A study of speaker adaptation for DNN-based speech synthesis

Zhizheng Wu Pawel Swietojanski Christophe Veaux
Steve Renals Simon King

The Centre for Speech Technology Research, University of Edinburgh, United Kingdom
zhizheng.wu@ed.ac.uk

Abstract

A major advantage of statistical parametric speech synthesis (SPSS) over unit-selection speech synthesis is its adaptability and controllability in changing speaker characteristics and speaking style. Recently, several studies using deep neural networks (DNNs) as acoustic models for SPSS have shown promising results. However, the adaptability of DNNs in SPSS has not been systematically studied. In this paper, we conduct an experimental analysis of speaker adaptation for DNN-based speech synthesis at different levels. In particular, we augment a low-dimensional speaker-specific vector with linguistic features as input to represent speaker identity, perform model adaptation to scale the hidden activation weights, and perform a feature space transformation at the output layer to modify generated acoustic features. We systematically analyse the performance of each individual adaptation technique and that of their combinations. Experimental results confirm the adaptability of the DNN, and listening tests demonstrate that the DNN can achieve significantly better adaptation performance than the hidden Markov model (HMM) baseline in terms of naturalness and speaker similarity.

Index Terms: Speech synthesis, acoustic model, deep neural network, speaker adaptation

1. Introduction

A significant amount of effort have been made to improve the naturalness of speech synthesis through the annual Blizzard Challenge[1]. Apart from naturalness, a speech synthesis system is also expected to be able to generate an arbitrary speaker's voice with minimum training/adaptation data. To respond this issue, speaker adaptation and voice conversion techniques have been developed for the two mainstream speech synthesis techniques, statistical parametric speech synthesis (SPSS) and unit selection speech synthesis, respectively. Due to the robust performance of speaker adaptation, a major advantage of SPSS over unit-selection speech synthesis is its flexibility in changing speaker characteristics, speaking styles and emotions [1], and there has been a significant improvement in naturalness in recent years [2].

Hidden Markov model (HMM) speech synthesis has dominated SPSS in the past decade. Many speaker adaptation techniques have been explored to improve naturalness and the degree of speaker similarity for HMM speech synthesis. These techniques can be grouped into two categories: maximum likelihood linear regression (MLLR) [3] and maximum a *posteriori* (MAP) [4] adaptation. The family of MLLR techniques attempt to learn a linear transformation that can transform average voices to sound like a target speaker, while MAP techniques employ speaker-independent (or speaker-clustered) models as a prior distribution to estimate the target speaker model. These speaker adaptation techniques have been shown to be effective in mimicking a target speaker's voice using a small amount of adaptation data [5].

Recently, deep neural networks (DNNs) have re-emerged as potential more powerful acoustic models for SPSS following the success in automatic speech recognition [6], as DNNs can learn complex mappings from linguistic features to acoustic features. Several independent studies have shown that DNNs can produce more natural synthesised speech than the conventional HMM-based speech synthesis for a single speaker in various training conditions [7, 8, 9, 10, 11, 12, 13, 14], but only few studies have addressed the question of whether DNN-based speech synthesis can offer adaptation techniques of similar flexibility to HMM-based speech synthesis – even though there has been successful work in this area in the context of DNN-based speech recognition [15, 16, 17, 18, 19, 20]. A preliminary study was conducted on speaker adaptation for DNN synthesis in [21], but only a feature transformation was used to modify the output of the DNN.

In this work, we conduct a systematic study of speaker adaptation techniques for DNN-based speech synthesis. As discussed in [20], there are three ways to adapt a neural network. The first way is to perform feature space transformations, the second one is to augment speaker-specific features as input to neural nets, and the last one is to perform model adaptation, that is to modify neural network parameters directly. In this paper, we perform speaker adaptation at different levels. In particular, at the input level, we augment an i-vector to represent speaker identity, do model adaptation using the learning hidden unit contributions (LHUC) [20] at the middle level and perform feature space transformations at the output level. As these adaptation techniques are performed at different levels, they may be usefully combined. We have performed experimental analysis on the performance of each individual adaptation technique and that of their combinations.

