10th ISCA Speech Synthesis Workshop
20-22 September 2019, Vienna, Austria

Low computational cost speech synthesis based on deep neural networks using hidden semi-Markov model structures

Motoki Shimada, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda

Department of Computer Science and Engineering, Nagoya Institute of Technology, Nagoya, Japan
moto@sp.nitech.ac.jp

Abstract

We propose a method of changing the units of input features from states used conventionally to phonemes and moras to reduce the computational cost of deep neural networks (DNNs) with a hidden semi-Markov model structure for speech synthesis, which can model acoustic features and a temporal structure in a unified framework. Neural networks with very deep and wide structures have recently been applied successfully in the field of speech synthesis. However, such models have very high computational cost, so they are not being applied on platforms with limited resources. To solve this problem, we increased the length of time of DNN input units. We used phoneme or mora units, which are longer than the state units used conventionally. Increasing the length in time of units of input features reduces the number of DNN forward propagations required for speech synthesis, reducing the computational cost. Since a mora in Japanese exhibits isochronism, the duration can be represented more appropriately than the phoneme units expressing consonants and vowels of different lengths with one neural network. Experimental results indicate that compared with speech synthesis based on a DNN with frame inputs, computational cost can be reduced by 97% without degrading the naturalness of the synthesized speech with the proposed method.
Index Terms: Text-To-Speech, statistical model, Deep Neural Network, computation costs reduction of a neural network

1. Introduction

Statistical parametric speech synthesis (SPSS) [1] has recently attracted attention in the field of speech synthesis and is being studied actively. With SPSS, a speech waveform is predicted from text representing the utterance content by using a statistical model called an acoustic model, which is usually a hidden Markov model (HMM) [2]. An HMM is suitable for modeling sequential data such as speech data and is mathematically easy to handle. With a hidden semi-Markov model (HSMM) [3], state durations are represented by explicitly assigning state duration distributions, so both the sequence of acoustic features and its temporal structure can be modeled simultaneously in a unified framework.

Speech synthesis based on deep neural networks (DNNs) has been proposed for synthesizing more natural speech [4]. DNNs can model the one-to-one non-linear relationships between inputs and outputs and have performed well in various fields [5, 6, 7]. They can also accurately represent the complex relationship between text and speech for speech synthesis and have demonstrated ability to produce very natural speech. The performance of DNN-based speech synthesis has also be improved [8, 9]. However, since a DNN models a one-to-one correspondence between input and output features, it is not possible to directly use phonemes as input features and frames as output features. Therefore, an external model must be used to arrange them into sequences of linguistic features and acoustic features with the same time units. Therefore, the temporal structure of speech is not adequately considered in the training of the acoustic model. To solve this problem, end-to-end models, such as Tacotron 2 [10] and char2wav [11], were proposed that can directly model input and output features with different lengths. One problem with end-to-end models is the large computational cost of speech synthesis compared with standard DNN-based models. To solve this problem, we previously propose a neural network model that extends a mixture density network (MDN) and introduces an HSMM structure. We call this model MDN-HSMM [12] and use it as an acoustic model for speech synthesis. MDN-HSMM learns both the acoustic feature sequence and its temporal structure at the same time by computing parameters of the output probability distribution and state duration distribution, which are HSMM parameters, as the outputs of the neural network. MDN-HSMM-based speech synthesis achieves the naturalness of synthesized speech equivalent to conventional DNN-based speech synthesis while reducing the computational cost of speech synthesis.

In this study, we developed a method of changing the units of input features from states used conventionally to phonemes and moras, which are longer in time, for MDN-HSMM. This MDN-HSMM outputs HSMM parameters for the number of states expressing one phoneme or one mora for inputting linguistic features in phoneme or mora units, respectively. This makes it possible to greatly reduce the times of running neural networks during synthesis. We expect to further reduce the computational cost of speech synthesis without degrading the quality of the synthesized speech.

In Section 2, we describe SPSS and the structure, training, and synthesis of the acoustic model for MDN-HSMM-based speech synthesis in Section 3. In Section 4, we discuss the evaluation of MDN-HSMM using the proposed method through objective and subjective experiments and conclude the paper and discuss future work in Section 5.

