ADAPTING AND CONTROLLING DNN-BASED SPEECH SYNTHESIS USING INPUT CODES

Hieu-Thi Luong1⇤, Shinji Takaki2, Gustav Eje Henter2, Junichi Yamagishi2,3

1VNUHCM – University of Science, Ho Chi Minh City, Vietnam
2National Institute of Informatics, Tokyo, Japan
3The University of Edinburgh, Edinburgh, UK

ABSTRACT

Methods for adapting and controlling the characteristics of output speech are important topics in speech synthesis. In this work, we investigated the performance of DNN-based text-to-speech systems that in parallel to conventional text input also take speaker, gender, and age codes as inputs, in order to 1) perform multi-speaker synthesis, 2) perform speaker adaptation using small amounts of target-speaker adaptation data, and 3) modify synthetic speech characteristics based on the input codes. Using a large-scale, studio-quality speech corpus with 135 speakers of both genders and ages between tens and eighties, we performed three experiments: 1) First, we used a subset of speakers to construct a DNN-based, multi-speaker acoustic model with speaker codes. 2) Next, we performed speaker adaptation by estimating code vectors for new speakers via backpropagation from a small amount of adaptation material. 3) Finally, we experimented with manually manipulating input code vectors to alter the gender and/or age characteristics of the synthesised speech. Experimental results show that high-performance multi-speaker models can be constructed using the proposed code vectors with a variety of encoding schemes, and that adaptation and manipulation can be performed effectively using the codes.

Index Terms—Speech synthesis, DNNs, speaker adaptation, speech manipulation, voice morphing

1. INTRODUCTION

In some applications of speech technology, the flexibility and controllability of speech synthesis systems are important factors, in addition to the quality of the synthetic speech. It is well known that speech synthesis based on hidden Markov models (HMMs) is highly flexible and controllable. This is exemplified by 1) speaker adaptation, a technique to estimate a new acoustic model based on small amounts of speech data uttered by a new target speaker or in a new speaking style (e.g., a different emotion) [1, 2]; 2) speaker interpolation, a technique to morph together several speakers or emotions by interpolating mean vectors and variance matrices of Gaussian distributions from representative acoustic models [3]; and 3) multiple regression HMMs (MR-HMMs) [4], a technique that modifies the acoustic characteristics of synthesised speech by augmenting the text input with additional input values (control parameters).

Recently, speech synthesis using deep neural networks (DNNs) has become an active area of research, with applications to acoustic modelling [5, 6], duration modelling [7], feature extraction [8], and text analysis [9] having been investigated by various groups. It has been reported that DNN-based techniques have improved the quality of synthetic speech significantly; cf. [10]. A new DNN baseline system [11] added to the Blizzard Challenge 2016 also turned out to be significantly better than the standard HMM-based baseline (that uses a toolkit called HTS [12]), again confirming the speech quality improvements brought on by deep learning approaches.

In this paper, we consider augmenting the conventional text-based inputs of DNN-based acoustic models with auxiliary input features – collectively referred to as input codes – that include speaker codes as well as encodings of other labelled attributes, such as gender and age. We show that these input codes have capabilities that unify previous approaches to speaker adaptation, speaker interpolation, and multiple regression. More specifically, we first show that we can train a useful multi-speaker model based on input codes, even from speech database with over 100 speakers, each contributing only about 100 utterances. The model is trained exactly like a speaker-dependent acoustic model, but augmenting the linguistic inputs with additional features encoding each speaker’s identity, gender, and age. Next, we show that the new model can perform speaker adaptation from small amounts of speech by estimating speaker codes and/or input codes using backpropagation. Finally, we show that we can achieve speaker, gender, and age morphing by gradually changing the input code from one value to another.

We also investigate several ways of representing speaker, gender, and age features in vector form. We compare three different encodings, namely the standard one-hot vector form, random vectors,
and a data-driven numerical encoding sometimes called “discriminant condition codes” (DCCs) [17]. We further analyse the dimensionality required for the input code representations.