2. DNN adaptation

As discussed above, we perform speaker adaptation at three different levels (Fig. 1): At the input layer, we augment a speaker-specific vector, namely i-vector, with linguistic features; at the middle model level, we employ the recently proposed learning hidden unit contributions approach (LHUC) [20] to perform model-based adaptation; and at the output layer, we use a linear transformation approach to perform feature space adaptation, that is to modify the output of a DNN directly. These individual adaptation techniques are briefly introduced in following paragraphs.

[1] http://www.synsig.org/index.php/Blizzard_Challenge

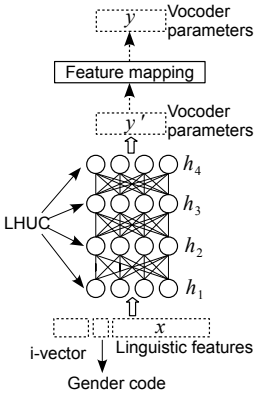


Figure 1: Three ways to do speaker adaptation for DNN-based speech synthesis. LHUC=Learning Hidden Unit Contributions.

2.1. i-vector

An i-vector is a low-dimensional vector representing speaker identity. I-vectors have dramatically improved the performance of text-independent speaker verification and now define the state-of-the-art [22]. Given a speaker-dependent GMM, the corresponding mean supervector $\mathbf{s}$ can be represented as,

$$\mathbf{s} \approx \mathbf{m} + \mathbf{T}\mathbf{i}, \quad \mathbf{i} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{1}$$

where $\mathbf{m}$ is the super-vector defined by the mean super-vector of a speaker-independent universal background model (UBM) that benefits from multiple speakers training corpora, $\mathbf{s}$ is the speaker super-vector which is the mean super-vector of the speaker-dependent GMM model (adapted from the UBM), $\mathbf{T}$ is the total variability matrix estimated on the background data, and $\mathbf{i}$ is the speaker identity vector, also called the *i-vector*.

In the context of DNN synthesis, when training a speaker-independent DNN model – an average voice model (AVM) – linguistic features are augmented with an i-vector as an additional to capture speaker identity. With the augmented i-vector, inputs with the same linguistic content but spoken by different speakers can be distinguished. At the adaptation phase, the target speaker's i-vector is first estimated by using the adaptation data and the total variability $\mathbf{T}$ through Eq. (1), and then the i-vector is appended with linguistic features as input to generate the target speaker's voice. As suggested in the literatures [23], length normlisation is performed on all the i-vectors. In practice, we used the ALIZE toolkit [24] to extract i-vectors.

2.2. Learning hidden unit contribution (LHUC)

An average voice, or speaker independent, DNN is build using a number of hidden units hierarchically structured into a sequence of layers implementing some non-linear transformations. Each such unit, at training stage, act as an adaptable basis function capturing certain patterns in its inputs. A learning process of all the units in the model is driven by a single objective (e.g., to minimize the squared error as in this work) and the units, in order improve the objective, are encouraged to specialize and become complementary to each other in explaining different patterns in training data well – they learn some joint representation of the problem the model was tasked to solve.

However, when the model is applied to unseen data, the relative importance of particular units may no longer be optimal. LHUC, given adaptation data, rescales the contributions (amplitudes) of the hidden units in the model without actually modifying their feature receptors. In this work, contrary to [20], we use an unconstrained variant of LHUC, i.e. its amplitudes are not re-parametrised in any way. This decision was motivated by preliminary results and the need to limit the number of comparisons for evaluation tests.

Another way to re-weight hidden units is their interpolation within pooling regions [25], however, this method is model dependent requiring certain differentiable pooling operators to be implemented across layers, while LHUC is model-agnostic and can work with arbitrary model non-linearities [20] and architectures [26].

2.3. Feature space transformation

At the output level, we perform a feature space transformation to modify the output of a DNN as

$$\mathbf{y} \approx \mathcal{F}(\mathbf{y}'), \tag{2}$$

where $\mathbf{y}'$ is the output feature of a DNN, $\mathbf{y}$ is the reference target vocoder parameter, and $\mathcal{F}(\cdot)$ is the transformation function.

For adaptation, the AVM is first used to predict a sequence of vocoder parameters given a sequence of linguistic features extracted from adaptation material. Then, a transformation function is built based the parallel data: predicted vocoder parameters and reference vocoder parameters of the target speaker. At runtime synthesis, the transformation function is applied to the predicted vocoder parameters of the AVM DNN to perform feature space adaptation to mimic the target speaker's voice.

