Interspeech 2018
2-6 September 2018, Hyderabad

Artificial Bandwidth Extension with Memory Inclusion using Semi-supervised Stacked Auto-encoders

Pramod Bachhav, Massimiliano Todisco and Nicholas Evans

EURECOM, Sophia Antipolis, France
{bachhav,todisco,evans}@eurecom.fr

Abstract

Artificial bandwidth extension (ABE) algorithms have been developed to improve quality when wideband devices receive speech signals from narrowband devices or infrastructure. The utilisation of contextual information in the form of dynamic features or explicit memory captured from neighbouring frames is common to ABE research, however the use of additional cues augments complexity and can introduce latency. Previous work shows that unsupervised, linear dimensionality reduction techniques help to reduce complexity. This paper reports a semi-supervised, non-linear approach to dimensionality reduction using a stacked auto-encoder. In further contrast to previous work, it operates on raw spectra from which a low dimensional narrowband representation is learned in a data-driven manner. Three different objective speech quality measures show that the new features can be used with a standard regression model to improve ABE performance. Improvements in the mutual information between learned features and missing higher frequency components are also observed whereas improvements in speech quality are corroborated by informal listening tests.

Index Terms: artificial bandwidth extension, auto-encoder, dimensionality reduction, mutual information

1. Introduction

While legacy narrowband (NB) telephony infrastructure is limited to a bandwidth of 0.3-3.4kHz, today's wideband (WB) technology supports improved speech quality using a bandwidth extending from 50Hz-7kHz. Artificial bandwidth extension (ABE) algorithms have been developed to improve speech quality when WB devices are used with NB devices or infrastructure. Using the correlation between the two [1], ABE is used to estimate missing highband (HB) frequency components above 3.4kHz from available NB components, typically using a regression model learned from WB training data.

ABE approaches based on source-filter modelling estimate separate spectral envelope and excitation components [2, 3]. Other approaches operate directly on complex short-term spectral estimates derived, e.g., using the Fourier transform (STFT) [4, 5] or constant-Q transform [6]. Complementary to short-term spectral estimates, is some form of contextual information, or *memory* which can be harnessed to improve the reliability of HB component estimation. Some specific back-end regression models, e.g., Hidden Markov models (HMMs) [7, 8] and deep neural networks (DNNs) [9–11], capture memory in the form of temporal information. Some DNN solutions, e.g. [4, 12, 13], capture memory in the front-end instead, e.g., via delta features or static features from neighbouring frames. Following an investigation of front-end feature extraction for ABE [14], the work in [15–17] investigates the merit of memory inclusion through information theoretic analysis. This body of work demonstrates the benefit of memory inclusion through delta features under the constraint of fixed dimensionality. However, the inclusion of memory necessitates the loss of informative higher-order static HB features in order to accommodate dynamic delta features. Our own work [18] analyses quantitatively the benefit of explicit memory inclusion in a fixed ABE solution. The work also addresses the latency and complexity problem. Complexity is managed using principal component analysis (PCA) in order to incorporate memory without increasing feature dimensionality; regression complexity is unaffected. An unsupervised, linear approach to dimensionality reduction, PCA aims only to produce a low dimensional representation which retains as much as possible the variation in the input representation. The hypothesis of the research presented in this paper is that supervised or semi-supervised and non-linear dimensionality reduction techniques offer potential to learn lower dimensional representations tailored specifically to ABE, thereby giving better performance.

Auto-encoders (AEs) are an increasingly popular approach to non-linear dimensionality reduction and have been applied widely to many speech processing tasks, e.g., phoneme/speech recognition [19–21] and speech synthesis [22]. Common to these examples is the use of AEs to learn so-called bottleneck features, namely compact feature representations tailored to pattern recognition and classification. This paper reports the use of AEs for non-linear dimensionality reduction in ABE and specifically the use of stacked (deep) AEs trained in a semi-supervised manner. The objectives are to (i) harness memory in a compact, low dimensional representation in order to improve the reliability of estimated HB components and (ii) to learn NB features directly from raw spectral coefficients instead of hand-crafted features. The merit of both contributions is assessed through objective assessment, an information theoretic approach and informal listening tests.

The remainder of this paper is organised as follows. Section 2 describes a baseline ABE algorithm. Section 3 shows how semi-supervised stacked AEs can be applied to improve its performance. Experimental work is described in Section 4, whereas results are presented in Section 5. Conclusions are presented in Section 6.

2. Baseline ABE system

Fig. 1 illustrates the baseline ABE system. It is identical to the source-filter model based approach presented in [18]. Since full details are available there, only a *brief* overview is provided here. The algorithm comprises three blocks: training, estimation and resynthesis.

