INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Whether To Pretrain DNN or Not?: An Empirical Analysis for Voice Conversion

Nirmesh J. Shah[1], *Hardik B. Sailor*[2], *Hemant A. Patil*[1]

[1]Speech Research Lab, DA-IICT, Gandhinagar, India
[2]Speech and Hearing Research Group, University of Sheffield, UK
{nirmesh88_shah, hemant_patil}@daiict.ac.in, h.sailor@sheffield.ac.uk

Abstract

Recently, Deep Neural Network (DNN)-based Voice Conversion (VC) techniques have become popular in the VC literature. These techniques suffer from the issue of overfitting due to less amount of available training data from a target speaker. To alleviate this, pre-training is used for better initialization of the DNN parameters, which leads to faster convergence of parameters. Greedy layerwise pre-training of the stacked Restricted Boltzmann Machine (RBM) or the stacked De-noising AutoEncoder (DAE) is used with extra available speaker-pairs‘ data. This pre-training is time-consuming and requires a separate network to learn the parameters of the network. In this work, we propose to analyze the DNN training strategies for the VC task, specifically with and without pre-training. In particular, we investigate whether an extra pre-training step could be avoided by using recent advances in deep learning. The VC experiments were performed on two VC Challenge (VCC) databases 2016 and 2018. Objective and subjective tests show that DNN trained with Adam optimization and Exponential Linear Unit (ELU) performed comparable or better than the pre-trained DNN without compromising on speech quality and speaker similarity of the converted voices.

Index Terms: DNN, Voice Conversion, Dropout, Adam Optimization, Exponential Linear Unit (ELU).

1. Introduction

Voice Conversion (VC) converts the perceived speaker identity in a given speech signal from a source to a particular target speaker [1, 2]. Among various available VC techniques, Gaussian Mixture Model (GMM)-based methods have been considered state-of-the-art method since the last two decades [2]. Recently, Neural Network (NN)-based architectures, such as Artifical Neural Network (ANN) [3], Deep Neural Network (DNN) [4–7], Conditional Restricted Boltzmann Machine (CRBM) [8], Recurrent Neural Network (RNN) [9–11], Adversarial Network [12–17] have become more popular in the VC literature. These complex architectures of DNN require a huge amount of training data for their better performance. If the degrees of freedom, (i.e., the number of parameters to be learned) is more compared to the amount of available training data then the issue of overfitting may occur [18]. Most of the practical applications of VC generally face the issue of availability of less training data from the target speaker and hence, it leads to overfitting in DNN.

To alleviate this issue, pre-training is very popular [19, 20]. It was shown that pre-training is all the more helpful than the random initializations of the network parameters for DNN with a large number of layers [19]. Furthermore, it was shown that the pre-training works as a kind of regularization that resembles manifold learning [21]. In particular, Deep Belief Network (DBN) or the stacked De-noising AutoEncoder (DAE) architectures are being used to pre-train the DNN in the VC literature [4, 5, 22, 23]. This pre-training helps in achieving better initialization and results into the faster convergence of the networks. However, it requires separate training of the network from the different speaker-pairs‘ training data, which is time-consuming and costly. Recently, the idea of pre-training has become obsolete in most of the research area with proper activation function and regularization techniques [24–26]. In particular, it has been shown empirically that DNN with the Rectified Linear Unit (ReLU) [27] activation can be trained successfully with random initialization and also converges fast [24]. However, recent DNN-based methods still use the pre-training in VC due to overfitting and poor convergence issues during DNN training [6, 9].

In this study, we present an empirical analysis of strategies for training DNN for the VC task. Our study arises from recent progress in deep learning to train the DNN. We seek to understand that whether we really need to pretrain a DNN for the VC task. We have shown a comparison between pretraining a DNN before an actual VC task or directly training DNN for VC task. We have also used popular regularization and optimization techniques as well as recently proposed activation functions. The VC experiments were performed using the publicly available first and second Voice Conversion Challenge (VCC) databases [28, 29].

