2	6.4	5.0	8.2
N2	6.3	5.4	6.7	6.3	6.2
N3	14.7	7.1	5.9	5.5	8.3
N4	17.0	10.8	10.3	8.3	11.6
Avg. (Known noise)	13.1	7.6	7.3	6.3	8.6
N5	67.0	40.1	19.0	14.7	35.2
N6	99.4	44.4	20.1	11.1	43.7
N7	43.7	29.9	14.2	11.4	24.8
N8	101.9	77.2	42.5	12.3	58.5
Avg. (Unknown noise)	78.0	47.9	23.9	12.3	40.5
Avg.	45.5	27.8	15.6	9.3	24.6

Table. 5.5 Adaptation of the filterbank layer using 20 utterances (CER %).

	dB				
	5	10	15	20	Avg.
N1	10.9	3.8	3.5	3.4	5.4
N2	4.1	4.0	3.7	3.5	3.8
N3	11.1	6.9	3.1	3.8	6.2
N4	15.7	9.8	9.7	6.6	10.4
Avg. (Known noise)	10.4	6.1	5.0	4.3	6.5
N5	48.4	38.6	18.0	9.6	28.7
N6	70.9	32.8	11.2	7.7	30.7
N7	42.7	22.6	13.2	8.8	21.8
N8	47.0	26.9	13.6	8.7	24.0
Avg. (Unknown noise)	52.3	30.2	14.0	8.7	26.3
Avg.	31.4	18.2	9.5	6.5	16.4

Table. 5.6 Adaptation of the filterbank layer using 30 utterances (CER %).

	dB				
	5	10	15	20	Avg.
N1	8.8	3.5	1.5	1.7	3.9
N2	2.3	1.1	1.8	2.1	1.8
N3	9.0	2.6	2.7	1.7	4.0
N4	13.2	8.9	8.2	5.8	9.0
Avg. (Known noise)	8.3	4.0	3.5	2.8	4.7
N5	36.6	23.3	17.1	9.0	21.5
N6	55.1	28.3	13.7	11.0	27.0
N7	37.6	25.7	13.2	10.0	21.6
N8	43.1	23.7	10.6	8.9	21.6
Avg. (Unknown noise)	43.1	25.2	13.7	9.7	22.9
Avg.	25.7	14.6	8.6	6.3	13.8

5.4.4 Noise and speaker adaptation

In this section, the filterbank layer was adapted to specific noise and speaker jointly, and compared with other noise suppression methods and model adaptation techniques. We adapted the filterbank layer using 10, 20, or 30 utterances and the corresponding results were shown in Table 5.4, 5.5, and 5.6. The average CER of the known noises without adaptation was 11.7%, and it was decreased to 8.6% and achieved 26.5% relative improvement by adapting the filterbank layer using the 10 adaptation utterances. The average CER was further decreased to 4.7% by adapting the filterbank layer using the 30 adaptation utterances. The average CER of the unknown noises without adaptation was 52.3%, and it was decreased to 40.5% and achieved 22.6% relative improvement by adapting the filterbank layer using the 10 adaptation utterances. The average CER was further decreased to 22.9% by adapting the filterbank layer using the 30 adaptation utterances. Our proposed method obtained significant performance improvement without over-fitting problem even though available adaptation data was limited to 10 utterances, and the CERs were monotonically decreased over the increase of adaptation data.

The above recognition performance was better than the ones in the Figure 5.1 and Figure 5.2 even though the adaptation data size was limited. In this experiment, the models were adapted to the specific noise and speaker jointly. Therefore, this improvement was also brought by speaker adaptation but not only from the noise adaptation. We further investigate the contribution of each factor in the next section.

5.4.5 Speaker-independent noise adaptation targeting filterbank layer

In the previous section, the filterbank layer was adapted towards specific noise condition and speaker. To investigate the contribution of each factor, we made adaptation data consisting of 12 speakers and also made test data using the excluded single open speaker. For the evaluation, we made 13 folds of adaptation-evaluation data and averaged the recognition performance over 13 speakers (which can be regarded as speaker-independent noise adaptation). The results were shown in Table 5.7.

In the case of adaptation for known noises, the CERs showed almost the same recognition performance as the model without adaptation (as in Table 5.3) independently from the number of adaptation utterances. In other words, the adaptation

Table. 5.7 CERs (%) of speaker-independent noise adaptation using 20 utterances.

	dB				
	5	10	15	20	Avg.
N1	19.7	10.7	10.0	10.0	12.6
N2	10.3	9.0	10.3	10.2	9.9
N3	18.7	12.8	10.3	9.5	12.8
N4	18.1	12.0	12.1	8.0	12.5
Avg. (Known noise)	16.7	11.1	10.7	9.4	12.0
N5	47.2	29.0	13.1	11.1	25.1
N6	74.9	35.7	14.3	8.8	33.4
N7	40.4	26.8	14.3	10.2	22.9
N8	43.6	21.7	14.1	9.8	22.3
Avg. (Unknown noise)	51.5	28.3	13.9	10.0	25.9
Avg.	34.1	19.7	12.3	9.7	19.0

targeting known noises shown in Table 5.5 was trivial and the speaker adaptation mainly contributed to the performance improvement.

In the case of unknown noises, the average CER of Table 5.7 was comparable to the ones in Table 5.5, and we can see that the noise was the main contribution factor. In the experiment, the number of speakers of adaptation data was 12. Therefore, it is also considered that the model learned 12-speaker-specific filter shapes and it was not a canonical filters for unknown speakers.

5.4.6 Comparison with earlier works

Oracle IBM

Table 5.8 shows the CERs obtained by IBM-based noise suppression assuming oracle condition. The Gammatone filterbank incorporated DNN was trained by taking the masked feature. In the case of IBM-based systems, we masked time- and frequency-regions where the corresponding SNR bin was less than 0.

In the case of known noises, the improvement of our system, i.e. Table 5.4), were significant while the improvement of the IBM-based system was limited. This is because of the ability of the filterbank layer to adapt to noise and speaker jointly while

[*1] N5-N8 are known for IBM because of availability of clean speech.

Table. 5.8 CER (%) of ASR system with oracle IBM.

	dB				
	5	10	15	20	Avg.
N1	13.1	9.3	10.6	10.1	10.8
N2	9.2	9.0	10.6	9.4	9.6
N3	11.8	9.1	11.3	8.6	10.2
N4	10.8	10.1	11.8	9.7	10.6
Avg. (Known noise)	11.2	9.4	11.1	9.5	10.3
N5	10.7	11.6	8.9	10.0	10.3
N6	11.6	11.0	9.2	7.9	9.9
N7	11.2	9.2	9.5	8.0	9.5
N8	11.9	10.3	10.0	9.8	10.5
Avg. (Unknown noise) *1	11.3	10.5	9.4	8.9	10.1
Avg.	11.3	10.0	10.2	9.2	10.2
IBM thr = -5 dB (known)	13.1	10.6	11.0	10.2	11.2
IBM thr = -5 dB (unknown)	32.1	22.5	15.1	10.7	20.1
IBM thr = 5 dB (known)	11.7	9.9	10.4	9.3	10.3
IBM thr = 5 dB (unknown)	12.8	10.5	9.3	9.1	10.4

the IBM only focuses on the noise suppression. The CERs of unknown noises were better than the result of filterbank adaptation especially in low SNR setup. However, it should be noted that the performance is oracle result due to the requirement of corresponding clean speech.

IRM

Table 5.9 shows the performance of the IRM based system. The average CER of known noises was 11.6% and it was worse than the result of filterbank adaptation, i.e. the CER of adapted model using 10 utterances was 8.6% as shown in Table 5.4 and was similar to the model without adaptation as shown in Table 5.3.

