Interspeech 2018
2-6 September 2018, Hyderabad

# DNN-based Speech Synthesis for Small Data Sets Considering Bidirectional Speech-Text Conversion

*Kentaro Sone, Toru Nakashika*

Graduate School of Informatics and Engineering,
The University of Electro-Communications, Japan
sone@sd.is.uec.ac.jp, nakashika@uec.ac.jp

## Abstract

In statistical parametric speech synthesis, approaches based on deep neural networks (DNNs) have improved qualities of the synthesized speech. General DNN-based approaches require a large amount of training data to synthesize natural speech. However, it is not practical to record speech for many hours from a single speaker. To address this problem, this paper presents a novel pre-training method of DNN-based speech synthesis systems for small data sets. In this method, a Gaussian-Categorical deep relational model (GCDRM), which represents a joint probability of two visible variables, is utilized to describe the joint distribution of acoustic features and linguistic features. During the maximum-likelihood-based training, the model attempts to obtain parameters of a deep architecture considering the bidirectional conversion between 1) generated acoustic features given linguistic features and 2) re-generated linguistic features given acoustic features generated from itself. Owing to considering whether the generated acoustic features are recognizable, our method can obtain reasonable parameters from small data sets. Experimental results show that pre-trained DNN-based systems using our proposed method outperformed randomly-initialized DNN-based systems. This method also outperformed DNN-based systems in a speaker-dependent speech recognition task.

**Index Terms**: speech synthesis, Boltzmann distribution, pre-training method

## 1. Introduction

In the last decade, statistical parametric speech synthesis using hidden Markov models (HMMs) [1] has been investigated. HMM-based approaches have various advantages over the concatenative speech synthesis systems [2], such as the capability to model spectrum, pitch and state duration simultaneously in a unified framework [3], the flexibility to convert voice characteristics by changing HMMs parameters [4]. However, it is reported that the quality of the synthesized speech using HMMs is problematic [5]. One of the reasons is that the decision-tree-clustered context-dependent HMMs have a limitation that it is inefficient to represent complex context dependencies. To address this issue, Zen *et al.* [6] proposed an alternative scheme to replace a decision tree with a deep neural network (DNN). It is reported that the DNN-based system improved the quality of the synthesized speech.

When it comes to recent DNN-based approaches, A. van den Oord *et al* [7] proposed WaveNets that attempt to model raw speech waveforms directly in an end-to-end framework. This approach has been reported that qualities of synthesized speech are improved over parametric approaches thanks to modeling raw waveforms directly and representing recurrent dependencies of speech. Furthermore, in order to accelerate the training procedure, several approaches [8, 9] have been proposed in this frame work. However, these approaches still take large costs in the training and synthesis stages compared with statistical or frame-wise approaches.

In any case, to represent complex feedforward dependencies, a DNN has been reported its effectiveness in the various domains (*e.g.* image recognition, speech recognition and natural language processing). However, a large amount of training data is generally required to optimize the parameters of a DNN. In speech synthesis, though it is necessary to record speech for many hours from a single speaker for a DNN-based system, that is not practical. In order to construct the DNN-based speech synthesis system for small data sets, we focus on a "cyclic training" which takes into account whether generated acoustic features are recognizable during a training stage. The cyclic training is the maximum-likelihood-based training that models a bidirectional (feedforward and backward) conversion. The feedforward conversion represents the dependencies from inputs to outputs of a DNN. The backward conversion represents the dependencies from the predicted outputs from a DNN to inputs. In this paper, we attempt to extract the bidirectional conversion between text and speech.

In the domain of binary-valued image classification and generation, Nakashika [10] proposed a deep relational model (DRM), which can potentially classify and generate binary-valued images. The DRM models a joint distribution of the two variables and contains multiple hidden layers to capture their latent dependencies of those. In this paper, we define a Gaussian-Categorical DRM (GCDRM) to apply DRM concepts to the domain of speech synthesis, and propose a GCDRM-based pre-training method for DNN-based speech synthesis systems.