In the remainder of the paper, Section 2 describes training a neural network modelling several training speakers. Sections 3 and 4 explain speaker code estimation for new speakers via backpropagation and the modification of input codes to change the speaker, gender, and age characteristics of output speech. Section 5 evaluates the approaches in a number of experiments, while Section 6 concludes.

2. MULTI-SPEAKER MODELS USING INPUT CODES

We can train a multi-speaker acoustic model like a standard speaker-dependent model by simply pooling the data in a multi-speaker speech database. However, in order to retain speaker voice characteristics and to allow speaker adaptation later on, we augment the standard input features with auxiliary features such as the speakers’ identity, gender, and age. These input codes allow the DNN to discriminate individual speakers, gender, and age bands during training and synthesis; by inputting the average value of the input code, a kind of average voice is produced. This architecture is trivial, and similar ideas have cropped up in prior literature, not only in speech synthesis [13, 18, 19], but also in other fields [17, 20, 21, 22].

We first address the question of how to represent auxiliary features like speaker, gender, and age in vector form. In particular, we compare three different encodings, namely standard one-hot vectors, random vectors, and DCC vectors, for the speaker codes:

One-hot vector speaker codes: The standard one-hot vector speaker code is defined as follows: If there are \( N \) speakers in the training set, the one-hot vector (sometimes called the I-of-\( k \) vector) for the \( i \)-th speaker is \( s_i = (s_{i1}, s_{i2}, ..., s_{iK}) \) where each value \( s_{ik} \) is 0 when \( n \neq i \) and 1 when \( n = i \).

Random-vector speaker codes: Speaker codes based on random vectors are vectors of a predetermined dimension \( K \). The vector for speaker \( i \) is \( s_i = (s_{i1}, s_{i2}, ..., s_{iK}) \) where \( s_{ik} \) are random values, here sampled from a uniform distribution on \([0, 1]\). This scheme provides a simple method to reduce or expand the size of the speaker code.

Discriminant condition codes: DCCs were initially introduced by Xue et al. [17] for speech recognition. Figure 1 provides a simplified overview of the main idea. As seen in the figure, discriminant codes constitute hidden-unit activations obtained by projecting one-hot speaker codes into \( K \) dimensions using a matrix \( W_{DCC} \in \mathbb{R}^{N \times K} \). The DCCs are trained jointly with the weights and biases of a DNN [17, 20]. This code is equivalent to the one proposed by Watts et al. [16], except that the latter work considers synthesis and used DCCs to distinguish sentences instead of speakers.

2.1. Gender and age codes

For the gender and age codes, we consider two different representations: 1) one-dimensional numerical or binary representations, or 2) 1-of-\( k \) categorical representations. Gender information, for instance, may be represented by binary values, e.g., 0 for female and 1 for male. Alternatively, we may use a two-dimensional 1-of-\( k \) categorical representation where, e.g., the first dimension corresponds to male and the second dimension corresponds to female. For the age information, we may use the raw age as a one-dimensional numerical value, or we may divide speakers into several age bands and assign a similar 1-of-\( k \) categorical representation.

3. ADAPTING DNN ACOUSTIC MODELS USING INPUT CODES

To adapt the above multi-speaker models to a new speaker, one wants to find the speaker code that makes generated speech resemble the target speaker the most. Following [23], we use backpropagation (BP) to minimise the mean square prediction error over a small amount of data uttered by the target speaker.\(^2\) This differs from [13, 24], whose adapted speaker codes do not minimise prediction error directly. Note that the BP algorithm only updates the speaker codes, without changing the DNN weights, in contrast to [19], which used fixed codes but added new weights. We use a well-trained multi-speaker acoustic model as described Section 2, initialised from the average speaker code, since average voice models are known to provide good starting points for HMM-based speaker adaptation [1]. Forward propagation and backpropagation are then iterated to estimate the new speaker code until a stopping criterion is satisfied.