In the case of unknown noises, the average CER was 38.6% and it was better than the result of baseline system, 52.3%, reported in Table 5.3. Comparison with the results of joint adaptation indicates that more than 20 utterances are required to outperform the result of IRM based system. It is considered that the use of ground truth transcription brings rich information and it leads to the best performance

Table. 5.9 Gammatone filterbank incorporated encoder decoder networks retrained by masked feature based on IRM (CER %).

	dB				
	5	10	15	20	Avg.
N1	18.0	10.6	9.7	10.0	12.1
N2	10.0	8.4	10.5	9.8	9.7
N3	17.9	11.9	10.3	9.7	12.5
N4	15.6	12.2	12.1	9.3	12.3
Avg. (Known noise)	15.4	10.8	10.6	9.7	11.6
N5	50.6	42.0	22.8	17.3	33.2
N6	84.1	54.7	26.0	10.9	43.9
N7	47.1	32.0	18.3	13.6	27.8
N8	88.9	62.1	34.6	12.5	49.5
Avg. (Unknown noise)	67.7	47.7	25.4	13.6	38.6
Avg.	41.5	29.2	18.0	11.7	25.1

compared with IRM. The existence of intersection point indicates the necessity of detailed application scenario for the following adaptation/suppression strategies.

SS

Table 5.10 shows the result of spectral subtraction (SS). In the case of known noises, the average CER was 12.2% and it did not show performance improvement compared with the model before retraining based on SS (11.7% in Table 5.3). In other words, the multi-condition training ans SS do not provide complimentary function and the multi-condition training can handle the diversity of noises included in the training data (in comparison with SS). In the case of unknown noises, the average CER was 25.9%. Comparison with the filterbank adaptation in Table 5.6 indicates that more than 30 utterances are required to outperform the result of SS.

5.5 Summary

Table 5.11 summarizes the CERs of proposed method and other comparison methods. In the case of known noises, the proposed method showed the best performance because the filterbank was optimized targeting unknown speaker in the evaluation data. The SS showed competitive performance in the case of unknown noises while it

Table. 5.10 CERs(%) of ASR system with spectral subtraction.

	dB				
	5	10	15	20	Avg.
N1	20.8	10.7	11.2	8.9	12.9
N2	11.0	8.7	10.6	9.8	10.0
N3	20.7	12.0	10.5	8.9	13.0
N4	18.3	13.3	11.2	8.6	12.9
Avg. (Known noise)	17.7	11.2	10.9	9.0	12.2
N5	54.2	37.6	18.4	10.8	30.2
N6	53.2	26.1	11.4	9.9	25.2
N7	45.6	41.6	16.3	9.5	28.2
N8	35.3	20.6	13.7	9.9	19.9
Avg. (Unknown noise)	47.1	31.5	14.9	10.0	25.9
Avg.	32.4	21.3	12.9	9.5	19.0

Table. 5.11 Average CERs of the baseline model (base), proposed model (GtF), and other method (IRM/SS). CERs are averaged over known noises and unknown noises, respectively.

known noise	Table 5.10	Table 5.3	Table 5.9	Table 5.4	Table 5.5	Table 5.6
	SS	base	IRM	GtF10	GtF20	GtF30
	12.2	11.7	11.6	8.6	6.5	4.7
unknown noise	Table 5.3	Table 5.4	Table 5.9	Table 5.5	Table 5.10	Table 5.6
	base	GtF10	IRM	GtF20	SS	GtF30
	52.3	40.5	38.6	26.3	25.9	22.9

was worse than the baseline system in the case of known noises. In addition, it should be noted that the proposed method requires ground-truth transcription for supervised adaptation. These results indicate each method has difference characteristics, and we developers need to consider strategies depending on the available data size, incoming known/unknown noise types, and availability of transcription.

Chapter 6

Conclusions

In this thesis, we proposed rapid adaptation methods based on addition of speaker class information and re-estimation of filterbank layer parameters.

6.1 Rapid speaker class adaptation using speaker information

Chapter 3 described the rapid adaptation based on the auxiliary feature.

We extended cluster dependent acoustic modeling mainly targeting GMM-HMM hybrid system to DNN-HMM hybrid system. A set of likelihoods defined between the input speech and the multiple clusters were defined as the speaker class information, and these features were fed into the DNN as the auxiliary feature aim to adapt rapidly. For this purpose, we restricted the available time period for estimation of speaker class to the first 50 frames, i.e., 500 ms, of an utterance. In the experiment of DNN-HMM system, all methods, speaker-class dependent CMVN and addition of speaker-class information, showed better performance than the baseline class-independent DNN-HMM system. Even when the available frames of an utterance used for the estimation speaker information was 0.5 second, the WER was decreased from 11.2% to 10.4% and the relative error reduction rate of 7.0% was obtained. When we evaluated the system using only the first words of the test set, the WER was decreased from 11.0% to 9.7% and the relative error reduction rate of 12.0% was obtained. These results demonstrated that speaker class, which was estimated from only the first 50 frames in the utterance, provided an important information to suppress the diversity of speakers, and it is applicable to the recognition of short time utterances consists of 0.5∼ second, i.e., speech retrieval, speech assistance, and speech command input.

One future direction is a combinational usage of i-vector for the recognition of longer utterance. Since the speaker class information and i-vector extract speaker characteristics in different ways, it is considered to provide complementary information. Samarakoon, et al., [16] proposed the factorized hidden layer (FHL) which makes a final weight matrix by interpolating *bases* based on i-vector. The speaker class information is also applicable to FHL. In addition, comparison with i-vector provides important information to know the upper bound of the speaker class information.

6.2 Rapid speaker adaptation by neural network based feature extraction

Chapter 4 described the filterbank incorporated DNN which had a filterbank layer at the bottom of the DNN, and evaluated its effectiveness for speaker adaptation. Compared with the baseline DNN, which uses log mel-scale triangular filterbank features as its input, the proposed method can discriminatively learn data-driven filter shapes. We conducted gender adaptation from male to female speakers and discussed the relation between the physical characteristics of the vocal tract length and an optimal filterbank shape from an engineering viewpoint. Experiments on gender adaptation showed that the optimization of the filterbank layer caused the shift of center frequencies to discriminatively perform frequency warping which corresponds to the VTLN function. Next, we conducted speaker adaptation by re-estimating the optimal filterbank shape for specific target speakers. By adapting the filterbank layer of GFDNN (Gaussian filterbank incorporated DNN) using 5 utterances, WER was improved from 12.5% to 12.0%, and a word error reduction rate (WERR) of 4.0% was obtained. By adapting the filterbank layer of GtFDNN (Gammatone filterbank incorporated DNN) using 15 utterances, WER was improved from 12.1% to 11.2%, and a word error reduction rate (WERR) of 7.4% was obtained. This WERR was better than the unadapted GtFDNN at a significance level of 0.005 under a statistical sign test. These results showed that the adjustment of the filter shapes can handle the diversity of speakers. The recognition performance of our proposed model was better than the other adaptation methods, although the baseline DNN with LHUC and ExpFDNN also showed performance improvements when more than 15 or 10 utterances were available for each method.

One future direction is an adaptation to child speakers and elder speakers. An average vocal tract length of child speakers is approximately 9.0 cm [139] and is different from the ones of adult speakers. We considered it is applicable to the recognition of child speech although there are other factors which degrade recognition performance, e.g., speaking style and vocabulary. Another direction is a change of filter types. Kleinschmidt [140] and Chang, et al., [141] used 2-dimensional Gabor feature as its input to the DNN. Insertion of the parameterized 2-dimensional Gabor filterbank and its data-driven optimization is a simple extension of our method.