## 2. Deep Relational Model

The same as a restricted Boltzmann machine (RBM) [11] and a deep Boltzmann machine (DBM) [12], a DRM is an undirected graphical model with a set of visible and hidden units [10]. A DRM consists of two visible layers (the first visible variables $\boldsymbol{x} \in \{0,1\}^I$ and the second visible variables $\boldsymbol{y} \in \{0,1\}^K$) and multiple hidden variables $\boldsymbol{h}^{(l)} \in \{0,1\}^{J_l} (l = 1, ...., L)$, where $L$ is the number of hidden layers. A DRM has symmetric connections between the units in adjacent layers and no connections between the units in the same layer. A DRM is defined on the basis of the energy function to capture high-order relationships between two observable variables $\boldsymbol{x}$ and $\boldsymbol{y}$. The joint probability distribution using a DRM is defined as follows:

$$p(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) = \sum_{\forall \boldsymbol{h}^{(l)}} p(\boldsymbol{x}, \boldsymbol{y}, \forall \boldsymbol{h}^{(l)}; \boldsymbol{\theta}) \quad (1)$$

$$p(\boldsymbol{x}, \boldsymbol{y}, \forall \boldsymbol{h}^{(l)}; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\{-E(\boldsymbol{x}, \boldsymbol{y}, \forall \boldsymbol{h}^{(l)}; \boldsymbol{\theta})\}, \quad (2)$$



10.21437/Interspeech.2018-1460

where $Z$ is the partition function. In a DRM, the energy function $E$ is defined as:

$$E(\boldsymbol{x}, \boldsymbol{y}, \forall \boldsymbol{h}^{(l)}; \boldsymbol{\theta}) = -\boldsymbol{b}^T \boldsymbol{x} - \sum_{l=1}^{L} \boldsymbol{c}^{(l)T} \boldsymbol{h}^{(l)} - \boldsymbol{d}^T \boldsymbol{y} - \boldsymbol{x}^T \boldsymbol{W}^{(1)} \boldsymbol{h}^{(1)} - \sum_{l=2}^{L} \boldsymbol{h}^{(l-1)T} \boldsymbol{W}^{(l)} \boldsymbol{h}^{(l)} - \boldsymbol{h}^{(L)T} \boldsymbol{W}^{(L+1)} \boldsymbol{y},$$

where $\boldsymbol{b} \in \mathbb{R}^I$, $\boldsymbol{c}^{(l)} \in \mathbb{R}^{J_l}$ and $\boldsymbol{d} \in \mathbb{R}^K$ are the bias parameters corresponding to the units in the first visible layer, the $l$th hidden layer and the second visible layer. $\boldsymbol{W}^{(1)} \in \mathbb{R}^{I \times J_1}$, $\boldsymbol{W}^{(l)} \in \mathbb{R}^{J_{l-1} \times J_l}$ and $\boldsymbol{W}^{(L+1)} \in \mathbb{R}^{J_L \times K}$ are the weight parameters of connections between the first visible layer and the first hidden layer, $(l-1)$th hidden layer and $l$th hidden layer and $L$th hidden layer and the second visible layer, respectively.

Under the definition of the energy function, the conditional distributions for each visible and hidden unit given adjacent units are

$$p(x_i = 1|\boldsymbol{h}^{(1)}) = \sigma(b_i + \boldsymbol{W}^{(1)}_{i:} \boldsymbol{h}^{(1)}) \quad (3)$$

$$p(h^{(l)}_j = 1|\boldsymbol{h}^{(l-1)}, \boldsymbol{h}^{(l+1)}) = \sigma(c^{(l)}_j + \boldsymbol{W}^{(l)T}_{:j} \boldsymbol{h}^{(l-1)} + \boldsymbol{W}^{(l+1)}_{j:} \boldsymbol{h}^{(l+1)}) \quad (4)$$

$$p(y_k = 1|\boldsymbol{h}^{(L)}) = \sigma(d_k + \boldsymbol{W}^{(L)T}_{:k} \boldsymbol{h}^{(L)}), \quad (5)$$

where $\sigma(\cdot)$ denotes the logistic sigmoid function. Note that the hidden variables $\boldsymbol{h}^{(0)}$ and $\boldsymbol{h}^{(L+1)}$ are regarded as $\boldsymbol{h}^{(0)} = \boldsymbol{x}$ and $\boldsymbol{h}^{(L+1)} = \boldsymbol{y}$, respectively, in Eq. (4).