Our contributions:

- Empirical analysis with recent DNN strategies to overcome the need of pre-training in DNN-based VC.
- We applied recent variants of the Rectifier Linear Units (ReLU) [27], such as Leaky ReLU (i.e., LReLU) [30] and Exponential Linear Unit (i.e., ELU) [31] to get better and fast convergence.
- We also presented the impact of Xavier initialization over the random initialization to get better convergence.
- Motivated from [32], we propose to use dropout to overcome the overfitting issue in VC.
- Presented analysis w.r.t. different optimization strategies, such as Stochastic Gradient Descent (SGD) and Adam optimization in the context of pretraining of DNN.
- Detailed objective and subjective analysis is presented on the VCC 2016 and VCC 2018 databases.

2. DNN-based VC

The relation between the spectral feature vectors $\mathbf{X}$, and $\mathbf{Y}$ (from the source, and target speakers, respectively) are obtained using the DNN, which consists of $K > 2$ multiple layers, where

http://dx.doi.org/10.21437/Interspeech.2019-2608

K is the total number of layers. Here, each layer performs either nonlinear or linear transformation. The transformation at i^{th} layer is given by [26]:

$$\mathbf{h}_{i+1} = f(\mathbf{W}_i^T \mathbf{h}_i + \mathbf{b}_i), \tag{1}$$

where $\mathbf{h}_i$, $\mathbf{h}_{i+1}$, $\mathbf{W}_i$, $\mathbf{b}_i$ are called as input, output, weights and bias, respectively, of the i^{th} layer, and f is an activation function (such as, tangent hyperbolic, sigmoid, ReLU linear units) or linear. $\mathbf{h}_1 = \mathbf{X}$ and $\mathbf{h}_{K+1} = \mathbf{Y}$ are the input and output layers of DNN. Due to the increased number of layers, DNN can capture the more complex relationship between the source and target speakers' spectral features. The Stochastic Gradient Descent (SGD) algorithm is used to train the weights and biases of the DNN such that Mean Square Error (MSE), i.e., $E = ||\mathbf{Y} - \hat{\mathbf{Y}}||^2$ is minimized. Here, $\hat{\mathbf{Y}}$ is the predicted output. In the next section, we will briefly discuss techniques for training DNN along with recent proposals for DNN parameterization.

3. Strategies for Training DNN for VC

3.1. Pre-training of DNN

The random initialization of weights and biases of the DNN results in poor convergence, i.e., the likelihood will be stuck into local minima [33]. One of the possible solutions for faster and better convergence for the training of DNN is to set initial parameters via pre-training of the network [19]. In particular, we have used a DAE that is created by stacking of layers of the autoencoders for pre-training of DNN [34]. The baseline DNN consists of encoding layers of DAE, followed by shallow ANN and decoding layers of DAE [5].

3.2. Regularization

Sometimes during the DNN training, weights of the neighboring neurons become more dependent on the current neuron's weight, and this dependency is called complex co-adaptation [32]. Hence, if neurons are randomly dropped out then the neighboring neurons have to step in and make accurate predictions for the missing neurons, which will make our network to generalize itself very well, and also make it less sensitive to overfitting. Dropout is one of the most simple yet very effective ways of preventing overfitting in DNN [32]. We used dropout as a regularization in our DNN training. We have taken dropout probability of 0.3 as recently suggested in the area of speech recognition [35, 36]. Here, dropout is not applied at input and the output layers. The term dropout refers to randomly dropping out neurons with probability p. Applying a dropout to DNN can be considered as a multiplying neural network activation with a binary mask (also known as the *dropout mask*). The dropout mask is created using the random variables drawn from the Bernoulli distribution, i.e., $\mathbf{m}_k = Bernoulli(p)$.

$$\mathbf{m} = Bernoulli(p), \tag{2}$$

where $P(m = 1) = 1 - p$ and $P(m = 0) = p$. Hence, the output at the i^{th} layer is given by:

$$\mathbf{h}_{i+1} = \mathbf{m} \odot f(\mathbf{W}_i^T \mathbf{h}_i + \mathbf{b}_i). \tag{3}$$

Hence, dropping out a neuron in DNN with p probability means that a neuron is dropped out and its output is set to zero irrespective of whatever the input is given. On the other hand, it will keep the neurons with $1 - p$ probability in the network. Once the neuron is dropped out, it will not be able to contribute in forward and backward pass of the backpropagation. Every time a neuron is dropped out, it is like training a new DNN and hence, dropout can be thought as the average result for the entire ensemble of DNNs than a single DNN [32].