6.3 Rapid noise adaptation by neural network based feature extraction

Chapter 5 extended the filterbank layer to the end-to-end attention-based encoder decoder networks and showed its effectiveness compared with the model which takes triangular filterbank feature as its input. We also conducted noise adaptation and joint noise and speaker adaptation by updating parameters of the filterbank layer. Experimental results on noise adaptation showed that the update of filterbank layer could adapt to noisy speech for both known noises and unknown noises. In the case of noise-and-speaker joint adaptation, the CER was decreased from 13.1% to 8.6% by adapting the filterbank layer using 10 utterances and achieved 34.4% relative improvement on average targeting the known noises. In the case of adaptation for the unknown noises, the average CER was decreased from 52.3% to 40.5% and achieved 22.6% relative improvement. Comparison with other noise suppression methods indicates that the necessity of strategic system design.

Flexible adaptation strategies that can consider (known/unknown) noise types, speaker characteristics, and data size are also an important future direction. Klejch, et. al., [138] proposed a meta-learning based adaptation targeting DNN- and TDNN (time delay neural network [142])-based acoustic model. We believe that meta-learning based adaptation of the end-to-end attention based encoder decoder networks can further reduce the burden of hyper-parameter tuning. The proposed filterbank layer takes power spectra is its input. Future work includes filterbank learning from raw waveform and its adaptation.

Acknowledgements

This thesis could not have been completed without advice and support from my advisers. I am very grateful to Dr. Seiichi Nakagawa at Chubu University for invaluable discussion and support of this thesis. I am grateful to Dr. Tomoyosi Akiba at Toyohashi University of Technology. I appreciate his generous support. I am also grateful to Dr. Kazumasa Yamamoto at Chubu University for advice especially for technical aspect. I would like to thank the leading graduate school at Toyohashi University of Technology. I have enjoyed tough but interesting program activities. I am indebted to Dr. Ryuki Tachibana at IBM Tokyo research laboratories and Dr. Richard Stern at Carnegie Mellon University for many advice and comments. I am grateful to Dr. Masaki Aono and Dr. Hitoshi Isahara for valuable comments and discussion. I would like to thank all members of the laboratories. This thesis is dedicated to them. Part of the work presented in this thesis is supported by the leading program at Toyohashi University of Technology.

Bibliography

[1] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, Vol. 29, No. 6, pp. 82–97, 2012.

[2] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. The Microsoft 2016 conversational speech recognition system. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5934–5938, 2017.

[3] Kartik Audhkhasi, Brian Kingsbury, Bhuvana Ramabhadran, George Saon, and Michael Picheny. Building competitive direct acoustics-to-word models for English conversational speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4759–4763, 2018.

[4] Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani. State-of-the-art speech recognition with sequence-to-sequence models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4774–4778, 2018.

[5] Jinyu Li, Guoli Ye, Amit Das, Rui Zhao, and Yifan Gong. Advancing acoustic-to-word CTC model. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5794–5798, 2018.

[6] Takaaki Hori, Shinji Watanabe, Yu Zhang, and Chan William. Advances in joint CTC-Attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM. In *Proc. Interspeech*, pp. 949–953, 2017.

[7] Sree Hari Krishnan Parthasarathi, Bjorn Hoffmeister, Spyros Matsoukas,

Arindam Mandal, Nikko Strom, and Sri Garimella. fmllr based feature-space speaker adaptation of dnn acoustic models. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[8] Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. Deep convolutional neural networks for LVCSR. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8614–8618. IEEE, 2013.

[9] Shakti P Rath, Daniel Povey, Karel Veselỳ, and Jan Cernockỳ. Improved feature processing for deep neural networks. In *Proc. Interspeech*, pp. 109–113, 2013.

[10] Frank Seide, Gang Li, Xie Chen, and Dong Yu. Feature engineering in context-dependent deep neural networks for conversational speech transcription. In *Automatic Speech Recognition and Understanding (ASRU)*, pp. 24–29. IEEE, 2011.

[11] Joao Neto, Luís Almeida, Mike Hochberg, Ciro Martins, Luis Nunes, Steve Renals, and Tony Robinson. Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system. In *Fourth European Conference on Speech Communication and Technology*, 1995.

[12] Bo Li and Khe Chai Sim. Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems. In *Proc. Interspeech*, 2010.

[13] Shaofei Xue, Hui Jiang, Lirong Dai, and Qingfeng Liu. Speaker adaptation of hybrid NN/HMM model for speech recognition based on singular value decomposition. *Journal of Signal Processing Systems*, Vol. 82, No. 2, pp. 175–185, 2016.

[14] Pawel Swietojanski, Jinyu Li, and Steve Renals. Learning hidden unit contributions for unsupervised acoustic model adaptation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 24, No. 8, pp. 1450–1463, 2016.

[15] Tian Tan, Yanmin Qian, and Kai Yu. Cluster adaptive training for deep neural network based acoustic model. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, Vol. 24, No. 3, pp. 459–468, 2016.

[16] Lahiru Samarakoon and Khe Chai Sim. Factorized hidden layer adaptation for deep neural network based acoustic modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 24, No. 12, pp. 2241–2250, 2016.

[17] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny. Speaker adaptation of neural network acoustic models using i-vectors. In *ASRU*, pp.

55–59, 2013.

[18] Ossama Abdel-Hamid and Hui Jiang. Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7942–7946. IEEE, 2013.

[19] Andrew Senior and Ignacio Lopez-Moreno. Improving DNN speaker independence with i-vector inputs. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 225–229. IEEE, 2014.

[20] Hengguan Huang and Khe Chai Sim. An investigation of augmenting speaker representations to improve speaker normalisation for DNN-based speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4610–4613. IEEE, 2015.

[21] Shaofei Xue, Ossama Abdel-Hamid, Hui Jiang, Lirong Dai, and Qingfeng Liu. Fast adaptation of deep neural network based on discriminant codes for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 22, No. 12, pp. 1713–1725, 2014.

[22] Mukund Padmanabhan, Lalit R Bahl, David Nahamoo, and Michael A Picheny. Speaker clustering and transformation for speaker adaptation in large-vocabulary speech recognition systems. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 2, pp. 701–704. IEEE, 1996.

[23] Kazuki Konno, Masaharu Kato, and Tetsuo Kosaka. Speech recognition with large-scale speaker-class-based acoustic modeling. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pp. 1–4. IEEE, 2013.

[24] Masaki Naito, Li Deng, and Yoshinori Sagisaka. Speaker clustering for speech recognition using vocal tract parameters. *Speech Communication*, Vol. 36, No. 3-4, pp. 305–315, 2002.

[25] R Faltlhauser and G Ruske. Robust speaker clustering in eigenspace. In *Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 57–60. IEEE, 2001.

[26] Hiroaki Nanjo and Tatsuya Kawahara. Speaking-rate dependent decoding and adaptation for spontaneous lecture speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. I–725. IEEE, 2002.

[27] Yu Zhang, Jian Xu, Zhi-Jie Yan, and Qiang Huo. An i-vector based approach to training data clustering for improved speech recognition. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[28] Misaki Tsujikawa, Tsuyoki Nishikawa, and Tomoko Matsui. Study on i-vector based speaker identification for short utterances (in japanese). *IEICE technical report*, Vol. 115, No. 99, pp. 65–70, 2015.

[29] Yulan Liu, Penny Karanasou, and Thomas Hain. An investigation into speaker informed DNN front-end for LVCSR. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4300–4304. IEEE, 2015.

[30] Zhuo Chen, Shinji Watanabe, Hakan Erdogan, and John R Hershey. Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[31] Zhenyao Zhu, Jesse H Engel, and Awni Hannun. Learning multiscale features directly from waveforms. pp. 1305–1309, 2016.