In practice, we employed joint density Gaussian mixture model (JD-GMM) with full-covariance matrices to implement the transformation function, since JD-GMM is current state-of-the-art voice conversion technique [27], and can perform feature space mapping well.

3. Experiments

3.1. Experimental setup

In the experiments, we used the Voice Bank corpus [28] to assess the performance of the adaptation techniques. 96 speakers – 41 male and 55 female – were used to train a DNN average voice model (AVM). Two speakers, one male and one female, were used as target speakers for speaker adaptation. We considered two training conditions: 10 utterances and 100 utterances of adaptation data for both target speakers. 70 utterances were used as a development set and 72 utterances were used as a testing set for both target speakers.

The sampling rate of the corpus was 48 kHz. The STRAIGHT vocoder [29] was employed to extract 60-dimensional Mel-Cepstral Coefficients (MCCs), 25 band aperiodicities (BAPs) and F_0 in log-scale at 5 msec step.

We trained context-dependent hidden Semi-Markov models (HSMMs) as the baseline. The HSMMs have 5 states with separate output distributions representing the 60-D MCCs, 25-D BAPs, F_0 in log-scale and their delta and delta-features. The global variances are also used to refine the parameter trajectories using the Maximum likelihood parameter generation (MLPG) algorithm, and spectral enhancement post-filtering is applied to the MCCs. Finally, separate decision trees are used to cluster the state duration probabilities and the state output probabilities using input linguistic features such as quinphone, part-of-speech, phoneme, syllable, word and phrase positions. The 96 speakers of the Voice Bank corpus are used for learning an average voice model and the CSMAPLR algorithm is employed for adaptation [5].

Table 1: Objective results of DNN adaptation techniques. MCD and BAP are Mel-Cepstral Distortion and Band APeriodicity distortion, respectively. V/UV error means frame-level voiced/unvoiced swipping error. Root Mean Squared Error (RMSE) of F_0 was calculated in linear frequency.

DNN adaptation			10 utterances adaptation				100 utterances adaptation			
i-vector	LHUC	FT	MCD (dB)	BAP (dB)	F_0 RMSE (Hz)	V/UV error rate (%)	MCD (dB)	BAP (dB)	F_0 RMSE (Hz)	V/UV error rate (%)
Y			6.56	2.68	25.99	14.51	6.38	2.63	26.07	14.33
	Y		5.72	2.44	24.54	11.77	5.58	2.39	23.97	11.16
		Y	5.57	2.45	24.51	13.39	5.28	2.38	24.51	13.10
Y	Y		5.93	2.46	26.84	12.43	5.98	2.47	26.23	12.15
Y		Y	5.66	2.49	26.02	14.51	5.30	2.39	25.69	14.33
	Y	Y	5.53	2.41	24.22	11.71	5.27	2.35	24.20	11.97
Y	Y	Y	5.60	2.43	25.82	12.43	5.31	2.37	24.98	12.71

The input of a DNN contained 592 binary linguistic features, 9 numerical features and 1 binary feature to represent gender information. The linguistic features included quinphone, part-of-speech, phoneme, syllable, word and phrase positions. The 9 numerical features involved frame position in the HMM state and phoneme, state position in phoneme and state and phoneme duration. We note that when applying i-vector based speaker adaptation, an i-vector was appended with the linguistic features. The output acoustic features comprised 60-D MCCs, 25-D BAPs, 1-D F_0, their corresponding delta and delta-delta features, and a voice/unvoiced binary value. In total, the acoustic feature vector was 259 dimension. F_0 was linearly interpolated before extracting dynamic features, and the V/UV feature was used to decide the voiced and unvoiced region at runtime synthesis. The input features were normalised to the range of [0.01, 0.99], and the output features were normalised by speaker-dependent mean and variance. Similar normalisation was applied to the adaptation data. We applied the maximum likelihood parameter generation (MLPG) algorithm to the output features to generate smoothed parameter trajectories, followed by spectral enhancement post-filtering in the cepstral domain.

