MULTI-CLASS LEARNING ALGORITHM FOR DEEP NEURAL NETWORK-BASED STATISTICAL PARAMETRIC SPEECH SYNTHESIS

Eunwoo Song and Hong-Goo Kang

Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea

ABSTRACT

This paper proposes a multi-class learning (MCL) algorithm for a deep neural network (DNN)-based statistical parametric speech synthesis (SPSS) system. Although the DNN-based SPSS system improves the modeling accuracy of statistical parameters, its synthesized speech is often muffled because the training process only considers the global characteristics of the entire set of training data, but does not explicitly consider any local variations. We introduce a DNN-based context clustering algorithm that implicitly divides the training data into several classes, and train them via a shared hidden layer-based MCL algorithm. Since the proposed MCL method efficiently models both the universal and class-dependent characteristics of various phonetic information, it not only avoids the model over-fitting problem but also reduces the over-smoothing effect. Objective and subjective test results also verify that the proposed algorithm performs much better than the conventional method.

Index Terms— Statistical parametric speech synthesis, deep neural network, context clustering, shared hidden layer

1. INTRODUCTION

Statistical parametric speech synthesis (SPSS) systems with context-dependent hidden Markov models (HMMs) have been researched in recent decades [1]. However, such synthesized speech remains unnatural, mainly due to the problems of vocoding and statistical modeling [2]. Zen et al. proposed a deep neural network (DNN)-based SPSS system to improve modeling accuracy; specifically, they attempted to overcome an over-fitting problem of a decision tree-based state clustering paradigm [3]. In DNN-based SPSS systems, a centralized network enables compact modeling of the complex dependencies between input contexts and output acoustic features. Various analyses have also confirmed that DNN-based SPSS systems perform significantly better than HMM-based ones [3–6].

A limitation of DNN-based acoustic modeling is that it is not suited to representing temporal variations in speech because it adopts a *frame-by-frame* modeling method. Typically, the problem can be relieved by utilizing a speech parameter generation (SPG) algorithm to produce a smooth trajectory of acoustic features [7]. However, this often results in over-smoothed outputs because the DNN only estimates the mean values of acoustic features. Note that in the HMM-based SPSS systems, the input space is first divided into several classes, and then the mean and variance of each class are modeled by a Gaussian mixture model (GMM). On the other hand, the DNN output cannot represent the local variance (LV) of acoustic features because the single network is trained from the entire data set without considering localized signal characteristics. In other words, since the training process with the single network generates statistically averaged outputs, it cannot clearly identify the characteristics of various phonetic information. Therefore, the synthesized speech often becomes muffled due to over-smoothing effect.

The problem can be reduced by introducing a number of networks, such that each network individually models the unique characteristics of each phoneme class. However, this approach is structurally inefficient, and it is difficult to avoid the over-fitting problem that typically occurs when the training database is comparatively small. Consequently, it is very important to develop an alternative modeling algorithm that balances the trade-off between the problems of over-fitting and over-smoothing.

This paper proposes a DNN-based multi-class learning (MCL) algorithm that consists of two networks: a context clustering network and an MCL network. The former is used to segment the training data into several classes, and each segmented class is trained by the latter network. In the context clustering network, one of the hidden layers is replaced by a bottleneck layer, the output of which is then used as a classifier. Since the activation of bottleneck unit compactly represents the relationship between linguistic features and corresponding acoustic features [8], the training data can be effectively clustered into several classes with common statistical characteristics. Furthermore, the LVs obtained from each class are used for the SPG algorithm, which is highly efficient at reducing the over-smoothing effect.

In the proposed MCL network, clustered input and output features are trained using a shared hidden layer (SHL) structure [9, 10]. During the training process, the hidden layers are trained to model the universal attributes among the different classes, whereas the regression layers are trained to represent class-dependent characteristics. Since all the information for each class is shared with hidden layers, the over-fitting problem can also be minimized. Consequently, by introducing the DNN-based context clustering and SHL-based MCL algorithms, the proposed system successfully reduces both over-smoothing and over-fitting problems. Experimental results also verify that the synthesized quality of proposed system is superior to the conventional approach.