2. Statistical parametric speech synthesis

SPSS consists of a training phase and synthesizing phase. In the training phase, a sequence of linguistic features, such as phonemes and parts of speech, converted from the text and a sequence of acoustic features extracted from speech data, such as fundamental frequencies and the mel-cepstrum, are used to train a statistical model, which is called the acoustic model. In the synthesis phase, the text to be synthesized is converted to a sequence of linguistic features, and this is passed to the trained acoustic model to generate a sequence of corresponding acoustic features. The speech is then synthesized based on this sequence of acoustic features. Normally, the linguistic features are given in units of phonemes, and the acoustic features are

10.21437/SSW.2019-32

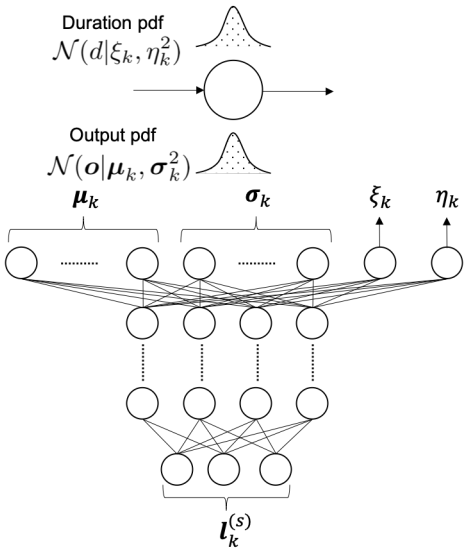

Figure 1: *Structure of MDN-HSMM*

given in units of frames, which are expressed as

$$\boldsymbol{l}^{(p)} = \left(\boldsymbol{l}_1^{(p)}, \boldsymbol{l}_2^{(p)}, \ldots, \boldsymbol{l}_I^{(p)}\right), \tag{1}$$

$$\boldsymbol{o}^{(f)} = \left(\boldsymbol{o}_1^{(f)}, \boldsymbol{o}_2^{(f)}, \ldots, \boldsymbol{o}_T^{(f)}\right). \tag{2}$$

Here, $\boldsymbol{l}^{(p)}$ is the sequence of linguistic features consisting of phoneme units, $\boldsymbol{o}^{(f)}$ is the sequence of acoustic features consisting of frame units, I is the number of phonemes, and T is the number of frames.

There has been extensive research on DNN-based speech synthesis recently. With linguistic features as inputs and acoustic features as outputs, a standard DNN models a one-to-one relationship between input and output features. The difference between input and output length makes it difficult to train DNNs. Therefore, an external model for modeling temporal structures of speech, such as an HSMM, is used to first convert the sequence of linguistic features in units of phonemes, to a sequence of linguistic features in units of frames. DNNs can model the complex non-linear relationship between inputs and outputs accurately, and DNN-based speech synthesis is known to produce more natural synthesized speech than conventional HMM-based speech synthesis.

3. MDN-HSMM-based speech synthesis

3.1. Model structure

With HSMM-based speech synthesis, both acoustic features and duration information can be modeled simultaneously, but with a DNN or MDN acoustic model, training must be done based on duration information estimated using an external model. For this reason, the temporal structure of the speech may not receive sufficient consideration during acoustic model training. We developed MDN-HSMM to solve this problem [12]. It simultaneously models both the acoustic feature sequence and temporal structure of the speech by outputting parameters for a state output distribution and state duration distribution of an HSMM, has been proposed [12]. The structure of MDN-HSMM with an HSMM structure is shown in Figure 1. Since the MDN outputs $\boldsymbol{\lambda}$, the model parameters of the HSMM, a likelihood function of MDN-HSMM is represented by the following equation.

$$p(\boldsymbol{o}^{(f)}|\boldsymbol{l}^{(s)}, \boldsymbol{\lambda}^{(s)}) = \sum_{\boldsymbol{q}} \left\{ \prod_{t=1}^{T} p(\boldsymbol{o}_t|q_t, \boldsymbol{l}^{(s)}, \boldsymbol{\lambda}^{(s)}) \prod_{k=1}^{K} p(d_k|k, \boldsymbol{l}^{(s)}, \boldsymbol{\lambda}^{(s)}) \right\}. \tag{3}$$