The parameters of a DRM $\boldsymbol{\theta} = \{\boldsymbol{b}, \boldsymbol{c}^{(l)}, \boldsymbol{d}, \boldsymbol{W}^{(1)}, \boldsymbol{W}^{(l)}, \boldsymbol{W}^{(L+1)}\}$ are optimized to maximize the joint log-likelihood $\mathcal{L} = \log \prod_t p(\boldsymbol{x}^t, \boldsymbol{y}^t; \boldsymbol{\theta})$. The partial derivative of $\mathcal{L}$ with respect to $\boldsymbol{\theta}$ is computed as:

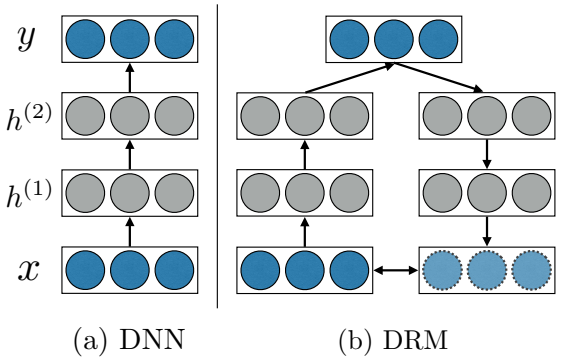
$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \left\langle -\frac{\partial E}{\partial \boldsymbol{\theta}} \right\rangle_d - \left\langle -\frac{\partial E}{\partial \boldsymbol{\theta}} \right\rangle_m, \quad (6)$$

where shorthand notations $\langle \cdot \rangle_d$ and $\langle \cdot \rangle_m$ denote the expectations computed over the data and model distributions, respectively. The training is described in more detail by Nakashika [10].

## 3. Applying DRM concepts to text and speech

In this section, we introduce our model, a Gaussian-Categorical DRM (GCDRM), whose visible layer consists of binary-valued and real-valued units. To distinguish a traditional DRM, which is described in Section 2, and a GCDRM explicitly, we refer to the former as a Bernoulli-Bernoulli DRM (BBDRM).

The DNN-based speech synthesis systems focus on converting a linguistic feature vector $\boldsymbol{w}^t$ into an acoustic feature vector $\boldsymbol{a}^t$ at the frame $t$. The input to the DNN is linguistic features, which contain binary values and real values. The binary values indicate the phoneme labels as a one-hot vector, and the real values indicate linguistic contexts of a given text (*e.g.* the relative position of the current frame in the current phoneme and durations of the current phoneme). Meanwhile, the output from the DNN is acoustic features, which include spectral parameters, and each element of $\boldsymbol{a}^t$ is real-valued. This approach represents feedforward dependencies from inputs to outputs (Fig. 1 (a)), and attempts to minimize the error between




(a) DNN (b) DRM

Figure 1: *Graphical representation of (a) a feedforward training of a DNN, and (b) a cyclic training of a DRM. Dotted circles indicate the generated $\boldsymbol{x}$ obtained from the generated $\boldsymbol{y}$.*

$\boldsymbol{y}$ in the training data and predicted outputs in the training stage. In order to improve the performance of the DNN-based speech synthesis systems, we propose a cyclic training method using a DRM (Fig. 1 (b)). In our approach, our method attempts to capture not only the feedforward dependencies (from $\boldsymbol{x}$ to $\boldsymbol{y}$), but also backward dependencies (from generated $\boldsymbol{y}$ to $\boldsymbol{x}$).

Since a BBDRM has been developed to model bidirectional relationships between two binary variables, it is not suitable to model bidirectional relationships between linguistic features and acoustic features. To address this issue, we propose a GC-DRM, which represents two mixed distributions such as 1) the categorical distributions for phonemes in linguistic features and 2) the Gaussian distributions for linguistic contexts in linguistic features and acoustic features.

In this paper, we focus on the frame-wise modeling. Therefore, we omit the subscript $t$ unless otherwise noted in the remaining of the paper.