[32] Ehsan Variani, Tara N Sainath, Izhak Shafran, and Michiel Bacchiani. Complex linear projection (CLP): A discriminative approach to joint feature extraction and acoustic modeling. In *Proc. Interspeech*, pp. 808–812, 2016.

[33] Hardik B Sailor and Hemant A Patil. Novel unsupervised auditory filterbank learning using convolutional RBM for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 24, No. 12, pp. 2341–2353, 2016.

[34] Tara N Sainath, Ron J Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals. Learning the speech front-end with raw waveform CLDNNs. pp. 1–5, 2015.

[35] Zhong-Qiu Wang and DeLiang Wang. Joint training of speech separation, filterbank and acoustic model for robust automatic speech recognition. pp. 2839–2843, 2015.

[36] Markus Kitza, Ralf Schlüter, and Hermann Ney. Comparison of BLSTM-layer-specific affine transformations for speaker adaptation. pp. 877–881, 2018.

[37] Dong Yu, Frank Seide, Gang Li, and Li Deng. Exploiting sparseness in deep neural networks for large vocabulary speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4409–4412. IEEE, 2012.

[38] Puming Zhan and Alex Waibel. Vocal tract length normalization for large vocab-

ulary continuous speech recognition. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 1997.

[39] Mark JF Gales and Philip C Woodland. Mean and variance adaptation within the MLLR framework. *Computer Speech & Language*, Vol. 10, No. 4, pp. 249–264, 1996.

[40] Steven B Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In *Readings in speech recognition*, pp. 65–74. Elsevier, 1990.

[41] Hynek Hermansky. Perceptual linear predictive (PLP) analysis of speech. *the Journal of the Acoustical Society of America*, Vol. 87, No. 4, pp. 1738–1752, 1990.

[42] Chanwoo Kim and Richard M Stern. Power-normalized cepstral coefficients (PNCC) for robust speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4101–4104. IEEE, 2012.

[43] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al. The HTK book. *Cambridge university engineering department*, Vol. 3, p. 175, 2002.

[44] Kai-Fu Lee. Context-dependent phonetic hidden markov models for speaker-independent continuous speech recognition. In *Readings in speech recognition*, pp. 347–365. Elsevier, 1990.

[45] Steve J Young, Julian J Odell, and Philip C Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of the workshop on Human Language Technology*, pp. 307–312. Association for Computational Linguistics, 1994.

[46] Seiichi Nakagawa, Kengo Hanai, Kazumasa Yamamoto, and Nobuaki Minematsu. Comparison of syllable-based HMMs and triphone-based HMMs in Japanese speech recognition. In *Proc. International Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 393–396, 1999.

[47] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics*, Vol. 41, No. 1, pp. 164–171, 1970.

[48] Lawrence R Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257–286, 1989.

[49] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, Vol. 13, No. 2, pp. 260–269, 1967.

[50] Reinhard Kneser and Hermann Ney. Improved backing-off for M-gram language modeling. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, p. 181e4, 1995.

[51] Ian H Witten and Timothy C Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on information theory*, Vol. 37, No. 4, pp. 1085–1094, 1991.

[52] Paul Placeway, Richard Schwartz, Pascale Fung, and Long Nguyen. The estimation of powerful language models from small and large corpora. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 2, pp. 33–36. IEEE, 1993.

[53] Slava Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on acoustics, speech, and signal processing*, Vol. 35, No. 3, pp. 400–401, 1987.

[54] Joerg Ueberla. Analysing a simple language model· some general conclusions for language models for speech recognition. *Computer Speech & Language*, Vol. 8, No. 2, pp. 153–176, 1994.

[55] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323, 2011.

[56] László Tóth. Phone recognition with deep sparse rectifier neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6985–6989. IEEE, 2013.

[57] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

[58] Xiaohui Zhang, Jan Trmal, Daniel Povey, and Sanjeev Khudanpur. Improving deep neural network acoustic models using generalized maxout networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 215–219. IEEE, 2014.

[59] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, Vol. 11, No. Feb, pp. 625–660, 2010.

[60] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, Vol. 14, No. 8, pp. 1771–1800, 2002.

[61] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, Vol. 18, No. 7, pp. 1527–1554, 2006.

[62] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, Vol. 323, No. 6088, p. 533, 1986.

[63] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, Vol. 61, pp. 85–117, 2015.

[64] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, Vol. 12, No. Jul, pp. 2121–2159, 2011.

[65] Matthew D Zeiler. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[66] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[67] Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, Vol. 16, No. 1, pp. 69–88, 2002.

[68] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. End-to-end attention-based large vocabulary speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4945–4949. IEEE, 2016.

[69] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

[70] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. 1999.

[71] Albert Zeyer, Kazuki Irie, Ralf Schlüter, and Hermann Ney. Improved training of end-to-end attention models for speech recognition. *arXiv preprint arXiv:1805.03294*, 2018.

[72] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964. IEEE, 2016.

[73] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yolta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. ESPnet: end-to-end speech processing toolkit. *arXiv preprint arXiv:1804.00015*, 2018.

[74] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pp. 369–376. ACM, 2006.

[75] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning*, pp. 1764–1772, 2014.

[76] Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.

[77] Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5884–5888. IEEE, 2018.

[78] Yajie Miao, Mohammad Gowayyed, and Florian Metze. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 167–174. IEEE, 2015.

[79] Olli Viikki, David Bye, and Kari Laurila. A recursive feature vector normalization approach for robust speech recognition in noise. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 2, pp. 733–736. IEEE, 1998.

[80] Takahiro Shinozaki, Yu Kubota, and Sadaoki Furui. Unsupervised acoustic model adaptation based on ensemble methods. *IEEE Journal of Selected Topics in Signal Processing*, Vol. 4, No. 6, pp. 1007–1015, 2010.

[81] Tian Tan, Yanmin Qian, Maofan Yin, Yimeng Zhuang, and Kai Yu. Cluster adaptive training for deep neural network. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4325–4329. IEEE, 2015.

[82] Tetsuo Kosaka, Kazuki Konno, and Masaharu Kato. Deep neural network-based speech recognition with combination of speaker-class models. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*

(APSIPA), pp. 1203–1206. IEEE, 2015.

[83] Daisuke Enami, Faqiang Zhu, Kazumasa Yamamoto, and Seiichi Nakagawa. Soft-clustering technique for training data in age-and gender-independent speech recognition. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pp. 1–4. IEEE, 2012.

[84] Pere Pujol, Dusan Macho, and Climent Nadeu. On real-time mean-and-variance normalization of speech recognition features. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. I–I. IEEE, 2006.

[85] Alberto Yoshihiro Nakano, Seiichi Nakagawa, and Kazumasa Yamamoto. Distant speech recognition using a microphone array network. *IEICE transactions on information and systems*, Vol. 93, No. 9, pp. 2451–2462, 2010.

[86] ASJ-JIPDEC. http://research.nii.ac.jp/src/en/ASJ-JIPDEC.html.

[87] Katunobu Itou, Mikio Yamamoto, Kazuya Takeda, Toshiyuki Takezawa, Tatsuo Matsuoka, Tetsunori Kobayashi, Kiyohiro Shikano, and Shuichi Itahashi. JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research. *Journal of the Acoustical Society of Japan (E)*, Vol. 20, No. 3, pp. 199–206, 1999.

[88] Japanese newspaper article sentences read speech corpus of the aged (S-JNAS). http://research.nii.ac.jp/src/en/S-JNAS.html.

[89] CIAIR video game command voice (CIAIR-VCV). http://research.nii.ac.jp/src/en/CIAIR-VCV.html.