3.3. Choice of Activation Functions

Early DNN used sigmoid or tanh nonlinear activation until the ReLU, and its recent variants were proposed [26, 37]. It has been empirically shown that with the ReLU activation function, DNNs can be trained without the need of pre-training in other areas of speech processing apart from the VC [24]. Hence, we propose to use ReLU and its recent variants, such as LReLU and ELU for avoiding the pre-training in the area of VC. The activation functions are defined in Table 1. Here, α (also called as leakage parameter) controls the slope of negative part.

Table 1: *Definition of Activation Functions. After [27, 30, 31]*

	ReLU	LReLU	ELU
$f(x)$ for $x \geq 0$	x	x	x
$f(x)$ for $x < 0$	0	$\alpha \cdot x$	$\alpha \cdot (e^x - 1)$

Figure 1 shows the three piecewise linear activation functions. Here, LReLU was plotted by taking $\alpha = 0.1$, and for ELU, $\alpha = 1$ is taken as suggested in [30], [31], respectively. The key advantage of the ReLU, LReLU, and ELU is that they do not face gradient vanishing problems that is faced by the sigmoid and tanh [31]. Furthermore, computations of these activations are simpler which results into speed up in the training and faster convergence. In addition, they generalize the DNN better, i.e., they can predict the values more accurately for unseen data. Moreover, sigmoid activation is easier to saturate and hence, derivative of the input is almost zero once the sigmoid reaches to the any side of the plateau [26]. However, ReLU saturates only when the input is less than zero. In addition, LReLU reduces this saturation regions due to its non-zero behavior with the negative input. Furthermore, ELU also saturate with the negative value in presence of smaller input, which helps in decreasing forward propagation variations [31]. Recently, effectiveness of ELU over ReLU and LReLU was shown in areas of image processing in terms of faster convergence, and better generalization of networks [31].

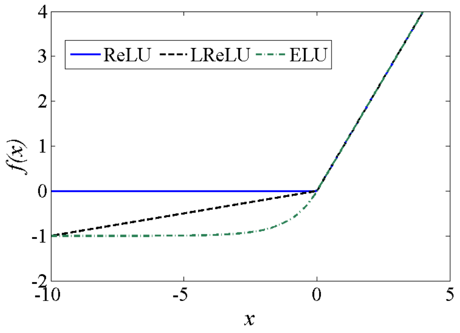

Figure 1: *Piecewise linear activation functions where input at neuron $x \in \Re$, and $f(x)$ is the activation function.*

3.4. Optimization

One of the key challenges of the SGD was that it requires a proper selection of learning rate. For example, too small value of learning rate will lead to the slower convergence, and higher learning rate can lead to miss the true convergence. In addition, the SGD applies the same learning rate to all the parameters. The Adam optimizer computes an individual adaptive learning rates for different parameters from the estimates of first and second moments of the gradients [38]. The name of Adam is derived from the adaptive moment estimation [38]. The Adam optimization has several advantages, such as the magnitudes of the parameter updates are invariant to rescaling of the gradient, and its step sizes are approximately bounded by the step size of hyperparameters. It also does not require a stationary objective, and works well with the sparse gradients, and it naturally performs a form of step size annealing. Due to the use of bias correction along with the first, and second-order moments of the gradient terms, the Adam optimization was shown to perform better than the SGD-based methods [38]. Recently, its convergence characteristics were also discussed [39].

3.5. Xavier Initialization

Proper initialization of the random weights play a key role in the training of the DNN [26]. For example, variance of the input start diminishing as it passes through each layer, if the weights are small. Hence, the inputs will not be useful during the training. Similarly, the variance of input data start increasing as it passes through each layer, if the weights are large. Hence, the inputs will explode, and will not be useful either. To tackle the issues of initializing a DNN with an arbitrary random weights, Xavier initialization technique was proposed [33]. It ensures that the variance of the weights remain same as it passes through each layer. This is achieved by initializing the weights from Gaussian distribution with zero mean and variance of $1/N$, where N is the number of input neurons [33]. In this paper, we also compare all the results w.r.t. the random and Xavier initialization techniques.

