VOICE CONVERSION USING DEEP NEURAL NETWORKS WITH SPEAKER-INDEPENDENT PRE-TRAINING

Seyed Hamidreza Mohammadi and Alexander Kain

Center for Spoken Language Understanding, Oregon Health & Science University
Portland, OR, USA

mohammah@ohsu.edu, kaina@ohsu.edu

ABSTRACT

In this study, we trained a deep autoencoder to build compact representations of short-term spectra of multiple speakers. Using this compact representation as mapping features, we then trained an artificial neural network to predict target voice features from source voice features. Finally, we constructed a deep neural network from the trained deep autoencoder and artificial neural network weights, which were then fine-tuned using back-propagation. We compared the proposed method to existing methods using Gaussian mixture models and frame-selection. We evaluated the methods objectively, and also conducted perceptual experiments to measure both the conversion accuracy and speech quality of selected systems. The results showed that, for 70 training sentences, frame-selection performed best, regarding both accuracy and quality. When using only two training sentences, the pre-trained deep neural network performed best, regarding both accuracy and quality.

Index Terms— voice conversion, pre-training, deep neural network, autoencoder

1. INTRODUCTION

To solve the problem of voice conversion (VC), various methods have been proposed. Most methods are generative methods which parametrize speech in short-time segments and map source speaker parameters to target speaker parameters [1]. One group of generative VC approaches use Gaussian mixture models (GMM). GMMs perform a linear multivariate regression for each class and weight each individual linear transformation according to the posterior probability that the input belonged to a specific class [2]. Kain and Macon [3] proposed to model the source and target spectral space jointly, using a joint-density GMM (JDGMM). This approach has the advantage of training mixture components based on the source-target feature space interactions. Toda et al. [4] extended this approach by using a parameter generation algorithm, which extends modeling to the dynamics of feature trajectories.

Another group of generative VC approaches use artificial neural networks (ANNs). Simple ANNs have been used for transforming short-time speech spectral features such as formants [5], line spectral features [6], mel-cepstrum [7], log-magnitude spectrum [8] and articulatory features [9]. Various ANN architectures have been used for VC: ANNs with rectified linear unit activation functions [8], bidirectional associative memory (a two-layer feedback neural network) [10], and restricted Boltzman machines (RBMs) and their variations [11, 12, 13]. In general, both GMMs and ANNs are universal approximators [14, 15]. The non-linearity in GMMs stems from forming the posterior-probability-weighted sum of class-based linear transformations. The non-linearity in ANNs is due to non-linear activation functions (see also 2.3). Laskar et al. [16] compared ANN and GMM approaches in the VC framework in more detail.

Recently, deep neural networks (DNNs) have shown performance improvements in the fields of speech recognition [17] and speech synthesis [18, 19, 20]. Four-layered DNNs have been previously proposed for VC but no significant difference was found between using a GMM and a DNN [6]. More recently, three-layered DNNs have achieved improvements in both quality and accuracy over GMMs when trained on 40 training sentences [7]. The previous two approaches use DNNs with randomly weight initialization; however, it has been shown in the literature that DNN training converges faster and to a better-performing solution if their initial parameter values are set via pre-training instead of random initialization [21]. Pre-training methods use unsupervised techniques such as stacked RBMs and autoencoders (AEs) [22, 23].

Pre-trained DNNs have also been applied to VC in a recent study [13], in which stacked RBMs were used to build high-order representations of cepstra for each individual speaker, using 63 minutes of speech for training the RBMs. The source speaker's representation features were then converted to the target speaker's representation features using ANNs, and the combined network was fine-tuned. However, we speculate that their approach may not be feasible for a small number of training sentences because (1) it employs high-dimensional features, and (2) it requires training of two separate RBMs, one for the source and one for the target speaker. To address these shortcomings, we propose to (1) train a deep autoencoder (DAE) for deriving *compact* representations of speech spectral features, and (2) to train the DAE on *multiple speakers* (which will not be included in VC training and testing), thus creating a speaker-independent DAE. The trained DAE will later be used as a component during the pre-training of the final DNN.