[90] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

[91] The mainichi newspapers. http://www.nichigai.co.jp/sales/mainichi/mainichi-series.html.

[92] Yasuhisa Fujii, Kazumasa Yamamoto, and Seiichi Nakagawa. Large vocabulary speech recognition system: SPOJUS++. In *Proc. International Conference on multimedia system & signal processing (MUSP)*, pp. 110–118. Citeseer, 2011.

[93] Yong Zhao, Jinyu Li, Jian Xue, and Yifan Gong. Investigating online low-footprint speaker adaptation using generalized linear regression and click-through data. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4310–4314. IEEE, 2015.

[94] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide. KL-divergence reg-

ularized deep neural network adaptation for improved large vocabulary speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7893–7897. IEEE, 2013.

[95] Daisuke Saito, Nobuaki Minematsu, and Keikichi Hirose. Rotational properties of vocal tract length difference in cepstral space. *Journal of Research Institute of Signal Processing*, Vol. 15, No. 5, pp. 363–374, 2011.

[96] Duc Hoang Ha Nguyen, Xiong Xiao, Eng Siong Chng, and Haizhou Li. Feature adaptation using linear spectro-temporal transform for robust speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 24, No. 6, pp. 1006–1019, 2016.

[97] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, Vol. 22, No. 10, pp. 1533–1545, 2014.

[98] Keisuke Kameyama, Kenzo Mori, and Yukio Kosugi. A neural network incorporating adaptive Gabor filters for image texture classification. In *Proceedings of the international conference on neural networks*, pp. 1523–1528, 1997.

[99] Alain Biem, Shigeru Katagiri, Erik McDermott, and Biing-Hwang Juang. An application of discriminative feature extraction to filter-bank-based speech recognition. *IEEE Transactions on Speech and Audio Processing*, Vol. 9, No. 2, pp. 96–110, 2001.

[100] RD Patterson, Ian Nimmo-Smith, John Holdsworth, and Peter Rice. An efficient auditory filterbank based on the Gammatone function. In *a meeting of the IOC Speech Group on Auditory Modelling at RSRE*, Vol. 2, 1987.

[101] Aniruddha Adiga, Mathew Magimai, and Chandra Sekhar Seelamantula. Gammatone wavelet cepstral coefficients for robust speech recognition. In *TENCON 2013-2013 IEEE Region 10 Conference (31194)*, pp. 1–4. IEEE, 2013.

[102] Jun Qi, Dong Wang, Yi Jiang, and Runsheng Liu. Auditory features based on gammatone filters for robust speech recognition. In *IEEE international symposium on Circuits and Systems (ISCAS)*, pp. 305–308. IEEE, 2013.

[103] Vikramjit Mitra, Wen Wang, Horacio Franco, Yun Lei, Chris Bartels, and Martin Graciarena. Evaluating robust features on deep neural networks for speech recognition in noisy and channel mismatched conditions. In *Proc. Interspeech*, 2014.

[104] Hector NB Pinheiro, Fernando MP Neto, Adriano LI Oliveira, Tsang Ing Ren,

George DC Cavalcanti, and André G Adami. Optimizing speaker-specific filter banks for speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5350–5354. IEEE, 2017.

[105] Christophe Charbuillet, Bruno Gas, Mohamed Chetouani, and Jean-Luc Zarader. Filter bank design for speaker diarization based on genetic algorithms. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. I–I. IEEE, 2006.

[106] Takumi Kobayashi and Jiaxing Ye. Discriminatively learned filter bank for acoustic features. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 649–653. IEEE, 2016.

[107] Lukáš Burget and Hynek Heřmanský. Data driven design of filter bank for speech recognition. In *International Conference on Text, Speech and Dialogue*, pp. 299–304. Springer, 2001.

[108] Youngjoo Suh and HoiRin Kim. Data-driven filter-bank-based feature extraction for speech recognition. In *SPECOM2004*, pp. 154–157, 2004.

[109] Yuji Tokozume and Tatsuya Harada. Learning environmental sounds with end-to-end convolutional neural network. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2721–2725. IEEE, 2017.

[110] Ting-Wei Su, Jen-Yu Liu, and Yi-Hsuan Yang. Weakly-supervised audio event detection using event-specific gaussian filters and fully convolutional networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 791–795, 2017.

[111] Tara N Sainath, Brian Kingsbury, Abdel-rahman Mohamed, and Bhuvana Ramabhadran. Learning filter banks within a deep neural network framework. In *IEEE workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 297–302. IEEE, 2013.

[112] Kikuo Maekawa. Corpus of spontaneous Japanese: Its design and evaluation. In *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[113] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*, Vol. 5, pp. 1–6, 2015.

[114] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint*

arXiv:1502.03167, 2015.

[115] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, No. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[116] Dong Yu and Jinyu Li. Recent progresses in deep learning based acoustic models. *IEEE/CAA Journal of Automatica Sinica*, Vol. 4, No. 3, pp. 396–409, 2017.

[117] Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach. An overview of noise-robust automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 22, No. 4, pp. 745–777, 2014.

[118] Zixing Zhang, Jürgen Geiger, Jouni Pohjalainen, Amr El-Desoky Mousa, Wenyu Jin, and Björn Schuller. Deep learning for environmentally robust speech recognition: An overview of recent developments. *ACM Transactions on Intelligent Systems and Technology (TIST)*, Vol. 9, No. 5, p. 49, 2018.

[119] Hynek Hermansky, Nelson Morgan, Aruna Bayya, and Phil Kohn. Compensation for the effect of the communication channel in auditory-like analysis of speech (rasta-plp). In *Second European Conference on Speech Communication and Technology*, 1991.

[120] Chanwoo Kim and Richard M Stern. Feature extraction for robust speech recognition based on maximizing the sharpness of the power distribution and on power flooring. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4574–4577. IEEE, 2010.

[121] Niko Moritz, Marc René Schädler, Kamil Adiloglu, Bernd T Meyer, Tim Jürgens, Timo Gerkmann, Birger Kollmeier, Simon Doclo, and Stefan Goetze. Noise robust distant automatic speech recognition utilizing nmf based source separation and auditory feature extraction. *Proc. of CHiME*, pp. 1–6, 2013.

[122] Sirko Molau, Florian Hilger, and Hermann Ney. Feature space normalization in adverse acoustic conditions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 656–659. IEEE, 2003.

[123] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Deep learning for monaural speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1562–1566. IEEE, 2014.

[124] Steven Boll. Suppression of acoustic noise in speech using spectral subtraction.

IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 27, No. 2, pp. 113–120, 1979.

[125] Jae S Lim and Alan V Oppenheim. Enhancement and bandwidth compression of noisy speech. *Proceedings of the IEEE*, Vol. 67, No. 12, pp. 1586–1604, 1979.

[126] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori. Speech enhancement based on deep denoising autoencoder. In *Proc. Interspeech*, pp. 436–440, 2013.

[127] Xue Feng, Yaodong Zhang, and James Glass. Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1759–1763. IEEE, 2014.

[128] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee. Dynamic noise aware training for speech enhancement based on deep neural networks. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[129] Michael L Seltzer, Dong Yu, and Yongqiang Wang. An investigation of deep neural networks for noise robust speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7398–7402. IEEE, 2013.

[130] Arun Narayanan and DeLiang Wang. Joint noise adaptive training for robust automatic speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2504–2508. IEEE, 2014.

[131] Ritwik Giri, Michael L Seltzer, Jasha Droppo, and Dong Yu. Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5014–5018. IEEE, 2015.

[132] Yusuke Shinohara. Adversarial multi-task learning of deep neural networks for robust speech recognition. In *Proc. Interspeech*, pp. 2369–2372, 2016.