The DNN systems had 6 hidden layers, and each hidden layer had 1536 units. A hyperbolic tangent function was used in the hidden layers followed by a linear activation at the output layer. During AVM training and LHUC adaptation, the mini-batch size was set to 256, and momentum was adopted to accelerate convergence. For the first 10 epochs, the momentum was set to 0.6, and was then increased to 0.9. A fixed learning rate of 0.0008 was used in the first 10 epochs for AVM DNN training. During LHUC adaptation, the learning rates were set to 0.06 for female speaker and 0.02 for male speaker for the first 10 epochs. In all cases, after 10 epochs, the learning rates were halved at each epoch. L2 regularization was applied to the weights with a penalty factor of 0.00001. The maximum number of epochs was set to 30 for AVM DNN training and LHUC adaptation. In the implementation, we used the CUDAMat library[2] which is a Python module for matrix calculations on a GPU using CUDA [30].

To extract i-vectors, the 96 speakers' data were used to train gender-dependent UBM and total variability. The dimensionality of i-vectors was set to 32. In the implementation of feature transformation, single mixture full-covariance JD-GMMs were trained for 10 utterances adaptation condition, while 4 mixture JD-GMMs were trained when adapting with 100 utterances.

3.2. Objective evaluation

We conducted objective evaluation to analyse the performance of each individual adaptation technique as well as their combinations. Even though objective results might not correlate with the perceived naturalness and speaker similarity, they are good predictors to optimise DNN hyper-parameters. Here, we only report the average distortions of the two target speakers.

The results are presented in Table 1. Across the three individual adaptation techniques, the feature transformation (FT) approach achieves the lowest Mel-Cepstral Distortions (MCDs) and BAP distortions, and LHUC achieves the lowest F0 RMSE and V/UV error rates, suggesting complementarity. In all cases, i-vector gives the highest distortions.

When combining two individual adaptation techniques, LHUC+FT gives the lowest distortions for all the measures and under all the adaptation conditions. Surprisingly, when i-vector is integrated with LHUC or FT, i-vector+LHUC and i-vector+FT achieve higher distortions than LHUC and FT, respectively.

We then compared the performance of these adaptation techniques by using different amounts of adaptation materials. When the adaptation data is expanded from 10 utterances to 100 utterances, all the techniques or combinations reduce the distortions except i-vector+LHUC. I-vector+LHUC slightly increases the MCD from 5.93 dB to 5.98 dB.

3.3. Subjective evaluation

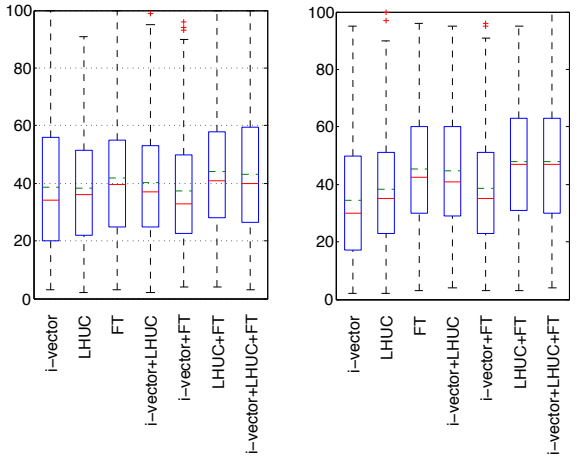
We conducted listening tests to assess the naturalness and speaker similarity of the synthesised speech obtained using various combinations of adaptation techniques.

We first evaluated the adaptation performance of DNN systems. Four MUSHRA (MUltiple Stimuli with Hidden Reference and Anchor) tests were conducted to assess the naturalness and speaker similarity. 30 native English listeners participated in each test. Each listener rated 20 sets which were randomly selected from the testing utterances, and each set consisted of 8 stimuli of the same sentence generated by each of the seven adaptation systems plus the copy-synthesis speech used as the hidden reference. The listeners were asked to rate each stimulus from 0 (extremely bad for naturalness test or totally different speaker for similarity test) to 100 (same naturalness as the reference speech or same speaker identity as the reference speech), and they were also instructed to give exactly one of the 8 stimuli in every set a rating of 100 in both naturalness and similarity tests.