The remainder of the paper is organized as follows: In Section 2, we describe the network architectures used in this study. In Subsection 2.1, we explain the architecture of shallow ANNs. In Subsection 2.2, we explain the speaker-independent DAE architecture. In Subsection 2.3, we explain the architecture of the final DNN. In Section 3, we present the evaluations that were performed to compare the proposed architecture to baseline methods. First, in Subsection 3.1, we will explain all the design decisions and system configurations. Then, in Subsection 3.2, we present the objective evaluations. The subjective evaluations are presented in Subsection 3.3. The conclusion of the study is presented in Subsection 4.

This material is based upon work supported by the National Science Foundation under Grant No. 0964468.

2. NETWORK ARCHITECTURES

2.1. Artificial Neural Network

In this section, let $\mathbf{X}_{N\times D} = [\mathbf{x}_1, ..., \mathbf{x}_N]^\top$, where $\mathbf{x} = [x_1, \ldots, x_D]^\top$, represent N examples of D-dimensional source feature training vectors. Using a parallelization method (e. g. time-alignment and subsequent interpolation), we can obtain the associated matrix $\mathbf{Y}_{N\times D} = [\mathbf{y}_1, ..., \mathbf{y}_N]^\top$, representing target feature training vectors.

An ANN consists of K layers where each layer performs a linear or non-linear transformation. The k^{th} layer performs the following transformation,

$$\mathbf{h}_{k+1} = f(\mathbf{W}_k\mathbf{h}_k + \mathbf{b}_k), \tag{1}$$

where $\mathbf{h}_k$, $\mathbf{h}_{k+1}$, $\mathbf{W}_k$, $\mathbf{b}_k$, are the input, output, weights, bias of the current layer, respectively, and f is an activation function. By convention, the first layer is called the input layer (with $\mathbf{h}_1 = \mathbf{x}$), the last layer is called the output layer (with $\hat{\mathbf{y}} = \mathbf{h}_{K+1}$), and the middle layers are called the hidden layers. The objective is to minimize an error function, often the mean squared error

$$E = \|\mathbf{y} - \hat{\mathbf{y}}\|^2 . \tag{2}$$

The weights and biases can be trained by minimizing the error function using stochastic gradient descent. The back-propagation algorithm is used to propagate the errors to the previous layers. In this study, we use a two-layered ANN as mapping function (see Figure 1a) during pre-training of the DNN.

2.2. Deep Autoencoder

ANNs are usually trained with a supervised learning technique, in which we have to know the output values (in our case target speaker features), in addition to input values (in our case source speaker features). An AE is a special kind of neural network that uses an unsupervised learning technique, i. .e. we only need to know the input values. In the AE, the output values are set to be the same as the input values. Thus, the error criterion becomes a reconstruction criterion with the goal of reconstructing the input using the neural network, allowing the AE to learn an efficient encoding of the data. This unsupervised learning technique has proven to be effective for determining the initial network weight values for the task of supervised deep neural network training; this process is called pre-training.

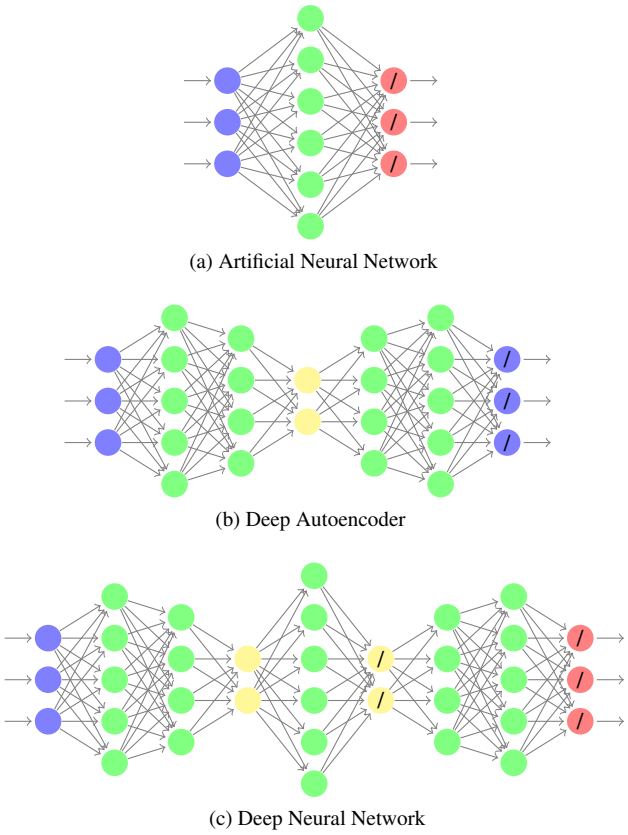
A simple AE has an identical architecture of a two-layered ANN. The first layer is usually called the encoding layer and the second layer is called the decoding layer. The encoding part of a simple AE maps the input to an intermediate hidden representation. The decoder part of an AE reconstructs the input from the intermediate representation. The first and second layers' weights are tied $\mathbf{W}_1 = \mathbf{W}_2^\top$, where $\top$ represents matrix transpose.