We assume a J-state left-to-right HSMM and omit the state transition probability. The term $\boldsymbol{l}^{(s)}$ represents the linguistic features in units of state, which consist of the linguistic features in units of phonemes $\boldsymbol{l}^{(p)}$ and vectors representing state indices in the phoneme. Also, $\boldsymbol{q}$ represents the state sequence, d_k represents the duration for state k, and K represents the number of HSMM states in the utterance. Then the relationship between $\boldsymbol{q}$ and d_k can be expressed as

$$\begin{aligned} \boldsymbol{q} &= (q_1, q_2, \ldots, q_T) \\ &= (\underbrace{1, \ldots, 1}_{d_1}, \underbrace{2, \ldots, 2}_{d_2}, \ldots\ldots, \underbrace{K, \ldots, K}_{d_K}). \end{aligned} \tag{4}$$

The state output distribution and state duration distribution for each is expressed as a normal distribution. If the mean of the state output distribution is $\boldsymbol{\mu}_k$, variance is $\boldsymbol{\sigma}_k^2$, mean of the state duration distribution is ξ_k, and variance is η_k^2, then the state output distribution of $p(\boldsymbol{o}_t|q_t, \boldsymbol{l}^{(s)}, \boldsymbol{\lambda}^{(s)})$ and state duration distributions of $p(d_k|k, \boldsymbol{l}^{(s)}, \boldsymbol{\lambda}^{(s)})$ can be expressed as

$$p(\boldsymbol{o}_t|q_t, \boldsymbol{l}^{(s)}, \boldsymbol{\lambda}^{(s)}) = \mathcal{N}(\boldsymbol{o}_t|\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2), \tag{5}$$

$$p(d_k|k, \boldsymbol{l}^{(s)}, \boldsymbol{\lambda}^{(s)}) = \mathcal{N}(d_k|\xi_k, \eta_k^2). \tag{6}$$

Given the above, $\boldsymbol{\lambda}$ can be expressed as

$$\boldsymbol{\lambda}^{(s)} = \left\{\boldsymbol{\lambda}_1^{(s)}, \boldsymbol{\lambda}_2^{(s)}, \ldots, \boldsymbol{\lambda}_K^{(s)}\right\}, \tag{7}$$

$$\boldsymbol{\lambda}_k^{(s)} = \left\{(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k), (\xi_k, \eta_k)\right\}. \tag{8}$$

3.2. Training phase

The training step of MDN-HSMM in state units is shown in Figure 2. The parameters of MDN-HSMM are trained by maximizing the likelihood of $p(\boldsymbol{o}^{(f)}|\boldsymbol{l}^{(s)}, \boldsymbol{\lambda}^{(s)})$ for the training data. Since it is difficult to maximize this likelihood directly, an expectation-maximization (EM) algorithm [13] is applied. The expected likelihood in the EM algorithm can be represented by the function $\mathcal{Q}$ below.

$$\mathcal{Q}(\boldsymbol{\lambda}^{(s)}, \hat{\boldsymbol{\lambda}}^{(s)}) = \sum_{\boldsymbol{q}} P(\boldsymbol{q}|\boldsymbol{o}^{(f)}, \hat{\boldsymbol{\lambda}}^{(s)}) \log p(\boldsymbol{o}^{(f)}, \boldsymbol{q}|\boldsymbol{\lambda}^{(s)}) \tag{9}$$

The state posterior probability $p(\boldsymbol{q}|\boldsymbol{o}^{(f)}, \hat{\boldsymbol{\lambda}}^{(s)})$ can be computed efficiently using the generalized forward-backward algorithm [3]. Negative $\mathcal{Q}$ is used as the error function, and parameters are optimized by finding neural-network parameters that minimize this error function. In doing so, parameters can be updated for both the acoustic features and duration distribution, so both the acoustic feature sequence and temporal structure can be modeled at the same time.