[133] Sining Sun, Ching-Feng Yeh, Mei-Yuh Hwang, Mari Ostendorf, and Lei Xie. Domain adversarial training for accented speech recognition. *arXiv preprint arXiv:1806.02786*, 2018.

[134] Jae S Lim and Alan V Oppenheim. Enhancement and bandwidth compression of noisy speech. *Proceedings of the IEEE*, Vol. 67, No. 12, pp. 1586–1604, 1979.

[135] Yariv Ephraim and David Malah. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 32, No. 6, pp. 1109–1121, 1984.

[136] Michael L Seltzer, Dong Yu, and Yongqiang Wang. An investigation of deep

neural networks for noise robust speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7398–7402. IEEE, 2013.

[137] Andrew Varga and Herman JM Steeneken. Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech communication*, Vol. 12, No. 3, pp. 247–251, 1993.

[138] Ondrej Klejch, Joachim Fainberg, and Peter Bell. Learning to adapt: a meta-learning approach for speaker adaptation. In *Proc. Interspeech*, 2018.

[139] Alison Behrman. *Speech and voice science*. Plural publishing, 2017.

[140] Michael Kleinschmidt. Localized spectro-temporal features for automatic speech recognition. In *Eighth European Conference on Speech Communication and Technology*, 2003.

[141] Shuo-Yiin Chang and Nelson Morgan. Robust CNN-based speech recognition with Gabor filter kernels. In *Proc. Interspeech*, 2014.

[142] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. A time delay neural network architecture for efficient modeling of long temporal contexts. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[143] IPA-98-TestSet. http://winnie.kuis.kyoto-u.ac.jp/pub/julius/result99/node1.html.

Publication list

- Papers/Journals with Referee's Review
  1. Hiroshi Seki, Daisuke Enami, Faqiang Zhu, Kazumasa Yamamoto, and Seiichi Nakagawa, "Speech recognition of short time utterance based on speaker clustering," IEICE Trans. Inf & Syst., Vol. J100-D, No. 1, pp. 81-92, 2017 (in Japanese).
  2. Hiroshi Seki, Kazumasa Yamamoto, Tomoyosi Akiba, and Seiichi Nakagawa, "Discriminative learning of filterbank layer within deep neural network based speech recognition for speaker adaptation," IEICE Trans. Inf & Syst., Vol. 102, No. 2, pp. 364-374, 2019.
- International Conference with Referee's Review
  1. Hiroshi Seki, Kazumasa Yamamoto, Tomoyosi Akiba, and Seiichi Nakagawa, "Rapid speaker adaptation of neural network based filterbank layer for automatic speech recognition," IEEE Spoken language technology workshop (SLT), pp. 574-580, 2018.
  2. Hiroshi Seki, Takaaki Hori, Shinji Watanabe, Jonathan Le Roux, John R Hershey, "A purely end-to-end system for multi-speaker speech recognition," 56th Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers, pp. 2620-2630, 2018.
  3. Hiroshi Seki, Shinji Watanabe, Takaaki Hori, Jonathan Le Roux, John R. Hershey, "An end-to-end language-tracking speech recognizer for mixed-language speech," in Proc. International conference on acoustics, speech and signal processing (ICASSP), pp. 4919-4923, 2018.
  4. Koya Sahashi, Norioki Goto, Hiroshi Seki, Kazumasa Yamamoto, Tomoyoshi Akiba, Seiichi Nakagawa, "Robust lecture speech translation for speech misrecognition and its rescoring effect from multiple candidates," in Proc. International conference on advanced informatics: concepts,

theory and applications (ICAICTA), 2017.

5. Hiroshi Seki, Kazumasa Yamamoto, Seiichi Nakagawa, "A deep neural network integrated with filterbank learning for speech recognition," in Proc. International conference on acoustics, speech and signal processing (ICASSP), pp. 4919-4923, 2017.
6. Hiroshi Seki, Kazumasa Yamamoto, Seiichi Nakagawa, "Deep neural network based acoustic modeling using speaker-class information for short-time utterance," in Proc. Asia-Pacific signal and information processing association annual summit and conference (APSIPA), pp. 1222-1225, 2015.
7. Hiroshi Seki, Kazumasa Yamamoto, Seiichi Nakagawa, "Comparison of syllable-based and phoneme-based DNN-HMM in Japanese speech recognition," in Proc. International conference on advanced informatics: concepts, theory and applications (ICAICTA), pp. 249-254, 2014.

- ArXiv
  1. Hiroshi Seki, Takaaki Hori, Shinji Watanabe, "Vectorization of hypotheses and speech for faster beam search in encoder decoder-based speech recognition," arXiv preprint arXiv:1811.04568, 2018 (submitted to International conference on acoustics, speech and signal processing (ICASSP)).
- Domestic Conference (in Japanese)
  1. Hiroshi Seki, Kazumasa Yamamoto, Tomoyosi Akiba, Seiichi Nakagawa, "Noise adaptation of filterbank feature under end-to-end encoder decoder networks," Acoustic Society of Japan, Spring meeting, 2019 (To be appear).
  2. Hiroshi Seki, Kazumasa Yamamoto, Tomoyosi Akiba, and Seiichi Nakagawa, "Evaluation of filterbank incorporated DNN and comparison of filter functions using Corpus of Spontaneous Japanese," Acoustic Society of Japan, Autumn meeting, 1-R-13, pp. 83-86, 2017.
  3. Hiroshi Seki, Kazumasa Yamamoto, Tomoyosi Akiba, and Seiichi Nakagawa, "Investigation of speaker class adaptation with filterbank learning based on DNN," Acoustic Society of Japan, Spring meeting, 1-Q-7, pp. 101-104, 2017.
  4. Hiroshi Seki, Kazumasa Yamamoto, and Seiichi Nakagawa, "Investigation of DNN-based filterbank learning for speech recognition," Acoustic Society of Japan, Autumn meeting, 3-Q-7, pp. 101-104, 2016.
  5. Hiroshi Seki, Kazumasa Yamamoto, and Seiichi Nakagawa, "Investigation of unsupervised iterative adaptive training for convolutional neural net-

work," Acoustic Society of Japan, Spring meeting, 3-P-3, pp. 155-158, 2016.

6. Hiroshi Seki, Kazumasa Yamamoto, and Seiichi Nakagawa, "Investigation on deep neural network based HMM for speech recognition based on age- and gender-independent clustering information," Acoustic Society of Japan, Spring meeting, 1-P-28, pp. 175-178, 2015.
7. Hiroshi Seki, Kazumasa Yamamoto, and Seiichi Nakagawa, "Consideration on age- and gender-independent speech recognition using DNN-HMM," IPSJ SIG technical report, Vol. 2014-SLP-104, No. 29, pp. 1-6, 2014.
8. Hiroshi Seki, and Seiichi Nakagawa, "Evaluation of syllable-unit based deep neural network for speech recognition," Acoustic Society of Japan, Spring meeting, 2-Q4-25, pp. 179-182, 2014.

Appendix A

Evaluation of filterbank layer on a small-size corpus

A.1 Introduction

In the Section 4, we reported the evaluation of neural network based filterbank layer using a middle size corpus, CSJ. We also conducted the evaluation on ASJ+JNAS corpus in addition to the CSJ corpus. We report the experimental results on ASJ+JNAS as an appendix.