The task of an AE is to reconstruct the input space. During AE training in its simplest form, weights are optimized to minimize the average reconstruction error of the data

$$E = \|\mathbf{h}_1 - \mathbf{h_3}\|^2 , \tag{3}$$

where $\mathbf{h}_3$ is the output of the last layer of the network when the input is $\mathbf{h}_1 = \mathbf{x}$. However, this training schema may not result in extracting useful features since it may lead to over-fitting. One strategy to avoid this phenomenon is to modify the simple reconstruction criterion to the task of reconstruction of clean input from noise-corrupted input [23]. The de-noising error function is

$$E = \|\mathbf{x} - \mathbf{h_3}\|^2 , \tag{4}$$



(a) Artificial Neural Network

(b) Deep Autoencoder

(c) Deep Neural Network

Fig. 1: Network architectures. The color of the nodes represent: blue for input features, red for output features, yellow for compact features, and green for hidden/intermediate values. Layers include a non-linear activation function, unless labeled with a diagonal line, indicating a linear activation function.

where $\mathbf{h}_3$ is the output of the last layer of the network when the input is $\mathbf{h}_1 = \mathbf{x} + \mathbf{n}$, and $\mathbf{n}$ is a Gaussian corruptor.

In this study, we compute a compact representation of spectral features using a stacked de-noising autoencoder (DAE). We obtain a deep structure by training multiple AEs layer-by-layer and stacking them [23]. The first AE is trained on the input. The input is then encoded and passed to the next AE, which is trained on these encoded values, and so on. Finally, the AEs are stacked together to form a DAE, as shown in Figure 1b.

2.3. Deep Neural Network

Having an ANN with more than two layers ($K > 2$) could allow the network to capture more complex patterns. Typically, the higher number of parameters makes parameter estimation more difficult, especially if we start the training from random initial weight and bias values. In the following experiment, we will create a DNN with a structure that is equivalent to first encoding the spectral features using DAE, then mapping the compact intermediate features using a shallow ANN, and finally decoding the mapped compact features using the DAE (see Figure 1c). The entire structure can effectively be regarded as a pre-trained DNN, whose parameters are further fine-tuned by back-propagation (without any weight tying).

feature \ mapping	FS	GMM	ANN	DNN
MCEP	**6.83 (0.31)**	6.90 (0.31)	6.85 (0.34)	**6.83 (0.31)**
DMCEP	7.05 (0.28)	6.93 (0.29)	6.89 (0.29)	-

(a) large training set

feature \ mapping	FS	GMM	ANN	DNN
MCEP	7.60 (0.35)	8.31 (0.29)	7.58 (0.28)	**7.40 (0.30)**
DMCEP	7.57 (0.31)	7.90 (0.29)	7.46 (0.26)	-

(b) small training set

Table 1: Average test error between converted and target mel-warped log-spectra in dB (with standard deviations in parentheses).

3. EXPERIMENT

3.1. Training

A corpus of eleven speakers was used in this study. Of these, approximately 1–2 hours of speech of seven speakers was used for training the speaker-independent DAE. The remaining four speakers (two males: M1, M2, two females: F1, F2) were used for testing the DAE, and for training and testing the voice conversion system. We selected two Harvard sentences as a "small" training set, and 70 Harvard sentences as a "large" training set. For testing, we used 20 conversational sentences. We considered four different conversions: two intra-gender (M1→M2, F2→F1) and two cross-gender (M2→F2, and F1→M1).

We used the SPTK toolkit [24] to extract the 24^{th}-order mel-cepstrum (MCEP). The DAE is composed of three stacked AEs with sizes 100, 40, 15. The first AE is a de-noising AE with a Gaussian corruptor [23]. The second and third AEs are contractive AEs, which have shown to outperform other regularized AEs [25]. The activation functions are sigmoid, except for the last layer, which uses a linear activation function. The number of iterations during training of each AE was set to 1,000 with a mini-batch size of 20. The test error of the network is monitored using a portion of the corpus that is excluded from training. The learning rate was set to 0.01 and decayed in each iteration. We refer to the compact features at the last layer as deep MCEPs (DMCEPs).