4. Experimental Results

4.1. Experimental Setup

In this paper, both VC Challenge (VCC) 2016 and 2018 databases have been used to build VC systems [28, 29]. *25*-D Mel Cepstral Coefficients (MCCs) (including the 0^{th} coefficient) and *1*-D F_0 for each frame (having *25* ms frame duration, and *5* ms frame shift) have been extracted. We have built the VC systems for all *25* and *16* speaker-pairs given in the VCC 2016 and VCC 2018 databases, respectively.

Table 2: *Descriptions of VC systems*

System	Pre-training	Opt.	Activation	Dropout
A (Baseline [5])	✓	SGD	Sigmoid	×
B	×	SGD	Sigmoid	×
C	×	SGD	ReLU	×
D	×	SGD	LReLU	×
E	×	SGD	ELU	×
F	×	SGD	Sigmoid	✓
G	×	Adam	Sigmoid	×
H	×	Adam	Sigmoid	✓
I	×	Adam	ReLU	✓
J	×	Adam	LReLU	✓
K	×	Adam	ELU	✓
L	✓	Adam	Sigmoid	✓
M	✓	Adam	ELU	✓
N	✓	SGD	ELU	✓

Opt.: Optimization, Here, ✓ indicates technique is used; × indicates technique is not used in this paper

The Dynamic Time Warping (DTW) algorithm was applied for the alignment task. The number of training utterances have been varied from, $n = 10, 20, 40, 100, \text{and} 150$. Four speakers' data from the CMU-ARCTIC database has been taken for the pre-training. We employ exactly the same architecture given in [5] for our baseline DNN system. We employ different optimization algorithm, nonlinear activation function and the dropout techniques in number of combinations w.r.t. the baseline VC systems. In this paper, we used $\alpha = 0.01$ for LReLU, and $\alpha = 1$ for ELU nonlinear activation function. We used Adam optimization with the $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The learning rate and the number of epochs were chosen from [5]. Mean-variance (MV) transformation is used for the F_0 (i.e., fundamental frequency) transformation. The AHOCODER is used for the analysis-synthesis [40]. The description of the developed VC systems is given in Table 1.

4.2. Objective Evaluation

For objective evaluation, we have selected the state-of-the-art Mel Cepstral Distortion (MCD) measure [2]. Figure 2 shows the average MCD for all the systems developed using 25 speaker-pairs along with 95 % confidence interval. It can be seen that the system A (i.e., baseline with pre-training) is having lower MCD value than the system B (i.e., baseline without pre-training). This clearly indicates that for a given baseline architecture [5], the pre-training is indeed helping to achieve lower MCD value. The MCD for system B is further reduced with the use of advanced activation functions (as shown for systems C, D, and E). Furthermore, systems G to K (i.e., systems with Adam and dropout and no pre-training) are able to perform equal or better compared to the baseline. This also shows the significance of Adam optimization over SGD. Moreover, it can also be observed that with SGD and Adam system with ELU is performing better compared to all other activations.

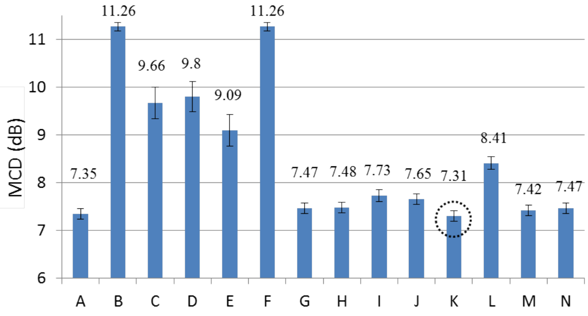

Figure 2: *The MCD analysis for various VC systems developed on VCC 2016 database. Dotted circle indicates relatively better performing proposed VC system.*

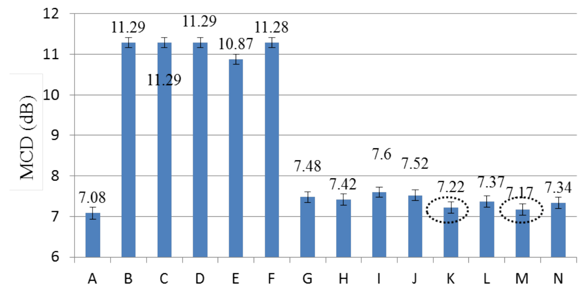

Figure 3: *The MCD analysis for various VC systems developed on VCC 2018 database. Dotted circle indicates relatively better performing proposed VC system.*