### 3.1. Gaussian-Categorical DRM

A Gaussian-Bernoulli RBM (GBRBM) [13] was originally proposed to model a real-valued data. Later, an improved GBRBM (IGBRBM) [14] has been proposed to improve training of GBRBMs, which is difficult due to the variance parameters. As a pre-training method for DBN-based speech synthesis, a Mixed GBRBM and a mixed categorical-Bernoulli RBM (Mixed CBRBM) [15] has been proposed. Referring to an IGBRBM, a Mixed GBRBM and a Mixed CBRBM, we define the energy function of a GCDRM as follows:

$$\begin{aligned} E(\boldsymbol{x}, \boldsymbol{y}, \forall \boldsymbol{h}^{(l)}; \boldsymbol{\theta}) = & \\ & \frac{1}{2} \left( \frac{\boldsymbol{x}^g - \boldsymbol{b}^g}{\boldsymbol{\sigma}^{(x)g}} \right)^T \left( \frac{\boldsymbol{x}^g - \boldsymbol{b}^g}{\boldsymbol{\sigma}^{(x)g}} \right) - \boldsymbol{b}^{cT} \boldsymbol{x}^c \\ & - \left( \frac{\boldsymbol{x}^g}{\boldsymbol{\sigma}^{(x)g} \circ \boldsymbol{\sigma}^{(x)g}} \right)^T \boldsymbol{W}^{(1)g} \boldsymbol{h}^{(1)} - \boldsymbol{x}^{cT} \boldsymbol{W}^{(1)c} \boldsymbol{h}^{(1)} \\ & - \sum_{l=1}^{L} \boldsymbol{c}^{(l)T} \boldsymbol{h}^{(l)} - \sum_{l=2}^{L} \boldsymbol{h}^{(l-1)T} \boldsymbol{W}^{(l)} \boldsymbol{h}^{(l)} \\ & + \frac{1}{2} \left( \frac{\boldsymbol{y}^g - \boldsymbol{d}^g}{\boldsymbol{\sigma}^{(y)g}} \right)^T \left( \frac{\boldsymbol{y}^g - \boldsymbol{d}^g}{\boldsymbol{\sigma}^{(y)g}} \right) - \boldsymbol{d}^{cT} \boldsymbol{y}^c \\ & - \boldsymbol{h}^{(L)T} \boldsymbol{W}^{(L+1)g} \left( \frac{\boldsymbol{y}^g}{\boldsymbol{\sigma}^{(y)g} \circ \boldsymbol{\sigma}^{(y)g}} \right) - \boldsymbol{h}^{(L)T} \boldsymbol{W}^{(L+1)c} \boldsymbol{y}^c, \end{aligned}$$

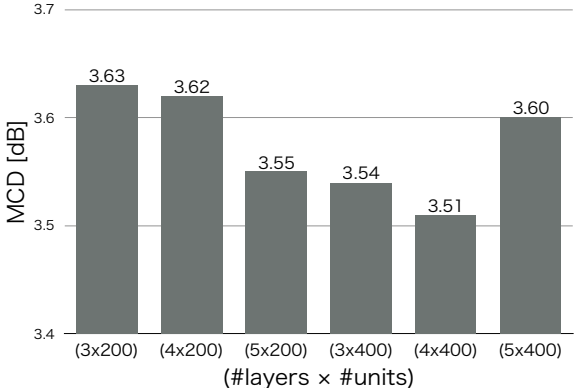



Figure 2: *Performance of our method when changing the numbers of hidden layers and hidden units at each hidden layer (MCD [dB]).*