3.3. Linguistic feature units

So far in our discussion, it was assumed that the linguistic features input to the neural network would be in units of state, but it is also possible to model output probability distributions and

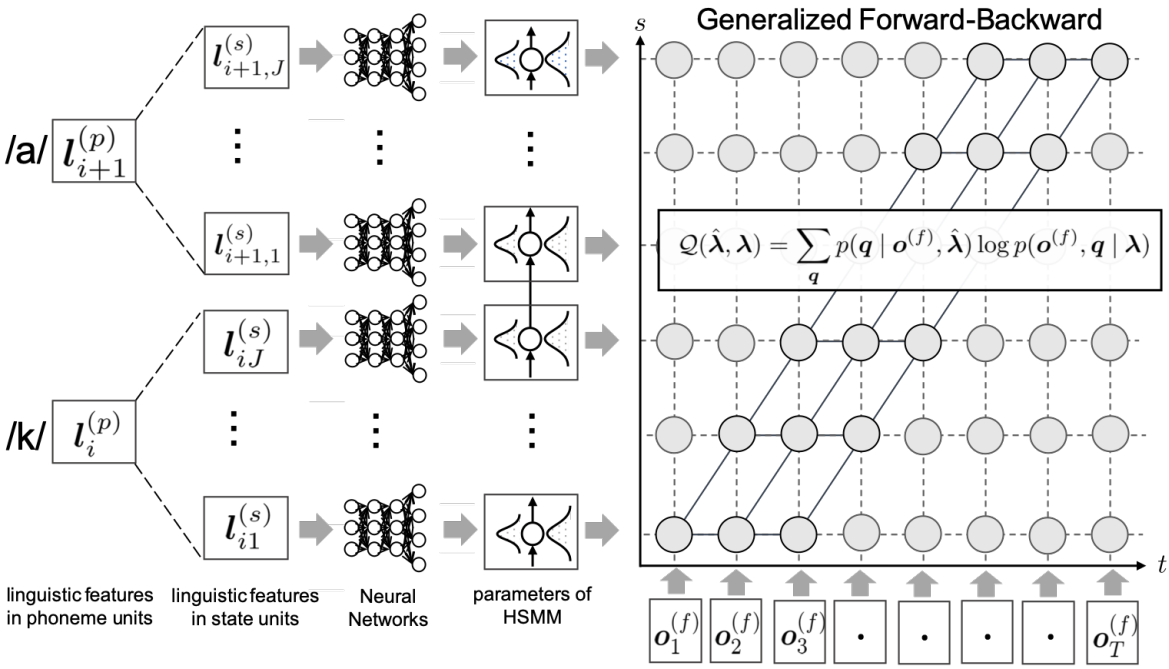

Figure 2: *Training step of MDN-HSMM in state units*

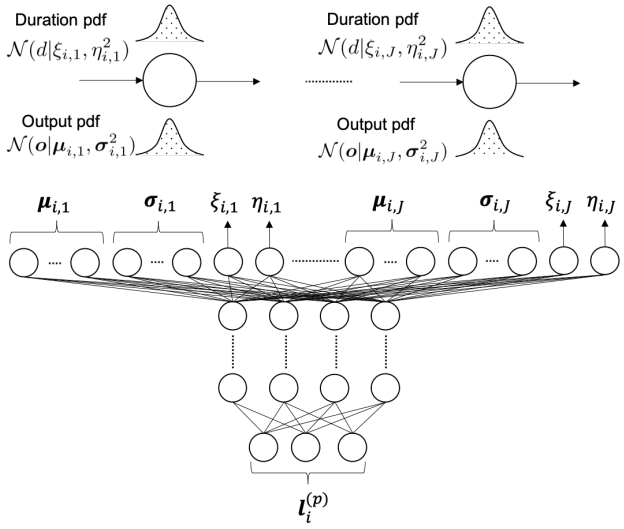

Figure 3: *MDN-HSMM in phoneme units*

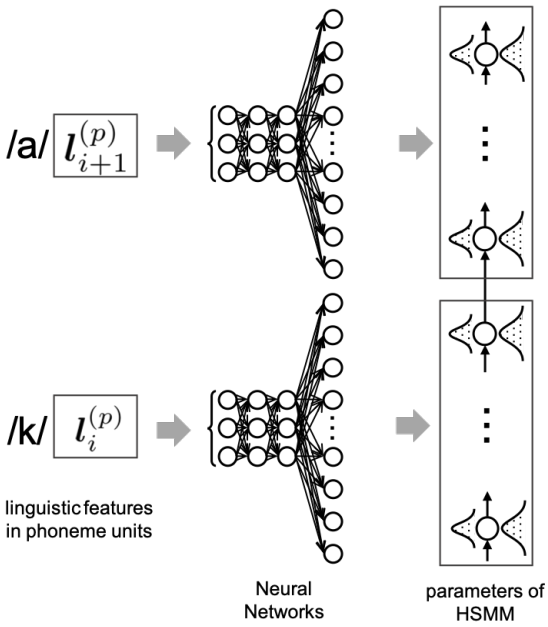

Figure 4: *MDN-HSMM in phoneme units in synthesis phase*

state duration distributions for each phoneme or mora by inputting linguistic features in units of phonemes or moras. Moras are a unit of time, and these units listed in order of increasing length of time are state, phoneme, and mora. By inputting linguistic features in units of phoneme or mora, HSMM parameters will be output in phoneme or mora units, respectively, rather than state. The MDN-HSMM with an HSMM structure in phoneme units is shown in Figure 3. By changing input features from phoneme units to mora units, MDN-HSMN in mora units is implemented. MDN-HSMM in units of phoneme in the synthesis phase is shown in Figure 4, and that in units of mora in this phase is shown in Figure 5.

The relationship among K, I, and number of moras N can be expressed as

$$K > I > N. \tag{10}$$

Thus, introducing phoneme or mora units can make it possible to reduce the running times of neural networks during speech synthesis, which reduces computational cost. For Example, when the consonant-vowel mora "ka" is synthesized, MDN-HSMM in state units runs neural networks $2 \times J$ times. However, since MDN-HSMM in phoneme units only needs to be im-