We used four mapping methods in our experiment: Frame selection (FS) [26], JDGMM [4], two-layered ANN, and the proposed DNN. FS is a memory-based approach similar to the unit-selection approach in text-to-speech synthesis. Hyper-parameters (e. g. the number of mixture components of the JDGMM) were determined based on cross-validation objective scores and informal perceptual tests. For training the DNN, we first trained ANNs that map DMCEPs derived from the source and target speakers. Then, the final DNN was constructed by concatenating the encoding DAE, followed by the mapping ANN, and finally the decoding DAE, using the original networks' weights and biases. The DNN is then fine-tuned using back-propagation with a mini-batch size of 20 and learning rate of 0.002. The network error was monitored, and training was stopped before overfitting occurred. The DAE, the ANN, and the DNN were trained using the *pylearn2* toolkit [27].

3.2. Objective Evaluation

We performed objective evaluations using the mel-scaled log-spectral distance in dB. First, we measured the reconstruction error of the trained DAEs on the four voice conversion speakers' test set; the average error was 2.12 dB. Second, we trained the four mapping models on the small training set and on the large training set, for each of the four conversions. We then compared the conversion outputs and the targets, averaged over all conversions. The results are shown in Table 1. As an upper bound, we calculated the average distance between the original source and target speakers' spectral envelopes to be 10.48 dB. For the large training set, we observed that, DNN and FS performed best of the four mapping methods, although the differences were not significant. For the small training set, the performance gap between DNN and other mapping methods is larger. This is likely due to the semi-supervised learning aspect of the DNN. Even using a shallow ANN on DMCEP features resulted in good performance, likely due to the efficient encoding produced by the DAE.

3.3. Subjective Evaluation

To subjectively evaluate voice conversion performance, we performed two perceptual tests: the first test measured speech quality and the second test measured conversion accuracy (also referred to as speaker similarity between conversion and target). The listening experiments were carried out using Amazon Mechanical Turk [28], with participants who had approval ratings of at least 90% and were located in North America. We have omitted the ANN mapping method to reduce the complexity of the subjective evaluation.

3.3.1. Speech Quality Test

To evaluate the speech quality of the converted utterances, we conducted a comparative mean opinion score (CMOS) test. In this test, listeners heard two utterances A and B with the *same* content and the *same* speaker but in two *different* conditions, and are then asked to indicate wether they thought B was better or worse than A, using a five-point scale consisting of +2 (much better), +1 (somewhat better), 0 (same), -1 (somewhat worse), -2 (much worse). The test was carried out identically to the conversion accuracy test. The two conditions to be compared differed in exactly one aspect (different features or different mapping methods). The experiment was administered to 40 listeners with each listener judging 20 sentence pairs. Three trivial-to-judge sentence pairs were added to the experiment to filter out any unreliable listeners.

Listeners' average response scores are shown in Figure 2. The VOC configuration represents the vocoded target (added as a baseline). We did not include FS for the small training set because the quality of the generated speech was poor as described in Section 3.2. The statistical analyses were performed using one-sample t-tests. For the large training set, FS performed statistically signifi-

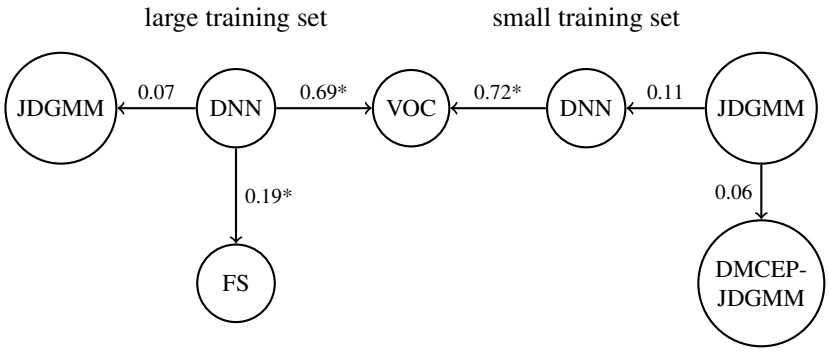


Fig. 2: Speech quality, with nodes showing a specific configuration, the edges showing comparisons between two configurations, the arrows pointing towards the configuration that performed better, and asterisks showing statistical significance.

cantly better compared to DNN ($p < 0.05$), which shows the effectiveness of memory-based approaches when sufficient data is available. JDGMM also performed slightly better than DNN, but not significantly. For the small training set, the results showed that using DMCEP features resulted in a slightly better quality score compared to MCEPs when a JDGMM was used. DNNs performed better but no statistical significant difference was found between DNN and JDGMM.