where $\boldsymbol{x}^c \in \{0,1\}^{X^c}$ and $\boldsymbol{x}^g \in \mathbb{R}^{X^g}$ are the categorical units and the Gaussian units in the first visible layer, $\boldsymbol{y}^c \in \{0,1\}^{Y^c}$ and $\boldsymbol{y}^g \in \mathbb{R}^{Y^g}$ are the categorical units and Gaussian units in the second visible layer ($X^g + X^c = I$, $Y^g + Y^c = K$, $\boldsymbol{x} = [\boldsymbol{x}^{gT}\ \boldsymbol{x}^{cT}]^T$, $\boldsymbol{y} = [\boldsymbol{y}^{gT}\ \boldsymbol{y}^{cT}]^T$), $\boldsymbol{W}^{(1)c} \in \mathbb{R}^{X^c \times J_1}$, $\boldsymbol{W}^{(L+1)c} \in \mathbb{R}^{J_L \times Y^c}$, $\boldsymbol{b}^c \in \mathbb{R}^{X^c}$ and $\boldsymbol{d}^c \in \mathbb{R}^{Y^c}$ are the weight matrices and bias parameters corresponding to the categorical units, $\boldsymbol{W}^{(1)g} \in \mathbb{R}^{X^g \times J_1}$, $\boldsymbol{W}^{(L+1)g} \in \mathbb{R}^{J_L \times Y^g}$, $\boldsymbol{b}^g \in \mathbb{R}^{X^g}$ and $\boldsymbol{d}^g \in \mathbb{R}^{Y^g}$ are the weight matrices and bias parameters corresponding to the Gaussian units and $\boldsymbol{\sigma}^{(x)g} \in \mathbb{R}^{X^g}$ and $\boldsymbol{\sigma}^{(y)g} \in \mathbb{R}^{Y^g}$ are the deviation parameters of the visible Gaussian units $\boldsymbol{x}^g$ and $\boldsymbol{y}^g$, respectively. Each is the parameter to optimize in the training stage. Note that the binary operator $\circ$ and each division in the energy function denote element-wise product and division.

Under the definition of the energy function of a GCDRM, the conditional probabilities for each visible unit given the adjacent hidden units are computed as:

$$p(x_i^c = 1|\boldsymbol{h}^{(1)}) = \frac{\exp(b_i^c + \boldsymbol{W}_{i:}^{(1)c}\boldsymbol{h}^{(1)})}{\sum_{i'}\exp(b_{i'}^c + \boldsymbol{W}_{i':}^{(1)c}\boldsymbol{h}^{(1)})} \tag{7}$$

$$p(x_i^g = x|\boldsymbol{h}^{(1)}) = \mathcal{N}\Big(x|b_i^g + \boldsymbol{W}_{i:}^{(1)g}\boldsymbol{h}^{(1)}, \sigma_i^{(x)2}\Big) \tag{8}$$

$$p(y_k^c = 1|\boldsymbol{h}^{(L)}) = \frac{\exp(d_k^c + \boldsymbol{W}_{:k}^{(L+1)cT}\boldsymbol{h}^{(L)})}{\sum_{k'}\exp(d_{d'}^c + \boldsymbol{W}_{:k'}^{(L+1)cT}\boldsymbol{h}^{(L)})} \tag{9}$$

$$p(y_k^g = y|\boldsymbol{h}^{(L)}) = \mathcal{N}\Big(y|d_k^g + \boldsymbol{W}_{:k}^{(L+1)gT}\boldsymbol{h}^{(L)}, \sigma_i^{(y)2}\Big), \tag{10}$$

where $\mathcal{N}(\cdot|\mu,\sigma^2)$ denotes the Gaussian probability density function with mean $\mu$ and variance $\sigma^2$. The conditional probabilities for hidden units $\boldsymbol{h}^{(1)}$ and $\boldsymbol{h}^{(L)}$ given adjacent hidden units and visible units are computed as:

$$p(\boldsymbol{h}_j^{(1)} = 1|\boldsymbol{x}, \boldsymbol{h}^{(2)}) = \sigma(c_j^{(1)} + \boldsymbol{W}_{:j}^{(1)T}\frac{\boldsymbol{x}}{\boldsymbol{\sigma}^{(x)} \circ \boldsymbol{\sigma}^{(x)}} + \boldsymbol{W}_{j:}^{(2)}\boldsymbol{h}^{(2)}) \tag{11}$$

$$p(\boldsymbol{h}_j^{(L)} = 1|\boldsymbol{y}, \boldsymbol{h}^{(L-1)}) = \sigma(c_j^{(L)} + \boldsymbol{W}_{:j}^{(L)T}\boldsymbol{h}^{(L-1)} + \boldsymbol{W}_{j:}^{(L+1)}\frac{\boldsymbol{y}}{\boldsymbol{\sigma}^{(y)} \circ \boldsymbol{\sigma}^{(y)}}). \tag{12}$$