To further investigate the effectiveness of pre-training, we also develop the systems L to N. In the case of pre-trained network, further reduction in the MCD for the system with Adam over SGD can be clearly seen in Figure 2 w.r.t. the activation function ELU and the dropout. Furthermore, reduction in the MCD can be clearly seen for the system M w.r.t. the system L, which is solely due to the ELU activation function. Overall, the system K is performing better w.r.t. the baseline and other pre-trained network. Hence, the proposed network can be used to overcome the need of pre-training in the VC. Similar observations can be made for the VCC 2018 database from Figure 3.

Figure 3 shows the average MCD for all the systems developed on the Hub task of VCC 2018 using 16 speaker-pairs along with 95 % confidence interval.Here, average MCD is calculated for all the VC systems developed on the 25 and 16 speaker-pairs in VCC 2016 and VCC 2018 databases, respectively. It can be seen that the system K without pre-training is performing relatively the *best* among all the systems. Furthermore, it can be seen that system K is consistently performing better across the number of training utterances. In particular, system K with only ten utterances in training (where the possibilities of overfitting is higher) is also performing better than the baseline system A. This may be due to the fact that the dropout prevents overfitting in the DNN by means of stochastic regularization.

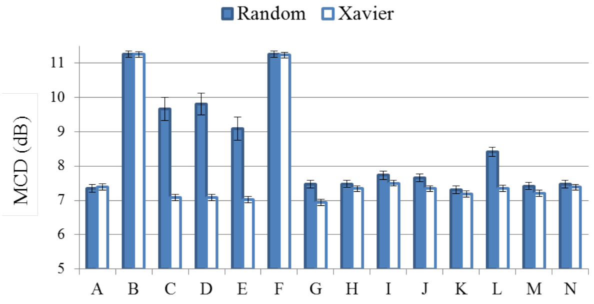

Figure 4: *Comparison of random initialization vs. Xavier initialization on the VCC 2016 database.*

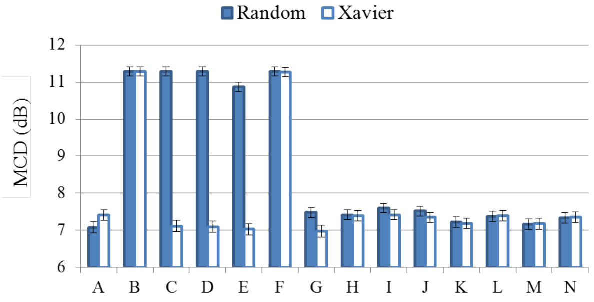

Figure 5: *Comparison of random initialization vs. Xavier initialization on the VCC 2018 database.*

Figure 4 and Figure 5 shows the comparison of the Xavier initialization w.r.t. the random initialization on the VCC 2016 and VCC 2018 databases, respectively. The effectiveness of the Xavier initialization can be seen on both the databases. Hence, the Xavier initialization can also be useful to overcome the need of pre-training in addition to the other proposed modifications. We can clearly see that there is a significant improvement in the performance of System C, D, and E with the Xavier initialization on both the database. It is possibly due to the fact that fixed variance in the Xavier initialization at each layers helping the inputs to not getting explode and hence, resulted in the better performance [33].

4.3. Subjective Evaluation

To measure both the speech quality and the Speaker Similarity (SS) of converted voice, Mean Opinion Score (MOS) test have been taken. The subjective tests were taken from the 14 subjects (2 female and 12 male with no known hearing impairments, and with the age variations between 21 to 30 years) from total 252 samples. In the MOS test, subjects were asked to evaluate the randomly played utterance for the speech quality and SS. For speech quality, subjects were asked to rate the converted voice on the scale of 1 (i.e., very bad) to 5 (i.e., very good). Similarly, for the SS, subjects were asked to rate the converted voice in terms of SS on the scale of 1 (not at all the target speaker) to 5 (exactly same as the target speaker). The result of the MOS test, for the VCC 2016 and VCC 2018, is shown in the Figure 6, and Figure 7 along with 95 % confidence intervals,respectively. It is clearly visible from the Figure 6 and Figure 7 that the proposed system K without pre-training is performing comparable and slightly better w.r.t. the baseline system with pre-training in the MOS tests for speech quality and speaker similarity, respectively. This indicates that the need of pre-training for the DNN can be reasonably avoided by using our proposed system.