3.3.2. Conversion Accuracy Test

To evaluate the conversion accuracy of the converted utterances, we conducted a same-different speaker similarity test [29]. In this test, listeners heard two stimuli A and B with *different* content, and were then asked to indicate wether they thought that A and B were spoken by the *same*, or by two *different* speakers, using a five-point scale consisting of +2 (definitely same), +1 (probably same), 0 (unsure), -1 (probably different), and -2 (definitely different). One of the stimuli in each pair was created by one of the three mapping methods, and the other stimulus was a purely MCEP-vocoded condition, used as the *reference* speaker. Half of all pairs were created with the reference speaker identical to the target speaker of the conversion (the "same" condition); the other half were created with the reference speaker being of the same gender, but *not* identical to the target speaker of the conversion (the "different" condition). The experiment was administered to 40 listeners, with each listener judging 40 sentence pairs. Four trivial-to-judge sentence pairs were added to the experiment to filter out any unreliable listeners.

Listeners' average response scores (scores in the "different" conditions were multiplied by -1) are shown in Figure 3. The statistical analyses were performed using Mann-Whitney tests [30]. For the large training set, FS performed significantly better compared to JDGMM ($p < 0.05$). When compared to DNN, FS performed better but no statistically significant difference was found. DNN also performed better than JDGMM but the difference was also not statistically significant. The superiority of the FS method is due to the high number of sentences (70 sentences) that were available in the large training set. One of the problems of GMM and ANN approaches is that they average features to generate the target features. However, FS is a memory-based approach, and thus it performs better in this task because of the fact that raw (not averaged) frames were produced. This only works when the number of training samples is high enough that it will find appropriate frames most of the time. For the small training set, DNN achieved a statistically significant superior score compared to other configurations (all with $p < 0.05$).

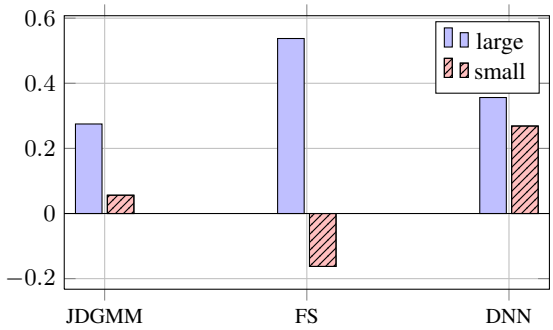


Fig. 3: Conversion accuracy, with blue solid bars representing the large training set, and red patterned bars representing the small training set.

As expected, FS performed poorly; there is not enough data in the small training case, and the search cannot find appropriate frames. The DNN performed statistically significantly better compared to JDGMM, which shows the robustness of DNN solutions when the training size is small. An interesting result is that, using only two sentences to train the DNN, we were able to match the conversion accuracy of JDGMM trained with 70 training sentences.

4. CONCLUSION

In this study we trained a speaker-independent DAE to create a compact representation of MCEP speech features. We then trained an ANN to map source speaker compact features to target speaker compact features. Finally, a DNN was initialized from the trained DAE and trained ANN parameters, which was then fine-tuned using back-propagation. Four competing mapping methods were trained on either a two-sentence or on a 70-sentence training set. Objective evaluations showed that the DNN and FS performed best for the large training set and DNN performed best for the small training set. Perceptual experiments showed that for the large training set, FS performed best regarding both accuracy and quality. For the small training set, the DNN performed best regarding both accuracy and quality. We were able to match the conversion accuracy of JDGMM trained with 70 sentences with the pre-trained DNN trained using only two sentences. These results are an example of the effectiveness of semi-supervised learning.

5. REFERENCES

[1] S. H. Mohammadi and A. Kain. Transmutative voice conversion. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6920–6924. IEEE, 2013.

[2] Y. Stylianou, O. Cappé, and E. Moulines. Continuous probabilistic transform for voice conversion. *IEEE Transactions on Speech and Audio Processing*, 6(2):131–142, March 1998.