A.2 Experimental work

A.2.1 Experimental setup

We used the ASJ+JNAS corpus [86, 87] for the validation. Details of the corpus are summarized in Table A.1. The corpus consists of 33 hours of male speakers' speech (Speakers-Male; SM) and 44 hours of female speakers' speech (Speakers-Female; SF). The numbers of speakers for SM and SF were 133 and 164, respectively. For the evaluation as the speaker independent model, we used an IPA 100 test-set [143] consisting of 100 utterances uttered by 23 male speakers and 100 utterances uttered by 23 female speakers. For the evaluation of speaker adaptation, 13 sentences per speaker were chosen from ASJ+JNAS whose speakers were not included in the training data. We used 23 male speakers, the same as the IPA 100 test-set, and all 13 sentences were split into adaptation and test data. We assigned three (or five) utterances for the adaptation data, and eight utterances for the test data. In total, the adaptation data consists of 115 (or 69) utterances (5 or 3 utterances × 23 speakers), and the test data consists of 184 utterances (8 utterances × 23 speakers ≈ 0.34 hours).

The architecture of networks are almost same as the experimental setup on CSJ. Please refer to Chapter 4.5 for the detailed setup.

GMM-HMM

To obtain the target labels for DNNs, GMM-HMMs are trained using SM, SF, and SM plus SF (mixed). The models are trained on standard MFCC features. The senones of SM, SF, and SM plus SF are set to 3234. The corresponding GMM-HMMs are used for forced alignment.

Baseline DNN (Triangle filterbank)

As a baseline system, we trained a fully connected DNN which has four hidden layers with 1,024 rectified linear units [55, 56]. Its input is 11 consecutive frames of 40-dimensional log mel-scale triangle-shape filterbank features extracted using the Hidden Markov Model Toolkit (HTK) [43]. The features are normalized to zero mean and unit variance.

Gaussian filterbank incorporated DNN (GFDNN)

The Gaussian filterbank layer was inserted to the bottom of the baseline DNN. Its input was 11 consecutive frames of 256-dimensional power spectra. The number of filters was set to 40, which is the same as the baseline system. The initial values of the gain parameter were set to 1.0. The center frequencies were spaced equally along the mel-scale. The bandwidths were set so that the two-sigma range equals the corresponding bandwidth of the mel-scale filterbank.

Gammatone filterbank incorporated DNN (GtFDNN)

The Gammatone filterbank layer was inserted to the bottom of the baseline DNN. The other setup was the same as GFDNN.

Learning hidden unit contribution (LHUC)

To adapt the models, we re-scaled the hidden units using LHUC. In the exper-

Table. A.1 Details of ASJ+JNAS corpus.

	Gender	Male (SM)	Female (SF)
Train	Speakers	133	164
	Data	33 hours	44 hours
Test	Speakers	23	23
	Data	0.2 hours	0.2 hours

iment, the hidden units of the third layer were adapted which showed the best performance in the preliminary experiment. I.e., the 1,024 parameters were updated at the adaptation stage.

Singular value decomposition (SVD)

SVD was applied on the 4th fully connected layer and kept top 300 singular values,

As a language model, a tri-gram word-based language model was trained on the Mainichi newspaper corpus (11,533,739 words, vocabulary of 20,000 words) [91]. As a decoder, we used the SPOJUS++ (SPOken Japanese Understanding System) WFST version [92].

A.2.2 ASJ+JNAS results

Speaker independent model

Table A.2 shows the WERs of the speaker-independent models. The baseline gender dependent DNN, which takes triangular filterbanks, achieved a WER of 4.9% for SM and 5.0% for SF speakers. The average WER was 5.0%. When we focus on the baseline models and the *fixed* (untrained) models, the latter outperformed the baseline models in all cases, even though the filter shape was the only difference between the two models. This difference changes the coverage of the frequency bin. The Gaussian and Gammatone filters focus on all the frequency bins while the baseline triangular filter zeroes out the frequency bins outside a certain bin distance. These results comparing fixed and baseline model showed the importance of refined acoustic features. Mitra, et al., also investigated the effectiveness of robust features for DNN

Table. A.2 WERs (%) of the baseline DNN and the filterbank-incorporated DNNs (matched condition). OOVs of SM and SF are 0.5% and 0.5%. Perplexities of SM and SF are both 125.7.

System	WER [%]			
	SM	SF	Ave.	SM+SF
Baseline (Triangle)	4.9	5.0	5.0	5.0
GFDNN (fixed)	4.3	4.8	4.5	4.1
GFDNN (trained)	4.1	4.7	4.4	4.1
GtFDNN (fixed)	4.8	4.5	4.7	4.0
GtFDNN (trained)	4.7	4.1	4.4	4.0
ExpFDNN	5.1	5.1	5.1	4.1

including (non data-driven) Gammatone filterbank feature [103]. The performance improvement of the fixed models correspond with their results.

In the case of gender dependent models, the average WERs of the *trained* GFDNN and GtFDNN were 4.4%. These systems outperformed the baseline DNN of 5.0%. In addition, the optimization of the filter shapes improved the recognition performance. These results indicate that the discriminatively trained filterbank layer improved recognition performance. The GFDNN and GtFDNN showed consistent improvement. The Gammatone filter is widely used as an auditory filter. However, the difference between filter types did not show any performance improvement. In the case of gender independent models (SM+SF), the trained models did not show performance improvement. We considered that the difference of optimal center frequencies between male and female speakers made it difficult to learn universal center frequencies for both male and female speakers. In the following experiment, we only present the results of the trained models.

Gender adaptation

In this section, we evaluated gender adaptation from SM to SF, and confirmed the presence of the filters' shift for the alleviation of the vocal tract lengths. Table A.3 shows the WERs of gender adaptation. The first column is the duration of female speech data for adaptation. The row of 0 utterance is the WERs of the model without adaptation. The WERs of SM specific GFDNN and GtFDNN were worse at 41.6% and 33.6%, due to the gender mismatched condition. For the evaluations of 10 and 20 utterances, 60 utterances in Table A.3 were split into six or three folds, and averaged

Table. A.3 WERs (%) of the gender adaptation from SM to SF speakers. Bold is the best performance among models.

Adaptation data	# speakers	GFDNN	GtFDNN				ExpFDNN
		filterbank	filterbank	fDLR	LHUC	SVD	filterbank
0.0 h	0	41.6	33.6	33.6	33.6	**31.3**	44.1
0.02 h	10	14.8	19.1	13.9	**11.5**	21.2	26.0
0.03 h	20	19.5	14.2	13.2	**10.1**	18.4	23.0
0.1 h	60	11.0	10.4	11.2	**8.2**	9.8	9.6
1.0 h	164	6.4	7.1	5.7	6.9	**5.6**	5.7
10.0 h	164	**4.7**	5.0	6.1	5.5	4.8	4.8
30.0 h	164	**4.3**	5.7	6.1	5.4	4.8	4.5

Table. A.4 Shift of center frequencies [Hz] caused by gender adaptation from SM to SF using 10 hours of training data. SM → SF shows the center frequencies of un-adapted and adapted models. SF column shows the center frequencies of SF-specific model trained using SF speech data.

	SM → SF			**SF**
n	Before Adaptation	After Adaptation	Difference	-
6	312.0	336.0	24.0 (7.7%)	306.0
7	375.0	392.0	17.0 (4.5%)	365.0
8	438.0	457.0	19.0 (4.3%)	433.0
9	529.0	549.0	20.0 (3.8%)	515.0
10	595.0	617.0	22.0 (3.7%)	578.0
11	695.0	712.0	17.0 (2.4%)	681.0
12	780.0	799.0	19.0 (2.4%)	783.0
13	872.0	889.0	17.0 (1.9%)	875.0
14	964.0	982.0	18.0 (1.9%)	963.0
15	1060.0	1070.0	10.0 (0.9%)	1054.0

to alleviate any selection bias. By adapting the filterbank of GtFDNN using 0.02 hour of adaptation data, the WER improved from 33.6% to 19.1%. We can see that the adjustment of the filterbank layer can deal with the mismatch caused by the vocal tract length. However, against our expectation, GtFDNN model based on LHUC adaptation obtained the best performance under low-resource adaptation data scenario. The WERs of GFDNN were further improved by increasing the size of adaptation data. As presented in Table A.2, the WER of SF-dependent GFDNN trained by 44 hours of data was 4.7%. This result was identical to the adapted GFDNN, which was adapted using 10 hours of speech data.