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

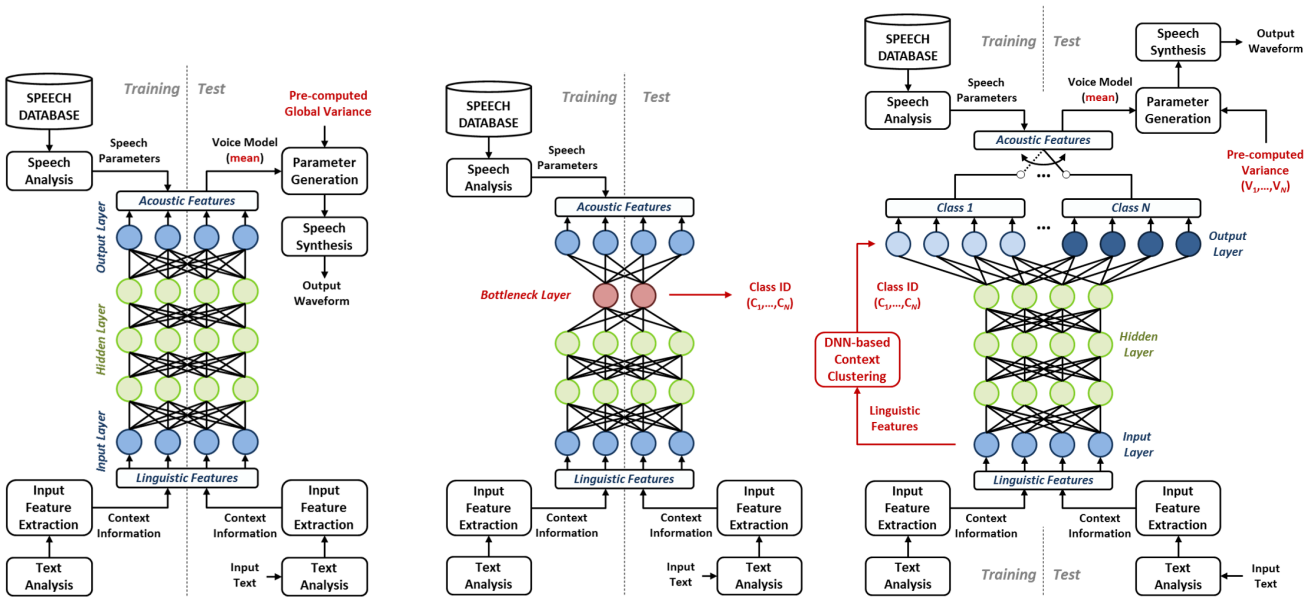

Fig. 1. Framework of the DNN-based SPSS system: (a) conventional algorithm, (b) proposed DNN-based context clustering algorithm, (c) proposed SHL-based MCL algorithm.

2. DNN-BASED SPSS SYSTEM

This section provides an overview of a conventional DNN-based SPSS system and its limitations. The schematic in Fig. 1. (a) shows a typical DNN-based SPSS system. A centralized DNN constructs a non-linear mapping function between linguistic features and corresponding acoustic features. In the training step, the contextual information is first analyzed, and then the sequences of input features are prepared. The output features are composed of acoustic features with their temporal dynamics [11]. The DNN weightings are then trained in order to minimize the mean square error between target and estimated outputs with regard to given inputs.

In the test step, the contextual information of the given text is first converted to the input features, which are then utilized by the trained DNN to estimate the output features. Since the DNN cannot generate smoothed output in the temporal domain, the SPG algorithm should be applied to the output features in order to generate continuous trajectories of speech parameters. Note that the estimated output and the pre-computed global variance (GV) are regarded as a mean and a variance vector of the SPG algorithm, respectively [3]. Finally, a speech synthesis module synthesizes a speech waveform with the generated parameters.

The quality of synthesized speech is quite good compared to the conventional HMM-based approach; however, it still sounds muffled, mainly due to the following two factors. Firstly, a single network cannot capture the diverse characteristics of phonetic information, since the DNN weightings are obtained from the entire training data set without considering its local characteristics. As a result, parameters that have phonetically different natures are statistically averaged during the model training process. Secondly, its structural limitations preclude estimating the variance of parameters which is required for the SPG algorithm. Since a single GV pre-computed from the entire training data is used, this approach is not appropriate to represent the dynamically evolving characteristics of speech parameters. Therefore, an alternative form of training algorithm, that overcomes the structural limitation of DNN-based acoustic modeling, should be developed to generate parameters for synthesis.

3. MULTI-CLASS LEARNING ALGORITHM FOR A DNN-BASED SPSS SYSTEM

This section describes an MCL algorithm for the DNN-based SPSS system. The proposed system consists of the following two networks. The first network (i.e., a context clustering network) clusters linguistic input features and corresponding acoustic output features into several classes. Then, each class of the input-output pairs is trained with the second network (i.e., an MCL network).