[3] A. Kain and M. Macon. Spectral voice conversion for text-to-speech synthesis. In *Proceedings of ICASSP*, volume 1, pages 285–299, May 1998.

[4] T. Toda, A. W. Black, and K. Tokuda. Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. *IEEE Transactions on Audio, Speech, and Language Processing Journal*, 15(8):2222–2235, November 2007.

[5] M. Narendranath, H. A. Murthy, S. Rajendran, and B. Yegnanarayana. Transformation of formants for voice conversion using artificial neural networks. *Speech communication*, 16(2): 207–216, 1995.

[6] K. S. Rao, R. Laskar, and S. G. Koolagudi. Voice transformation by mapping the features at syllable level. In *Pattern Recognition and Machine Intelligence*, pages 479–486. Springer, 2007.

[7] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad. Spectral mapping using artificial neural networks for voice conversion. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(5):954–964, 2010.

[8] E. Azarov, M. Vashkevich, D. Likhachov, and A. Petrovsky. Real-time voice conversion using artificial neural networks with rectified linear units. In *INTERSPEECH*, pages 1032–1036, 2013.

[9] N. W. Ariwardhani, Y. Iribe, K. Katsurada, and T. Nitta. Voice conversion for arbitrary speakers using articulatory-movement to vocal-tract parameter mapping. In *Machine Learning for Signal Processing (MLSP), 2013 IEEE International Workshop on*, pages 1–6. IEEE, 2013.

[10] L. J. Liu, L. H. Chen, Z. H. Ling, and L. R. Dai. Using bidirectional associative memories for joint spectral envelope modeling in voice conversion. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014.

[11] L. H. Chen, Z. H. Ling, Y. Song, and L. R. Dai. Joint spectral distribution modeling using restricted boltzmann machines for voice conversion. In *INTERSPEECH*, 2013.

[12] Z. Wu, E. S. Chng, and H. Li. Conditional restricted boltzmann machine for voice conversion. In *Signal and Information Processing (ChinaSIP), 2013 IEEE China Summit & International Conference on*, pages 104–108. IEEE, 2013.

[13] T. Nakashika, R. Takashima, T. Takiguchi, and Y. Ariki. Voice conversion in high-order eigen space using deep belief nets. In *INTERSPEECH*, pages 369–372, 2013.

[14] D. M. Titterington, A. F. Smith, U. E. Makov, et al. *Statistical analysis of finite mixture distributions*, volume 7. Wiley New York, 1985.

[15] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[16] R. Laskar, D. Chakrabarty, F. Talukdar, K. S. Rao, and K. Banerjee. Comparing ann and gmm in a voice conversion framework. *Applied Soft Computing*, 12(11):3332–3342, 2012.

[17] 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. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[18] H. Ze, 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*, pages 7962–7966. IEEE, 2013.

[19] H. Lu, S. King, and O. Watts. Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis. In *8th ISCA Workshop on Speech Synthesis*, pages 281–285, Barcelona, Spain, August 2013.

[20] Z. H. Ling, L. Deng, and D. Yu. Modeling spectral envelopes using restricted boltzmann machines and deep belief networks for statistical parametric speech synthesis. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21 (10):2129–2139, 2013.

[21] D. Erhan, Y. Bengio, A. Courville, P. A. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning? *The Journal of Machine Learning Research*, 11: 625–660, 2010.

[22] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786): 504–507, 2006.

[23] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010.

[24] Speech signal processing toolkit (sptk). URL `http://sp-tk.sourceforge.net/`.

[25] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 833–840, 2011.

[26] T. Dutoit, A. Holzapfel, M. Jottrand, A. Moinet, J. Perez, and Y. Stylianou. Towards a voice conversion system based on frame selection. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–513. IEEE, 2007.

[27] I. J. Goodfellow, D. Warde-Farley, P. Lamblin, V. Dumoulin, M. Mirza, R. Pascanu, J. Bergstra, F. Bastien, and Y. Bengio. Pylearn2: a machine learning research library. *arXiv preprint arXiv:1308.4214*, 2013.

[28] M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's mechanical turk — a new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1):3–5, January 2011.

[29] A. Kain. *High Resolution Voice Transformation*. PhD thesis, OGI School of Science & Engineering at Oregon Health & Science University, 2001.

[30] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.