4. MANIPULATING SPEECH USING INPUT CODES

Models from Section 2 can also be used to manipulate the acoustic characteristics of synthesised speech, achieving behaviour similar to HMM-based speaker interpolation as in [3]. We can linearly interpolate DCCs or other input codes, gradually changing them from the value for one speaker to that of another. This should perform speaker morphing.\(^3\) Having labelled control parameters (here age and gender) in a synthesiser also opens up the possibility of manipulating these characteristics in a given voice, by changing these control parameters while keeping the speaker code fixed. This is similar to what multiple regression HMM approaches can achieve, and allows us to manipulate the gender and/or age of a given voice. Related regression techniques have previously been used to change perceived voice age in GMM-based singing voice conversion [25].

5. EXPERIMENTS

5.1. Experimental conditions

For our experiments, we used a corpus of studio-quality native Japanese speech uttered by 65 males and 70 females aged between 10 and 89. Of these, speech from 56 males and 56 females was used to train the multi-speaker DNN, with the remaining speakers (9 males and 14 females) held out for speaker adaptation, as listed in Table 1. The training-set speakers were chosen to be equally distributed for each age band (8 speakers for each age band and gender).

\(^2\)The same technique may also be used to estimate age and gender codes, both to adapt synthesisers and to infer speaker age/gender. However, space limitations prevented us from exploring this possibility in the current paper.

\(^3\)An infinite supply of novel voices can also be created by randomising new speaker codes, although this has been left for future work.
With approximately 100 utterances for each speaker, this yielded a total of 11,170 training-data utterances.

For the adaptation experiments, we used 100 utterances as adaptation material from each of 23 speakers not included in the training set. In this case, we were not able to balance all factors due to the limited number of speakers in the corpus, though we made our best efforts to make it as balanced as possible.

To evaluate the multi-speaker speech synthesis and speaker adaptation tasks, ten additional utterances per speaker were held out as test data for every speaker in the training and adaptation datasets.

The speech signal waveforms were sampled at 48 kHz, 16 bits. STRAIGHT [26] analysis was used to obtain 259-dimensional acoustic feature vectors every 5 ms, each comprising 60 mel-cepstral coefficients (with a bilinear frequency warping parameter of 0.77), a linearly interpolated fundamental frequency in the mel scale, and 25-dimensional band aperiodicities, along with their delta and delta-delta counterparts. The 259th feature was a binary voiced/unvoiced flag. During synthesis, the static and dynamic features were combined as described in [27] to produce the most likely speech trajectory sequence, based on a forced-alignment against the held-out natural speech (i.e., an oracle duration model was used).

For the input linguistic features, Open JTalk\footnote{http://open-jtalk.sourceforge.net/} was used to perform standard analysis of Japanese text for synthesis, including grapheme-to-phoneme conversion, part-of-speech tagging, and morphological analysis with the McCab parser (based on [28]). The derived quin-phone identity and syntactic and prosodic text information were encoded into a 389-dimensional mixed discrete-and-numeric vector of linguistic features. This input vector was then augmented with the three types of auxiliary features mentioned earlier and used as the input to the neural network speech synthesiser.

All acoustic models were feedforward DNNs with five hidden layers of 1024 nodes each. Sigmoid activation functions were used for all units in the hidden and output layers. The models were initialised randomly and trained to minimise mean square error using stochastic gradient descent for 10 epochs with the learning rate fixed at 0.2 and used as the input to the neural network speech synthesiser. Sigmoid activation functions were used for all units in the hidden and output layers. The models were initialised randomly and trained to minimise mean square error using stochastic gradient descent for 10 epochs with the learning rate fixed at 0.2 and used as the input to the neural network speech synthesiser.
adaptation schemes. Contrasting the average voice ONE-a against
crinant code used for adaptation increased.