By adapting GFDNN from SM to SF, we considered that a frequency shift of the filters is caused by the differences of the vocal tract lengths. Table A.4 shows the relation among the center frequencies of SM-dependent GFDNN, adapted GFDNN from SM to SF using 10 hours of training data, and SF-dependent GFDNN. Theoretically, an ideal frequency shift is approximately 9.7%, as described in Section 4.4.2. Assuming that the standard deviation of the male vocal tract length is 1.0 cm, the SM-dependent DNN might learn speech ranging from 16.0 to 18.0 cm of the vocal tract length. The SM-dependent DNN with a 6.0% shift of the center frequencies may be adapted to the female speech by additionally assuming that the standard deviation of female vocal tract length is 0.5 cm. The column of difference shows the actual shift was approximately 0.9% to 7.7%, which resembles the theoretical value. We can see that the optimization of the filterbank layer caused a shift of center frequencies to discriminatively perform frequency warping. This characteristic corresponds to the VTLN function.

The last column of Table A.4 shows the center frequencies of the SF-dependent GFDNN. When we focused on the SM- and SF-dependent GFDNN, relations between the two models cannot be observed in the experiment. Instead, the learned center frequencies based on SF speakers showed lower frequencies than those of the SM speakers because the optimal position of the filters in the training stage depends of the condition of the following DNN. However, in the adaptation stage, the filterbank layer was updated, and the parameters of the following DNN were fixed. In this situation, the filterbank layer could be handled independently of the following DNN to perform frequency warping.

Figure A.1 shows the change of gain parameters of the SM-specific model and the SF-adapted model (from the SM-specific model). To emphasize the conspicuous change of gains, we plotted their relative changes by computing $(gains\_of\_SF - gains\_of\_SM)/gains\_of\_SM$. We also plotted the average log mel-scale triangular filterbank features of the SM-speakers and SF-speakers. Intuitively, the relative change of gain takes a negative value when the filterbank feature of the SF-speakers takes a higher amplitude value than that of the SM-speakers (channels 31-40), and vice versa (channels 1-3). The difference of the filterbank features partially satisfies this assumption but not completely. The variances of the filterbank features are relatively large enough to overlap the SM- and SF-speakers. Therefore, it is considered that the optimization of the gain parameters is a secondarily important factor in gender adaptation. In contrast to the above two parameters, no discrimination of the change of bandwidth parameters was observed.

Speaker adaptation

In this section, we evaluated speaker adaptation using the SM-dependent models depicted in Table A.2. Table A.5 shows the speaker adaptation results. The row of 0 utterance shows the WERs, which were recognized using the model without speaker adaptation. These WERs were worse than the result of Table A.2 because of its worse out of vocabulary ratio (OOV) and larger perplexity. By adapting the filterbank layer of GtFDNN using five utterances, the WER improved from 8.9% to

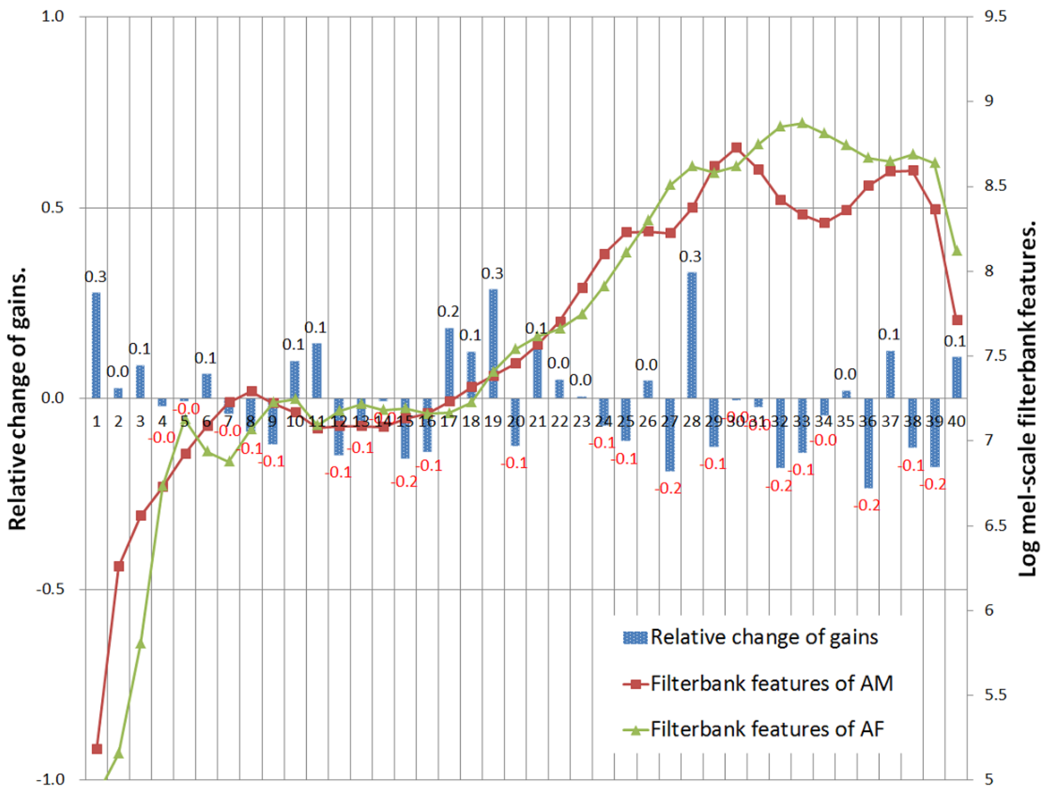


Fig. A.1 Changes of gain parameters from SM-specific model (SM) to SF-adapted model and averaged log mel-scale filterbank features of SM- and SF-speakers.

Table. A.5 WERs (%) of speaker adaptation. (OOV: 2.3%, Perplexity: 161.4, Corpus: SM).

#utt.	GFDNN	GtFDNN	DNN			ExpFDNN
	filterbank	filterbank	fDLR	LHUC	SVD	filterbank
0	9.1	**8.9**	10.0	10.0	10.0	9.5
3	9.0	**8.5**	9.1	13.5	9.7	8.6
5	8.7	**8.2**	9.7	13.3	9.6	8.4

8.2%, and a word error reduction rate (WERR) of 7.9% was obtained. This WERR is better than the unadapted GtFDNN at a significance level of 0.012 under a statistical sign test. The performance improvement was also observed in the experiment of GFDNN. These results showed that the adjustment of filter shapes can handle the diversity of speakers. The adaptation of filterbank layer in GtFDNN showed the best performance for all adaptation conditions while other methods, ExpFDNN, fDLR, LHUC and SVD, also showed performance improvement. Table A.6 shows the WER of GtFDNN with comparative adaptation methods. There was no significance among

Table. A.6 WERs (%) of speaker adaptation. (OOV: 2.3%, Perplexity: 161.4, Corpus: SM).

#utt.	GtFDNN			
	filterbank	fDLR	LHUC	SVD
0	**8.9**	**8.9**	**8.9**	9.2
3	**8.5**	8.6	8.6	8.9
5	**8.2**	8.4	8.3	8.5

them, however, adaptation of filterbank showed the best performance.