Note that for notational convenience, we set the weights $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(L+1)}$ as $\boldsymbol{W}^{(1)} \equiv [\boldsymbol{W}^{(1)gT}\ \boldsymbol{W}^{(1)cT}]^T$ and $\boldsymbol{W}^{(L+1)} \equiv [\boldsymbol{W}^{(L+1)gT}\ \boldsymbol{W}^{(L+1)cT}]^T$, respectively. Furthermore, we set the deviations $\boldsymbol{\sigma}^{(x)}$ and $\boldsymbol{\sigma}^{(y)}$ as:

$$\sigma_i^{(x)} = \begin{cases} \sigma_i^{(x)g} & (1 \le i \le X^g) \\ 1 & (X^g < i \le I) \end{cases} \tag{13}$$

$$\sigma_k^{(y)} = \begin{cases} \sigma_k^{(y)g} & (1 \le k \le Y^g) \\ 1 & (Y^g < k \le K) \end{cases}, \tag{14}$$

in Eqs. (11) and (12).

As with a BBDRM, the conditional probabilities for hidden units at the 2nd, ..., $(L-1)$th hidden layers are defined as Eq. (4). In the same fashion as a BBDRM, the parameters $\boldsymbol{\theta} = \{\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{W}^{(1)}, \boldsymbol{W}^{(l)}, \boldsymbol{W}^{(L+1)}, \boldsymbol{\sigma}^{(x)g}, \boldsymbol{\sigma}^{(y)g}\}$, where $\boldsymbol{b} \equiv [\boldsymbol{b}^{gT}\ \boldsymbol{b}^{cT}]^T$ and $\boldsymbol{d} \equiv [\boldsymbol{d}^{gT}\ \boldsymbol{d}^{cT}]^T$, are estimated to maximize the joint log-likelihood $\mathcal{L}$ in the training stage of a GCDRM. The gradients for each parameter are calculated as:

$$\frac{\partial \mathcal{L}}{\partial b_i} = \frac{1}{\sigma_i^{(x)2}}\Big(\langle x_i\rangle_d - \langle x_i\rangle_m\Big) \tag{15}$$

$$\frac{\partial \mathcal{L}}{\partial c_j^{(l)}} = \langle h_j^{(l)}\rangle_d - \langle h_j^{(l)}\rangle_m \tag{16}$$

$$\frac{\partial \mathcal{L}}{\partial d_k} = \frac{1}{\sigma_k^{(y)2}}\Big(\langle y_k\rangle_d - \langle y_k\rangle_m\Big) \tag{17}$$

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{(l)}} = \begin{cases} \frac{1}{\sigma_i^{(x)2}}\Big(\langle x_i h_j^{(1)}\rangle_d - \langle x_i h_j^{(1)}\rangle_m\Big) & (l=1) \\ \langle h_i^{(l-1)} h_j^{(l)}\rangle_d - \langle h_i^{(l-1)} h_j^{(l)}\rangle_m & (l = 2, ..., L) \\ \frac{1}{\sigma_j^{(y)2}}\Big(\langle h_i^{(L)} y_j\rangle_d - \langle h_i^{(L)} y_j\rangle_m\Big) & (l = L+1) \end{cases}. \tag{18}$$

In the training stage of a GCDRM, each parameter is updated iteratively using from Eqs. (15) to (18). The expectations over the model distribution $\langle \boldsymbol{x}_i\rangle_m$ is approximated by iterative inference in the same fashion as a BBDRM. As a result, those are computed by the cyclic propagation (Fig. 1 (b): dotted circles indicate $\langle \boldsymbol{x}_i\rangle_m$). $\langle \boldsymbol{y}_k\rangle_m$ is computed similarly. Additionally, we learn log-variances $z_i^{(x)} = \log \sigma_i^{(x)2}$ and $z_k^{(y)} = \log \sigma_k^{(y)2}$ to keep the variances positive following the training of an IG-BRBM.