The MUSHRA scores for all the DNN-based adaptation techniques that use 10 utterances as adaptation data are presented in Fig. 2. We used a paired t-test to examine the significance between systems. In the naturalness test, across the three individual adaptation techniques, FT achieves significantly better performance than i-vector and LHUC, and i-vector is slight better than LHUC, but the difference is not significant. When combining any two adaptation techniques, LHUC+FT achieves significantly better performance than other combina-

[2] https://github.com/cudamat/cudamat

tions, and i-vector+LHUC is significantly better than i-vector + FT. Even though i-vector+LHUC is slightly better than i-vector and LHUC, but the differences are not significant. Surprisingly, i-vector+FT is slightly worse than FT or i-vector, even though i-vector+FT achieves much better objective results than i-vector. The difference between i-vector+FT and i-vector is not significant, but the difference between i-vector+FT and FT is significant. When combining all the adaptation techniques, surprisingly, the resulted naturalness is almost the same as LHUC+FT. The performance of i-vector+LHUC+FT is significantly better than other systems except FT and LHUC+FT.



(a) Naturalness (b) Similarity

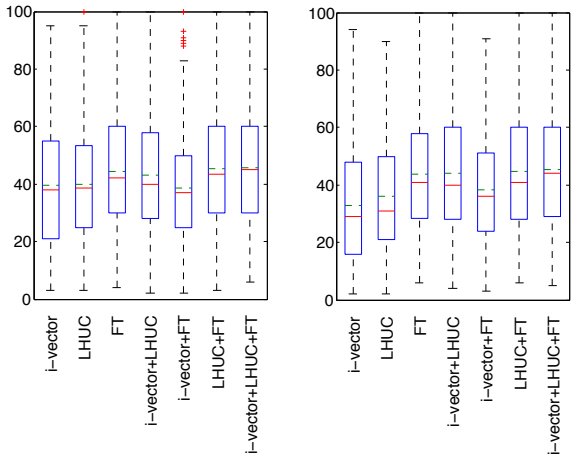
Figure 2: Box plot of MUSHRA results for DNN systems with 10 utterance adaptation data. (a) naturalness test results; and (b) speaker similarity test results.

The observations based on the similarity test are similar to that in the naturalness test, but here LHUC is significantly better than i-vector. i-vector+FT is still not as good as FT, but it is significantly better than i-vector. The difference between i-vector+LHUC+FT and LHUC+FT is not significant, but they are significantly better than all the other systems.

The MUSHRA scores for all the DNN-based adaptation techniques that use 100 utterances as adaptation data are shown in Fig. 3. The trend is similar to that using 10 utterances for adaptation. Comparing with the objective results, one interesting observation is that even though i-vector+FT achieves much lower spectral distortions than i-vector+LHUC in all conditions, the subjective results suggest that i-vector+LHUC is significantly better than i-vector+FT.

We then conducted preference tests to compare the naturalness and speaker similarity between the DNN and HMM adapted systems. Here we use the DNN systems with i-vector+LHUC+FT adaptation as they achieved relative better performance than other adaptation techniques. 27 listeners participated in each test. In the naturalness test, each listener listened to two samples generated by either DNN or HMM, and was asked to choose the one they preferred. In the similarity test, each listener first listened to the reference target speech, and then listened two samples generated by either DNN or HMM, after that, was asked to choose the one which is closer to the target speech.

The preference results are presented in Fig. 4. It is observed that in all the adaptation conditions, DNN achieves significantly



(a) Naturalness (b) Similarity

Figure 3: Box plot of MUSHRA results for DNN systems with 100 utterance adaptation data. (a) naturalness test results; and (b) speaker similarity test results.

better performance than HMM baseline in terms of naturalness and speaker similarity. This confirms the adaptability of DNN, and shows the effectiveness of i-vector, LHUC and FT based adaptation techniques.

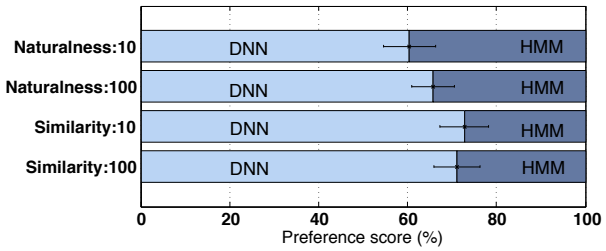


Figure 4: Preference scores between DNN and HMM adaptations. 10 and 100 mean the number of adaptation utterances.