3.1. DNN-based context clustering

Before training the model, the input linguistic and output acoustic features should be clustered into several classes in order to avoid over-smoothing the speech parameters. We propose a DNN-based context clustering algorithm where the bottleneck layer, which consists of a relatively small number of units [12], is employed as a classifier. Fig. 1. (b) depicts the framework for a DNN-based context clustering algorithm. The training procedure is similar to that described in Section 2, in which the mapping function between the lin-

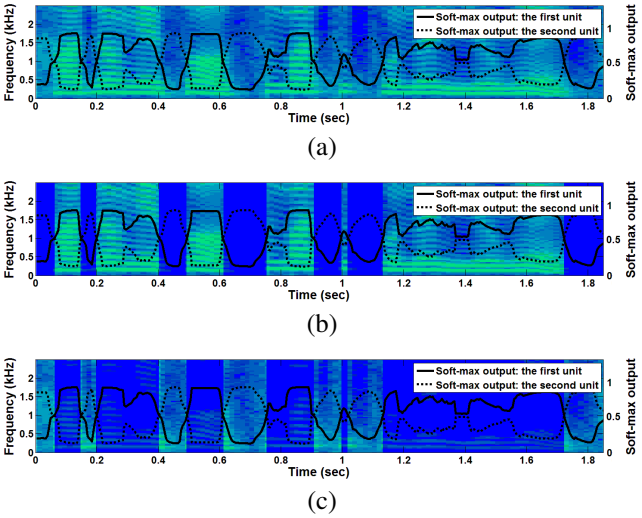

Fig. 2. Spectrogram of synthesized speech from automatically clustered input class and corresponding acoustic parameters (the additional bold and dashed line represent the soft-max outputs of the first and the second unit, respectively): (a) entire signal, (b) voice-like signal, (c) unvoice-like signal.

guistic and acoustic features is modeled by the central DNN structure. However, one of the hidden layers is replaced with the bottleneck layer (the third layer in Fig. 1. (b)), which functions as a classifier during the test step.

The test procedure is also slightly different from that described in Section 2. By setting the activation of the bottleneck unit as a soft-max function, the network estimates the class type of the given context instead of the output acoustic features. Since the bottleneck layer has been trained to represent the statistical characteristics of both linguistic and acoustic features, it can efficiently predict the most probable class for the given context.

Fig. 2 shows an example of output clusters when the bottleneck layer has two units (two classes). It represents the speech signal that is synthesized by each clustered input context and corresponding acoustic parameters. Although the classifier is trained by unsupervised learning procedure, the results show that the clustering network automatically divides input context into two classes, and each class well represents different types of voicing characteristics.

3.2. Shared hidden layer-based multi-class learning

To model the mapping function between linguistic and acoustic features, a number of DNNs should be trained in each class. However, the use of a large number of DNNs is not only inefficient but is also prone to over-fitting to the comparatively small database. In this paper, we propose an efficient MCL approach to train each class by using the SHL training paradigm; the hidden layers are shared to model the universal characteristics of all the classes, while the regression layers are class-dependently trained. The training follows a multi-task learning procedure, in which multiple related tasks are trained simultaneously and benefit from each other [13].

Fig. 1 (c) depicts the framework for an SHL-based MCL training process. In the training step, both input and output features are first clustered into a fixed number of classes, and then each class is modeled with the SHL training procedure. Although there are some differences in the training process due to architectural changes, it is still similar to the conventional back-propagation (BP) algorithm [14]. Note that a key of the SHL training procedure is that all the classes are trained simultaneously [9, 10]. To successfully train the network for all the classes using a mini-batch stochastic gradient descent procedure, the sequence of input-output pairs needs to be randomized across all the classes before they are input to the training process. Then, the hidden layers and each class-specific regression layer are updated through the BP algorithm while other regression layers are kept intact. As a result, both class-independent and class-dependent characteristics are efficiently modeled by the shared hidden layers and the regression layers, respectively.

In the test step, the DNN-based context clustering algorithm first determines the class of input linguistic features, and then the corresponding regression layer estimates the output acoustic features. By setting the estimated output as a mean vector and pre-computed class-dependent LVs of output features, the continuous trajectories of speech parameters are generated by the SPG algorithm. Finally, the speech synthesis module synthesizes speech waveforms using the generated parameters.