# 4. Experiments

## 4.1. Experimental conditions

We evaluated our method on NIT ATR503 M001 dataset [1] from a single Japanese male speaker. The dataset contains about 45 minutes of speech data and its linguistic labels. It consists of 10 small sets, and each set includes 50 sentences. We used 53 sentences as the test data for each experiment. The raw audio was transformed into 35 dimensional mel-cepstral coefficients with deltas and delta-deltas [16], which result in 105 dimensional features as acoustic features. We used 43 phoneme labels for 3 state phones and 47 linguistic contexts (*e.g.* accent types of the current word, position of the current syllable in a phrase) and then the dimension of linguistic features is 176. Each element in acoustic features and linguistic contexts in linguistic features are normalized to have zero-mean and unit-variance over the training data.

Before we compared our method with conventional methods, we investigated the performances of our model when

[1] http://hts.sp.nitech.ac.jp/

Table 1: *Comparison of MCD [dB] between the generated speech and the target speech obtained by each method.*

| method \# of train data | 50 | 100 | 150 | 200 | 450 |
|---|---|---|---|---|---|
| DNN | 3.90 | 3.77 | 3.61 | 3.50 | 3.25 |
| DBN | 3.77 | 3.60 | 3.56 | 3.46 | 3.25 |
| GCDRM | **3.73** | **3.51** | **3.48** | **3.37** | **3.23** |

Table 2: *Comparison of accuracy rate [%] of the current phoneme obtained by each method.*

| method \# of train data | 50 | 100 | 150 | 200 | 450 |
|---|---|---|---|---|---|
| DNN | 62.1 | 67.3 | 70.2 | 71.8 | 76.6 |
| DBN | 65.4 | 68.2 | 70.6 | 71.9 | 77.0 |
| GCDRM | **66.5** | **68.4** | **70.6** | **72.0** | **78.0** |

changing the number of hidden layers from 3 to 5 with 200 or 400 hidden units for speech synthesis. To determine our best architecture, we trained our model using 100 sentences. In the training of a GCDRM, acoustic and linguistic features were assigned to the first and second visible variables $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. Since acoustic features consist of only real-valued data, categorical variables $\boldsymbol{y}^c$ are disregarded. After that, the parameters of GCDRM were fine-tuned using back propagation. In the same fashion as the DNN-based approach, by setting the predicted output features from our method as mean vectors and pre-computed variances from training data as covariance matrices, the speech parameter generation algorithm [17] generates speech parameters. And then, to evaluate generated parameters objectively, mel-cepstral distortion (MCD) [18] was used. Figure. 2 shows that our best model consists of 4 hidden layers which have 400 units for each.

### 4.2. Objective evaluation

First, we compared our method objectively with two conventional methods: a DNN and a DBN when changing the number of the training data (50, 100, 150, 200 and 450 sentences). Each method consists of 4 hidden layers (each has 400 units). The weights of the DNN were initialized randomly, and those of the DBN and our method were fine-tuned using back propagation as a DNN after each training. We used the same conditions for each method: the number of the training sentences, a learning rate of 0.0001, a mini batch-size of 200, and the total number of 120 for epochs. As shown in Table 1, our method "GCDRM" performed best of all in each case. In particular, when the number of training data is less than or equal to 200, our method outperformed the other methods. It is assumed that this is due to the fact that our method models a dependency from a text to a speech in the stochastic training. However, when the number of the training data is 450, performances of each method with no noticeable difference.

As we showed that the performances were improved for speech synthesis using our method, our method will be possible to generate phoneme labels given speech, since a GCDRM models the joint distribution of text and speech. We also conducted a speaker-dependent phoneme recognition experiment, in order to confirm whether a cyclic training of a GCDRM is possible to capture the bidirectional relationships between linguistic features and acoustic features. In this experiment, we evaluated a frame-level phoneme accuracy rate for the current phoneme from estimated linguistic features given acoustic features. The initial weights and biases of a GCDRM are the same as that in the synthesis experiment, and then we fine-tuned parameters as a DNN. Both a DNN and a DBN are trained from scratch in the same conditions of the synthesis experiment. The results of the recognition experiments are shown in Table 2,

Table 3: *Comparisons of MCD [dB] and phoneme accuracy rate [%] obtained by each method trained for 200 sentences.*

| | MCD [dB] | Accuracy rate [%] |
|---|---|---|
| DNN | 3.50 | 71.75 |
| DBN | 3.46 | 71.90 |
| GCDRM-DNN | **3.37** | **72.03** |
| GCDRM | 4.83 | 32.00 |
| GCDRM (ideal) | 2.73 | 90.45 |