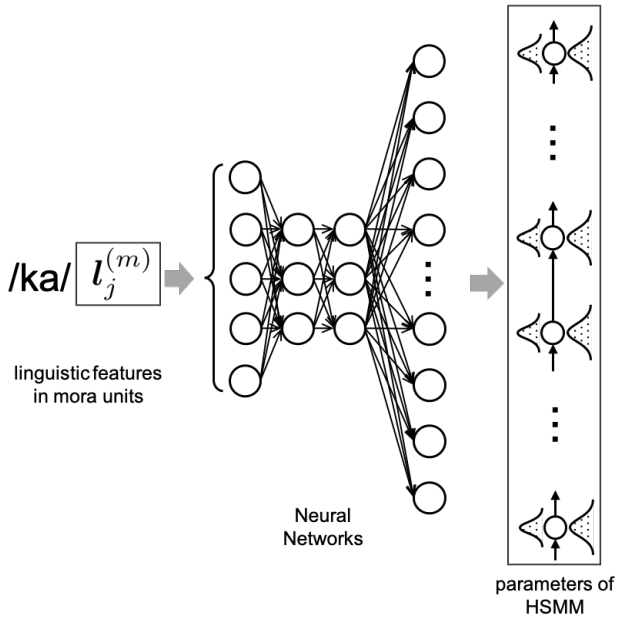

Figure 5: *MDN-HSMM in mora units in synthesis phase*

plemented twice, it can be synthesized by the number of running times of the neural networks of $1/J$ compared to MDN-HSMM in state units. Furthermore, by changing to MDN-HSMM in mora units, it is possible to reduce the times of running the neural networks in the consonant-vowel mora by half.

There is another advantage of changing input features to mora units. It means that a mora exhibits isochronism. In the case of phoneme units, it is necessary to express features that largely differ in time units, such as consonants and vowels, by one a neural network. However, since all moras have almost the same length, it is considered that modeling the duration is easy compared to using phoneme units.

4. Experiments

We conducted objective and subjective experiments to evaluate MDN-HSMM using the proposed method.

4.1. Experimental conditions

We used a phoneme-balanced dataset obtained from Japanese speakers in our laboratory. We used 450 utterances from a male speaker as training data, and 53 utterances not included in the training data as test data. Speech signals were sampled at 48 kHz, and frame shifts were 5 ms. A 154-dimentional vector consisting of a 49-dimensional mel-cepstrum extracted using STRAIGHT [14], the log fundamental frequency, dynamic features, and binary features representing voiced and unvoiced were used as acoustic features.