4. Conclusions

In this paper, we performed a systematic and experimental analysis of speaker adaptation for deep neural network (DNN) based speech synthesis. The experimental results confirmed the flexibility of DNN-based synthesis, also demonstrated that DNN-based adaptation can achieve even better performance than HMM-based adaptation. We also found that feature transformation at the output layer works well and the adaptation performance can be improved by combining with model based adaptation in this work the learning hidden unit contributions (LHUC). However, even though experimental results show that i-vector with combined LHUC can achieve good performance, it does not work well as expected when combined with feature transformation. Further analysis is required to understand the phenomenon much better.

The samples and listening test results used in the experiments are available online via this link: http://dx.doi.org/10.7488/ds/259.

Acknowledgement:This work was supported by EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology).

5. References

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

[2] S. King and K. Karaiskos, "The blizzard challenge 2013," in *Blizzard Challenge Workshop*, 2013.

[3] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, 1995.

[4] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Trans. on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994.

[5] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, "Analysis of speaker adaptation algorithms for hmm-based speech synthesis and a constrained smaplr adaptation algorithm," *IEEE Trans. Audio, Speech and Language Processing*, vol. 17, no. 1, pp. 66–83, 2009.

[6] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "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.

[7] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

[8] Z.-H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using Restricted Boltzmann Machines and Deep Belief Networks for statistical parametric speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2129–2139, 2013.

[9] S. Kang, X. Qian, and H. Meng, "Multi-distribution deep belief network for speech synthesis," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

[10] Y. Qian, Y. Fan, W. Hu, and F. K. Soong, "On the training aspects of deep neural network (DNN) for parametric TTS synthesis," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014.

[11] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014.

[12] Y. Fan, Y. Qian, F. Xie, and F. K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Proc. Interspeech*, 2014.

[13] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015.

[14] B. Uría, I. Murray, S. Renals, and C. Valentini-Botinhao, "Modelling acoustic feature dependencies with artificial neural networks: Trajectory-RNADE," in *Proc IEEE ICASSP*, 2015.

[15] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

[16] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using I-vectors," in *Proc IEEE ASRU*, 2013, pp. 55–59.

[17] P. Karanasou, Y. Wang, M. J. Gales, and P. C. Woodland, "Adaptation of deep neural network acoustic models using factorised i-vectors," in *Proc. Interspeech*, 2014.

[18] S. Xue, O. Abdel-Hamid, H. Jiang, L. Dai, and Q. Liu, "Speaker adaptation of deep neural network based on discriminant codes," *IEEE Trans. Audio, Speech and Language Processing*, vol. 22, no. 12, pp. 1713–1725, 2014.

[19] V. Gupta, P. Kenny, P. Ouellet, and T. Stafylakis, "I-vector-based speaker adaptation of deep neural networks for french broadcast audio transcription," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014.

[20] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *Proc. IEEE Spoken Language Technology Workshop*, 2014.

[21] B. Potard, P. Motlicek, and D. Imseng, "Preliminary work on speaker adaptation for dnn-based speech synthesis," Idiap, Tech. Rep., 2015.

[22] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[23] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems." in *Proc. Interspeech*, 2011.

[24] A. Larcher, J.-F. Bonastre, B. G. Fauve, K.-A. Lee, C. Lévy, H. Li, J. S. Mason, and J.-Y. Parfait, "ALIZE 3.0-open source toolkit for state-of-the-art speaker recognition." in *Proc. Interspeech*, 2013.

[25] P. Swietojanski and S. Renals, "Differentiable pooling for unsupervised speaker adaptation," in *Proc. ICASSP*, 2015.

[26] O. Abdel-Hamid and H. Jiang, "Rapid and effective speaker adaptation of convolutional neural network based models for speech recognition." in *Proc. Interspeech*. ISCA, pp. 1248–1252.

[27] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio, Speech and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[28] C. Veaux, J. Yamagishi, and S. King, "The voice bank corpus: Design, collection and data analysis of a large regional accent speech database," in *Proc. Int. Conf. Oriental COCOSDA*, 2013.

[29] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech communication*, vol. 27, no. 3, pp. 187–207, 1999.

[30] V. Mnih, "Cudamat: a cuda-based matrix class for python," University of Toronto, Tech. Rep., 2009.