Figure 3 (b) shows that, unlike the multi-speaker task, both mel-cepstral
code dimensionality we tested achieved better objective results. Fig-
ure 3 (a) and (b) show that, unlike the multi-speaker task, all one-hot vector elements
were assigned their average value. We view this as the average, reference
voice of our system. Notably, this average voice was signifi-
cantly less accurate than all other systems, showing that the multi-
speaker system overall was successful at approximating the many different speakers in the corpus. In general, the multi-speaker sys-
tems showed relatively small objective differences, though we note that the $F_0$ RMSE slightly improved for both RND and DCC when reducing the speaker code (S) dimension to 8. The effect of the DCC size on objective measures is explored in Figure 3 (a). Interestingly, while the mel-cepstral distortion steadily decreases as the DCC size increases, the $F_0$ RMSE is the lowest when the DCC size is 8.

Subjective test results for quality and speaker similarity, with 95% confidence intervals based on Student’s t-distribution, are plotted in Figures 2 (a) and (b), respectively. All models achieved sim-
ilar scores with no statistically significant differences in either test.
However, since ONE-ccc’ got the poorest scores for both quality and similarity, we view the numerical encodings of age and gender as preferred, especially since those also are easier to manipulate.

5.3. Evaluation of the adaptation task

To evaluate the adaptation task, we used a similar objective eval-
uation as for the multi-speaker task. We also employed the same labelling convention, with one addition: the letter ‘c’ indicates a fea-
ture that was estimated using the method proposed in Section 3. The results are summarised in Table 3 (b). As mentioned earlier, only the speaker code was estimated in these experiments.

Like before, the first system in the table, ONE-a, is an average,
reference voice based on ONE-S. It can be seen that all seven mod-
els, and thus all three types of speaker code, achieved significantly lower errors than the average voice by adapting to the 23 different target speakers. Compared to the known voices in Table 3 (a), we see that the performance was slightly worse on unknown voices, which is not unexpected. For both RND and DCC we note that the larger code dimensionality we tested achieved better objective results. Fig-
ure 3 (b) shows that, unlike the multi-speaker task, both mel-cepstral distortion and $F_0$ RMSE decreased as the dimensionality of the discriminant code used for adaptation increased.

Figure 4 shows the results of our AB tests comparing different adaptation schemes. Contrasting the average voice ONE-a against the adapted ONE-e (bottom test), listeners overwhelmingly preferred the adapted voice, confirming the efficacy of the adaptation.

5.4. Manipulating speech characteristics

Next we performed three types of manipulation of the input codes. The first manipulation was speaker interpolation, where we gradu-
ally interpolated between the speaker codes of two speakers. We also experimented with changing the gender or age codes while keeping
the other input codes constant; example manipulations can be found at http://www.hieuthi.com/papers/icassp2017/.

Although it is not straightforward to evaluate these manipula-
tions because correct references are not available, informally, we found that changing the inputs altered the perception of the voice in a plausible manner for gender conversion, including some changes to the pitch. Similar to [25], modifying speaker age inputs also pro-
duced plausible audible differences, at least for larger changes. Manipu-
lation effects were generally more pronounced when RND and DCC speaker codes were of lower dimensionality, with results ap-
pear to be most consistent for the model DCC008-SGA.

6. CONCLUSIONS

In this paper, we considered augmenting the conventional text-
derived inputs of DNN-based speech synthesis acoustic models with various input codes. We showed that a single DNN with input codes simultaneously was able to 1) learn a useful multi-speaker model even from a speech database with over 100 speakers and only a lim-
ited amount of speech from each; 2) perform adaptation to novel speakers; and 3) meaningfully control the synthetic speech output. This unifies previous approaches to adaptation and control in DNN-
based synthesis. We also found that using gender and age codes im-
proved speaker adaptation performance. Several speaker-code vari-
ations were investigated, finding that the most appropriate encoding
and code size may depend on the task and the acoustic feature(s) considered. As future work, it would be interesting to consider sim-
ilar input code setups for BLSTM-based acoustic models [29, 24].
7. REFERENCES