Table 4: *Subjective preference scores [%] of speech samples obtained by each method.*

| DNN | DBN | GCDRM | 95% confidence intervals |
|---|---|---|---|
| 43.7 | - | **56.3** | $\pm$ 5.9 |
| - | 49.3 | **50.7** | $\pm$ 10.0 |

which indicate that our method performed best of all in each case. Experimental results show that the accuracy rate of speech recognition is improved by using the same weights and biases despite optimizing the parameters of GCDRM for speech synthesis.

Finally in objective evaluation, we investigated the performances of bidirectional conversion using our method. We compared MCDs and phoneme accuracy rates obtained by DNN, DBN, GCDRM-DNN, GCDRM and GCDRM (ideal) trained for 200 sentences. Note that GCDRM-DNN, GCDRM and GCDRM (ideal) indicate the fine-tuned GCDRM, GCDRM without fine-tuning and GCDRM whose hidden variables are estimated using acoustic and linguistic features extracted from ground truth data, respectively. As shown in Table 3, the performance of our GCDRM-based synthesis can be seen reasonable even without fine-tuning scheme. Therefore, our method can potentially convert linguistic and acoustic features bidirectionally.

### 4.3. Subjective evaluation

Second, we conducted a listening XAB test to compare the performance of our method subjectively. The number of the training data is 200, and 10 subjects each evaluated 20 test sentences that are randomly chosen from the test data. Each method has 4 hidden layers (each has 400 units). In this experiment, we used natural fundamental F0 and phoneme durations, and then only mel-cepstral coefficients were generated from each method. Table 4 shows the comparison results. It can be seen from the table that our method was preferred significantly to the DNN. On the other hand, there was no significant difference between the performances of the DBN and our method.

## 5. Conclusion

In this paper, we proposed a pre-training method for DNN-based speech synthesis systems for small data sets using a Gaussian-Categorical DRM (GCDRM). In the synthesis experiment, our method has obtained improvements of performance, especially when the amount of training data is limited. Moreover, even when each method was trained for all training data, our method performed the best. Additionally, in the speaker-dependent recognition experiment, experimental results showed that the cyclic training has the potential for bidirectional conversion between text and speech. In the future, we will investigate its potential without the fine-tuning scheme.

## 6. Acknowledgements

This work was partially supported by JST ACT-I and by the Telecommunications Advancement Foundation Grants.

# 7. References


[1] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden Markov models," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

[2] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1996, pp. 373–376.

[3] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum pitch and duration in HMM-based speech synthesis," in *Proceedings of Eurospeech*, 1999, pp. 2347–2350.

[4] M. Tamura, T. Masuko, K.Tokuda, and T. Kobayashi, "Speaker adaptation for HMM-based speech synthesis system using mllr," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2001, pp. 805–808.

[5] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[6] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7962–7966.

[7] A. den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *arXiv:1609.03499*, 2016.

[8] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," in *arXiv:1612.07837*, 2016.

[9] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," in *Proceedings of ICLR2017 workshop submission*, 2017.

[10] T. Nakashika, "Deep relational model: A joint probabilistic model with a hierarchical structure for bidirectional estimation of image and labels," *IEICE Transactions on Information and Systems*, vol. E101-D, no. 2, pp. 428–436, 2018.

[11] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[12] R. Salakhutdinov and G. E. Hinton, "Deep Boltzmann machines," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 448–455.

[13] G. E. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[14] K. Cho, A. Ilin, and T. Raiko, "Improved learning of Gaussian-Bernoulli restricted Boltzmann machines," in *Proceedings of the International Conference on Artificial Neural Networks*, 2011, pp. 10–17.

[15] S. Kang, X. Qian, and H. Meng, "Multi-distribution deep belief network for speech synthesis," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8012–8016.

[16] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "Speech synthesis from HMMs using dynamic features," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1996, pp. 389–392.

[17] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2009, pp. 1315–1318.

[18] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, "Voice conversion using artificial neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 3893–3896.