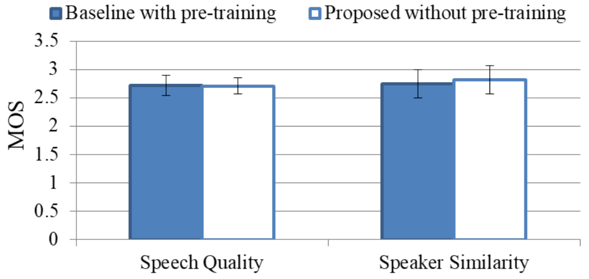

Figure 6: *The MOS analysis along with 95 % confidence interval on VCC 2016 database.*

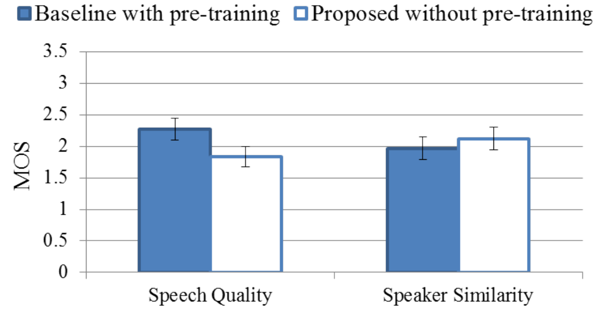

Figure 7: *The MOS analysis along with 95 % confidence interval on VCC 2018 database.*

5. Summary and Conclusions

The state-of-the-art DNN-based VC techniques require greedy layerwise pre-training of the network for better initialization and faster convergence of the DNN. In this work, we proposed to use the DNN with dropout and the ELU activation function to overcome pre-training in DNN-based VC systems. In addition, we proposed to use the Adam optimization-based techniques along with Xavier initialization. We found that the proposed DNN performs slightly better and/or comparable w.r.t. the DNN with the pre-training on both the VCC 2016 and VCC 2018 databases. The subjective evaluations also justify that the proposed model is able to overcome the need for pre-training in DNN-based VC. In future, we would like to perform analysis of the convergence speed for the DNN training.

6. References

[1] Y. Stylianou, "Voice transformation: A survey," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei,Taiwan, 2009, pp. 3585–3588.

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

[3] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, "Voice conversion using artificial neural networks," in *ICASSP*, Taipei, Taiwan, 2009, pp. 3893–3896.

[4] L. H. Chen, Z. H. Ling, L. J. Liu, and L. R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1859–1872, 2014.

[5] S. H. Mohammadi and A. Kain, "Voice conversion using deep neural networks with speaker-independent pre-training," in *IEEE Spoken Language Technology Workshop (SLT)*, Nevada, USA, 2014, pp. 19–23.

[6] S. H. Mohammadi and A.Kain, "Semi-supervised training of a voice conversion mapping function using a joint-autoencoder," in *INTERSPEECH*, Dresden, Germany, 2015, pp. 1–5.

[7] N. J. Shah and H. A. Patil, "Novel outliers removal approach for parallel voice conversion," *Computer Speech and Language, Elsevier*, vol. 58, no. 11, pp. 127–152, 2019.

[8] Z. Wu, E. S. Chng, and H. Li, "Conditional restricted Boltzmann machine for voice conversion," in *ChinaSIP*, Beijing, China, 2013, pp. 104–108.

[9] J. Wu, D. Huang, L. Xie, and H. Li, "Denoising recurrent neural network for deep bidirectional LSTM based voice conversion," in *INTERSPEECH*, Sweden, 2017, pp. 3379–3383.

[10] C. Zhou, M. Horgan, V. Kumar, C. Vasco, and D. Darcy, "Voice conversion with conditional SampleRNN," in *INTERSPEECH*, Hyderabad, India, 2018, pp. 1973–1977.

[11] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *ICASSP*, Brisbane, Queensland, Australia, 2015, pp. 4869–4873.