We compared the following four models in the experiments.

- **DNN**: DNN speech synthesis in frame units (baseline). The state alignment was generated using an HSMM with a five-state, left-to-right, no-skip structure. The input feature was a 411-dimensional vector obtained by adding 3-dimensional duration information to a 408-dimensional frame linguistic feature vector.
- **MDN-HSMM-state**: MDN-HSMM speech synthesis in state units. The input feature was a 413-dimensional vector obtained by adding a 5-dimenaional one-hot vector representing state position to a 408-dimensional frame linguistic feature vector.
- **MDN-HSMM-phoneme**: MDN-HSMM speech synthesis in phoneme units. The input feature was a 408-dimensional phoneme linguistic feature, and the neural network outputs were the output probability distribution and state duration distribution for a five-state HSMM.
- **MDN-HSMM-mora**: MDN-HSMM speech synthesis in mora units. The input feature was a 1514-dimensional mora linguistic feature, and the neural network outputs were the output probability distribution and state duration distribution for an eight-state HSMM.

A single network with three hidden layers of 1024 units each was used for all models. The activation function for the hidden layers was a sigmoid function, and the activation function for the output layer was a linear function. The input features were normalized to the range of (0, 1) and the output features were normalized to have zero-mean unit-variance. According to a previous study [15], **MDN-HSMM-state** can be expressed with a five-state HSMM, and preliminary experiments indicated that the best performance was achieved when implementing **MDN-HSMM-phoneme** with a five-state HSMM and **MDN-HSMM-mora** with an eight-state HSMM. The input features of the mora units are expressed by quin-mora by giving information such as what the mora is and what vowels and consonants are that make up the mora.

The network parameters for all models were initialized randomly and trained using error back-propagation based on stochastic gradient descent. Note that initial training was done in mini-batches for each phoneme with **MDN-HSMM-state** and **MDN-HSMM-phoneme**, and for each mora with **MDN-HSMM-mora**. Subsequent mini-batches were for each utterance.

4.2. Results and discussion

Mel-cepstral distortion (MCD), which represents the mel-cepstral error coefficients between synthesized speech and natural speech, was used as an objective measure to evaluate model performance.

The results of the objective experiment are shown in Figure 6. Table 1 shows the number of running times of the neural networks during synthesizing for **DNN**, **MDN-HSMM-state**, **MDN-HSMM-phoneme**, and **MDN-HSMM-mora**. The values in parenthesis are the number of running times relative to that for **DNN**.

Figure 6 shows that all methods using MDN-HSMM produced smaller values than **DNN**. This confirms the effectiveness of MDN-HSMM, which is able to model both acoustic features and their temporal structure in a unified framework rather than using externally estimated duration information. It

Table 1: *Number of running times of neural networks*

Model	Number of running times
DNN	45541
MDN-HSMM-state	11475(25.2%)
MDN-HSMM-phoneme	2295(5.0%)
MDN-HSMM-mora	1385(3.0%)