4. EXPERIMENTS

4.1. Experimental setup

Phonetically and prosodically balanced speech data recorded by a Korean male speaker were used for the experiments. The speech signals were sampled at 16 kHz, and each sample was quantized by 16 bits. In total, 2,700 utterances (around 3.5 hours) were used for training, 100 utterances were used for validation, and an additional 100 utterances not included in the training and validation steps were used for evaluation. A grapheme-to-phoneme (G2P) converter was developed according to the rules for standard Korean pronunciation grammar and the context information-labeling program.

In the analysis step, the frame length was set to 20 ms, and the spectral and excitation parameters were extracted at every 5 ms based on an improved time-frequency trajectory excitation (ITFTE) vocoder [15]. The 24-dimensional line spectral frequencies (LSFs) were extracted for the spectral parameter, whereas 18-dimensional slowly evolving waveform (SEW) and 4-dimensional rapidly evolving waveform (REW) coefficients were extracted for the excitation parameters; Logarithmic fundamental frequency (log-F0) and energy were also extracted for training them with the DNNs. In the synthesis step, all parameters were generated by the SPG algorithm. The excitation signal was reconstructed by the generated SEW, REW coefficients, and its pitch period. Finally, the single pitch-based speech signal was synthesized from the generated excitation signal and LSFs.

Table 1. Objective test results for baseline DNN and proposed MCL-DNN depending on the number of classes (for the voiced components).

	Number of classes	LSD (dB)	F0 RMSE (Hz)	SEW NRMSE	REW NRMSE
DNN	1	3.449	16.022	0.217	0.258
MCL -DNN	8	3.343	15.710	0.214	0.251
	16	3.337	15.429	0.214	0.257
	32	3.362	15.322	0.213	0.260
	64	3.340	14.898	0.216	0.257

Table 2. Objective test results for the baseline DNN and proposed MCL-DNN depending on the number of classes (for unvoiced and transition components).

	Number of classes	LSD (dB)	SEW NRMSE	REW NRMSE
DNN	1	3.257	0.340	0.336
MCL -DNN	8	3.142	0.336	0.114
	16	3.114	0.337	0.114
	32	3.147	0.333	0.115
	64	3.099	0.337	0.105

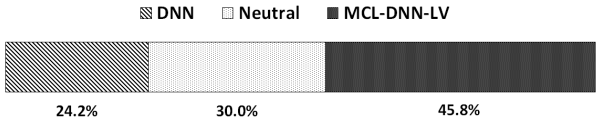

Fig. 3. Results of preference tests (%) comparing the proposed and baseline systems.

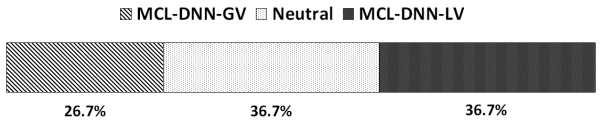

Fig. 4. Results of preference tests (%): proposed system using local variance (LV) compared with the baseline system using global variance (GV).

In the baseline and the proposed MCL-DNN, the input feature vector included 210-dimensional contextual information, consisting of 203 binary and 7 numerical features. The corresponding output feature vector contained 144-dimensional acoustic features that consisted of the ITFTE parameters with their temporal dynamics. The hidden layers comprised 6 layers of 1024 units. Before training, input features were normalized to yield zero-mean and unit-variance, whereas the output features were normalized to give minimum and maximum values of 0.01 and 0.99, respectively. The sigmoid activation function was used for the hidden and regression layers. For training, the weights were initialized randomly and trained using the BP procedure, based on the mini-batch stochastic gradient descent algorithm.

In the context clustering network, the DNN configuration was the same as the baseline and the proposed MCL-DNN system, except for those of the hidden layers: they had 6 layers, but the second layer was replaced with the bottleneck layer. Note that the position of the bottleneck layer was determined empirically. Each hidden layer had 256 units, whereas the number of units in the bottleneck layer varied (at 8, 16, 32, and 64). The hyperbolic tangent and soft-max activation function were used for the hidden and bottleneck layer, respectively.

4.2. Objective and subjective test results

In the objective test, we compared distortions in acoustic parameters obtained from the original speech with those estimated by DNNs. The metrics for measuring distortion were log-spectral distance (LSD) for LSFs in dB, root mean square error (RMSE) for F0 in Hz, and normalized RMSE (NRMSE) for SEW and REW. The test results for the baseline DNN and the proposed MCL-DNN are shown in Table 1. From the results, it is clear that all the parameters generated by the proposed system showed smaller estimation errors than those generated by the baseline system. Moreover, since the accuracy of these parameters is closely related to the quality of synthesized speech, the perceived quality of synthesized speech from the proposed system is expected to be better than that from the baseline system.