[12] F. Fang *et al.*, "High quality nonparallel voice conversion based on cycle-consistent adversarial network," in *ICASSP*, Calgary, Canada, 2018, pp. 5279–5283.

[13] Y. Gao, R. Singh, and B. Raj, "Voice impersonation using generative adversarial networks," in *ICASSP*, Calgary, Canada, 2018, pp. 2506–2510.

[14] S. Seshadri, L. Juvela, J. Yamagishi, O. Rasanen, and P. Alku, "Cycle-consistent adversarial networks for non-parallel vocal effort based speaking style conversion," in *ICASSP*, Brighton, UK, 2019.

[15] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "CYCLEGAN-VC2: Improved cyclegan-based non-parallel voice conversion," in *ICASSP*, Brighton, UK, 2019.

[16] N. J. Shah, M. C. Madhavi, and H. A. Patil, "Unsupervised vocal tract length warped posterior features for non-parallel voice conversion," in *INTERSPEECH*, Hyderabad, India, 2018, pp. 1968–1972.

[17] N. J. Shah, S. R., N. Shah, and H. A. Patil, "Novel unsupervised sorted GMM posteriorgram for DNN and GAN-based voice conversion framework," in *APSIPA Annual Summit and Conference*. Hawaii, USA: IEEE, 2018, pp. 1776–1781.

[18] E. Helander, H. Silén, T. Virtanen, and M. Gabbouj, "Voice conversion using dynamic kernel partial least squares regression," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 3, pp. 806–817, 2012.

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

[20] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. on Audio, Speech, and Lang. Proc.*, vol. 20, no. 1, pp. 30–42, 2012.

[21] Y. Furusho, T. Kubo, and K. Ikeda, "Roles of pre-training in deep neural networks from information theoretical perspective," *Neurocomputing*, vol. 248, pp. 76–79, 2017.

[22] S. H. Mohammadi and A.Kain, "A voice conversion mapping function based on a stacked joint-autoencoder." in *INTERSPEECH*, San Fransisco, USA, 2016, pp. 1647–1651.

[23] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion using RNN pre-trained by recurrent temporal restricted Boltzmann machines," *IEEE/ACM Trans. on Audio, Speech and Lang. Proces.*, vol. 23, no. 3, pp. 580–587, 2015.

[24] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *AISTATS*, FL, USA, 2011, pp. 315–323.

[25] M. D. Zeiler, M. Ranzato *et al.*, "On rectified linear units for speech processing," in *ICASSP*, Vancouver, BC, Canada, 2013, pp. 3517–3521.

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, First Edition, 2016.

[27] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *ICML*, United States, 2010, pp. 807–814.

[28] T. Toda, L.-H. Chen, D. Saito, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi, "The Voice Conversion Challenge 2016," in *INTERSPEECH*, San Fransisco, USA, 2016, pp. 1–5.

[29] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," in *The Speaker and Language Recognition Workshop Odyssey*, Les Sables d'Olonne, France, 2018, pp. 195–202.

[30] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML*, vol. 30, no. 1, Atlanta, USA, 2013.

[31] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *ICLR*, San Juan, Puerto Rico, 2016, pp. 1–14.

[32] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[33] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, Sardinia, Italy, 2010, pp. 249–256.

[34] 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," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[35] G. Cheng, V. Peddinti, D. Povey, V. Manohar, S. Khudanpur, and Y. Yan, "An exploration of dropout with LSTMs," in *INTERSPEECH*, Stockholm, Sweden, 2017, pp. 1586–1590.

[36] H. B. Sailor and H. A. Patil, "Auditory feature representation using convolutional restricted Boltzmann machine and Teager energy operator for speech recognition," *JASA-EL*, vol. 141, no. 6, pp. 500–506, 2017.

[37] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Advances in neural information processing systems*, California, USA, 2017, pp. 971–980.

[38] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, San Diego, USA, 2015, pp. 1–15.

[39] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *ICLR*, Vancouver, CANADA, 2018, pp. 1–23.

[40] D. Erro, I. Sainz, E. Navas, and I. Hernáez, "Improved HNM-based vocoder for statistical synthesizers." in *INTERSPEECH*, Florence, Italy, 2011, pp. 1809–1812.