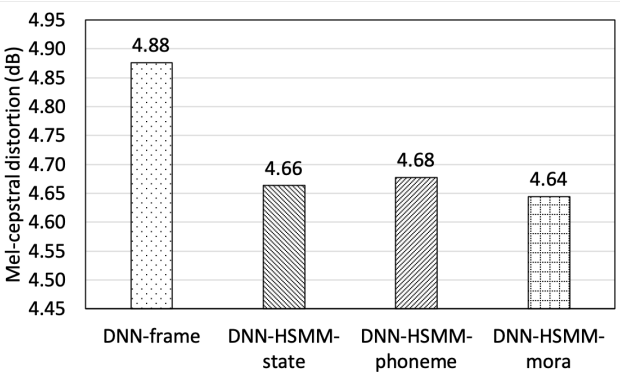

Figure 6: *Mel-cepstral distortion for each model*

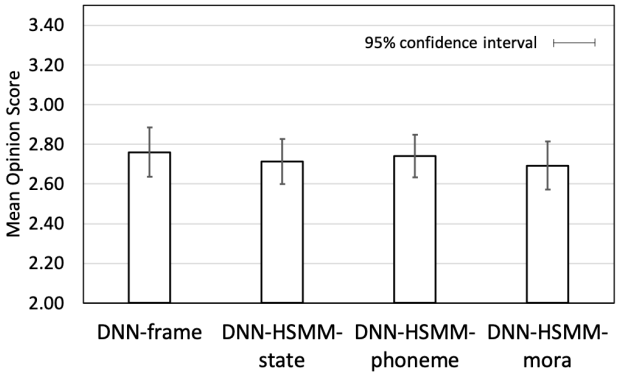

Figure 7: *Mean opinion score*

also shows that the difference in MCD values among **MDN-HSMM-state**, **MDN-HSMM-phoneme**, and **MDN-HSMM-mora** are not large. These results and Table 1 suggest that it is possible to significantly reduce the running times of the neural networks during synthesis to 3.0% of that without significantly degrading the naturalness of the synthesized speech.

We also evaluated the naturalness of the four models with a mean opinion score (MOS) test. Ten participants listened to 15 utterances from the test data, played in random order, and evaluated them for naturalness on a five-point scale. The synthesized speech produced with each model was also played back in random order.

The results of the subjective experiment are shown in Figure 7. Figure 7 shows that all models received similar scores. This indicates that any degradation in the synthesized speech is not noticeable, even when a human actually listens to the speech. The proposed method can reduce computational cost in the synthesis phase without significantly degrading the naturalness of the synthesized speech.

5. Conclusion

We evaluated MDN-HSMM using the proposed a method for reducing the computational cost of speech synthesis by outputting parameters for multiple states of HSMM, which represent a phoneme or mora, from a linguistic feature in units of phonemes or moras. Experimental results indicate that computational cost significantly decreased compared with a conventional DNN-based speech synthesis model , which drives frame-by-frame, without substantially degrading the naturalness of the synthesized speech.

For future work, we plan to further increase the length of linguistic features to units of syllables, investigating the effect of this change on reducing the running times of the neural networks. We also plan to examine reducing the size of the neural network to reduce computational cost.

6. Acknowledgements

The MIC/SCOPE #162106106.
JSPS KAKENHI Grant Number JP18H04128.

7. References

[1] A. W. Black, H. Zen, and K. Tokuda, "Statistical parametric speech synthesis," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4. IEEE, 2007, pp. IV–1229.

[2] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for hmm-based speech synthesis," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, vol. 3. IEEE, 2000, pp. 1315–1318.

[3] H. Zen, K. Tokuda, T. Masuko, T. Kobayasih, and T. Kitamura, "A hidden semi-markov model-based speech synthesis system," *IEICE transactions on information and systems*, vol. 90, no. 5, pp. 825–834, 2007.

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

[5] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. 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.

[6] D. C. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *CoRR*, vol. abs/1202.2745, 2012. [Online]. Available: http://arxiv.org/abs/1202.2745

[7] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 160–167. [Online]. Available: http://doi.acm.org/10.1145/1390156.1390177

[8] Z. Wu and S. King, "Minimum trajectory error training for deep neural networks, combined with stacked bottleneck features," in *INTERSPEECH*, 2015.

[9] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 3844–3848.

[10] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," *CoRR*, vol. abs/1712.05884, 2017. [Online]. Available: http://arxiv.org/abs/1712.05884

[11] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2Wav: End-to-end speech synthesis," in *International Conference on Learning Representations (Workshop Track)*, April 2017.

[12] K. Tokuda, K. Hashimoto, K. Oura, and Y. Nankaku, "Temporal modeling in neural network based statistical parametric speech synthesis," pp. 106–111, 09 2016.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[14] H. Kawahara, I. Masuda-Katsuse, and A. De Cheveigne, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds1," *Speech communication*, vol. 27, no. 3-4, pp. 187–207, 1999.

[15] K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Simultaneous modeling of acoustic feature sequences and its temporal structures for dnn-based speech synthesis," *IEICE Tech. Rep.*, vol. 116, no. 414, pp. 71–76, jan 2017. [Online]. Available: https://ci.nii.ac.jp/naid/40021102099/