To evaluate the perceptual quality of the proposed system, a preference listening test was performed. In the test, twelve listeners were asked to make a decision. Twenty utterances were randomly selected from the evaluation set, which were then synthesized by the baseline and the proposed systems (64 clusters). The results of the preference test (Fig. 3) show that the proposed system provided much higher perceptual quality ($p < 10^{-6}$) than the conventional system.

4.3. Analysis of over-smoothing effect of speech parameters

To verify how the proposed system alleviated the over-smoothing effect, the metrics used in Section 4.2 were also measured for the *transition* and *unvoiced* segments. From the results shown in Table 2, the estimation errors of the proposed system were much smaller than those of the baseline system. Since the SEW and REW represent time-varying voicing characteristics of the excitation signal [6, 15], it can be implied that the temporal variation of the excitation parameter is well represented by the proposed system. The effect of using LV was also evaluated via an additional listening test in which the quality of synthesized speech from the proposed system was compared as: the proposed MCL-DNN with and without LVs obtained by each class. In the latter case, the GV calculated from the entire training data was used for the SPG algorithm. As shown in Fig. 4, the preference test confirms that introducing local variances also improves the perceptual quality ($p < 10^{-2}$) of synthesized speech.

5. CONCLUSION AND FUTURE WORK

A multi-class learning (MCL) algorithm for a deep neural network (DNN)-based statistical parametric speech synthesis (SPSS) system was proposed. To overcome the over-smoothing effect of conventional SPSS systems, a DNN-based context clustering algorithm was introduced. Training data were first divided into several classes automatically, and were then trained by a shared hidden layer (SHL)-based MCL algorithm. The proposed system significantly reduced both the over-smoothing and over-fitting problems. Objective and subjective tests also confirmed the superiority of the proposed system compared to the conventional approach.

In future work, we plan to further improve the quality of the SPSS system by combining the proposed algorithm with the long short-term memory recurrent neural network (LSTM-RNN)-based training process, which is well known as the state-of-the-art modeling method for SPSS systems. Various training algorithms will be also investigated to refine the prediction procedure of DNN-based context clustering algorithm.

6. REFERENCES

[1] Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proc. EUROSPEECH*, 1999.

[2] Heiga Zen, Keiichi Tokuda, and Alan W Black, "Statistical parametric speech synthesis," *Speech Comm.*, vol. 51, no. 11, pp. 1039–1064, 2009.

[3] Heiga Zen, Andrew Senior, and Mike Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, 2013.

[4] Yao Qian, Yuchen Fan, Wenping Hu, and Frank K Soong, "On the training aspects of deep neural network (DNN) for parametric TTS synthesis," in *Proc. ICASSP*, 2014.

[5] Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, "The effect of neural networks in statistical parametric speech synthesis," in *Proc. ICASSP*, 2015.

[6] Eunwoo Song and Hong-Goo Kang, "Deep neural network-based statistical parametric speech synthesis system using improved time-frequency trajectory excitation model," in *Proc. INTERSPEECH*, 2015.

[7] Keiichi Tokuda, Takashi Masuko, Tetsuya Yamada, Takao Kobayashi, and Satoshi Imai, "An algorithm for speech parameter generation from continuous mixture HMMs with dynamic features," in *Proc. EUROSPEECH*, 1995.

[8] Zhizheng Wu, Cassia Valentini-Botinhao, Oliver Watts, and Simon King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *Proc. ICASSP*, 2015.

[9] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proc. ICASSP*, 2013.

[10] Yuchen Fan, Yao Qian, Frank K Soong, and Lei He, "Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis," in *Proc. ICASSP*, 2015.

[11] Sadaoki Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," *IEEE Trans. Acoust., Speech Signal Process.*, vol. 34, no. 1, pp. 52–59, 1986.

[12] Dong Yu and Michael L Seltzer, "Improved bottleneck features using pretrained deep neural networks.," in *Proc. INTERSPEECH*, 2011.

[13] Rich Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[14] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, "Learning internal representations by error propagation," Tech. Rep., DTIC Document, 1985.

[15] Eunwoo Song, Young Sun Joo, and Hong Goo Kang, "Improved time-frequency trajectory excitation modeling for a statistical parametric speech synthesis system," in *Proc. ICASSP*, 2015.