Training operates using both NB and WB frame-blocked signals x_t and y_t respectively, where t is the time index. The NB component is parametrised with 10 log-Mel filter energy (logMFE) coefficients (X_t^{NB} – top line in training block). The HB component is instead parametrised via selective linear pre-

10.21437/Interspeech.2018-2213

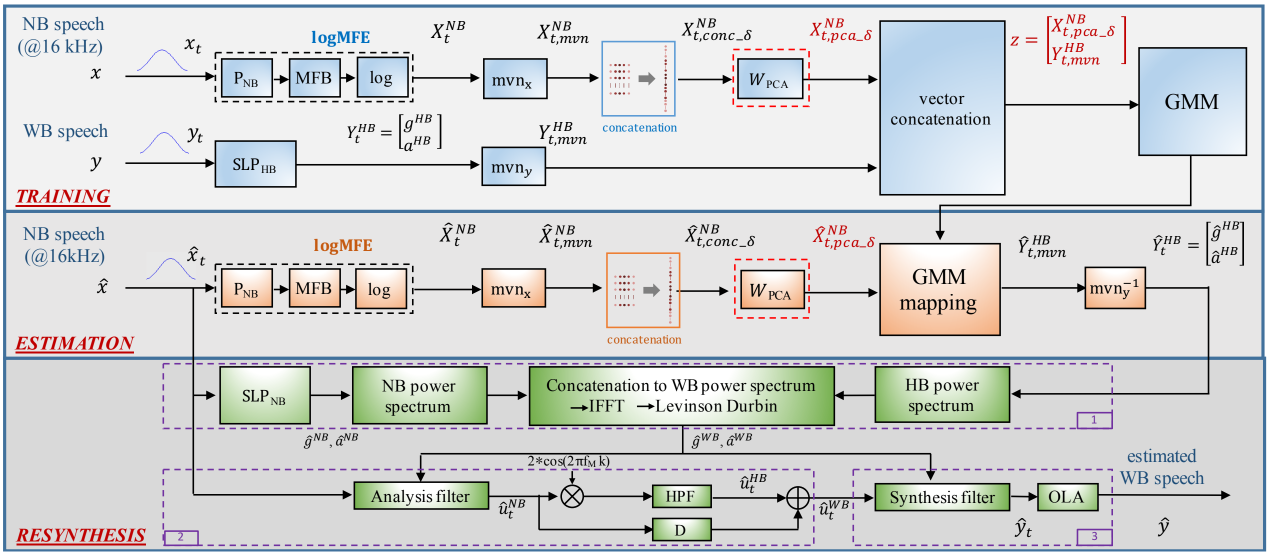

Figure 1: *A block diagram of the baseline ABE system with memory inclusion.*

diction (SLP) [23] giving 9 linear prediction (LP) coefficients and a gain parameter (Y_t^{HB} – bottom line in training block). Both NB and HB features are mean and variance normalised (mvn$_x$ and mvn$_y$) giving $X_{t,mvn}^{NB}$ and $Y_{t,mvn}^{HB}$. NB features at time t are concatenated with features extracted from δ neighbouring frames thus giving:

$$X_{t,conc\_\delta} \quad = \quad \left[X_{t-\delta,mvn}^{NB}, ..., X_{t,mvn}^{NB}, ..., X_{t+\delta,mvn}^{NB} \right]^T$$

In order to limit complexity, PCA is then applied to reduce $X_{t,conc\_\delta}$ to 10-dimensional features X_{t,pca_δ}^{NB}. The PCA matrix W_{PCA} is learned from training data and used unchanged in the estimation step. Finally, a 128-component, full-covariance Gaussian Mixture model (GMM) is learned from the training data using the concatenation $Z = [X_{t,pca\_\delta}^{NB}, Y_{t,mvn}^{HB}]^T$.

Estimation is applied to upsampled NB signals $\hat{x}$. They are treated according to the same NB processing and memory inclusion as in training to give 10-dimensional features $\hat{X}_{t,pca\_\delta}^{NB}$. A conventional regression model [2] defined by GMM parameters learned during training is then applied to estimate HB features $\hat{Y}_{t,mvn}^{HB}$. Using means and variances obtained from training, inverse mean and variance normalisation (mvn$_y^{-1}$) is then applied to estimate HB LP coefficients $\hat{a}^{HB}$ and gain $\hat{g}^{HB}$.

Resynthesis is performed according to the three distinct steps illustrated by the numbered blocks in Fig. 1. First (box 1), the missing WB power spectrum is estimated from the concatenation of NB and estimated HB power spectra for $\hat{x}_t$, defined by NB LP parameters $\hat{g}^{NB}$, $\hat{a}^{NB}$ and estimated HB parameters $\hat{g}^{HB}$, $\hat{a}^{HB}$. Estimated WB parameters $\hat{g}^{WB}$ and $\hat{a}^{WB}$ are then obtained from the WB power spectrum using the inverse fast Fourier transform (IFFT) followed by Levinson-Durbin recursion. Second (box 2), NB excitation $\hat{u}_t^{NB}$ is obtained using a LP analysis filter defined by $\hat{g}^{NB}$ and $\hat{a}^{NB}$. Spectral translation [3] followed by a high pass filter (HPF) is then applied to give HB excitation component $\hat{u}_t^{HB}$ which is then added to $\hat{u}_t^{NB}$ after appropriate delay D to give extended WB excitation $\hat{u}_t^{WB}$. Finally (box 3), $\hat{u}_t^{WB}$ is filtered using a synthesis filter defined by $\hat{g}^{WB}$ and $\hat{a}^{WB}$ in order to resynthesise speech frame $\hat{y}_t$. Overlap and add (OLA) then gives the extended WB speech $\hat{y}$.

3. ABE using semi-supervised stacked auto-encoders

The baseline ABE algorithm uses unsupervised, linear dimensionality reduction so that the complexity of the standard regression model learned in training and used in estimation, remains unchanged as a result of memory inclusion. The work presented in this paper seeks to improve ABE performance using a semi-supervised, non-linear dimensionality reduction technique using a stacked auto-encoder.

3.1. Stacked auto-encoders

An auto-encoder (AE) is an artificial neural network that is used widely for the learning of higher-level data representations. An AE consists of an encoder and a decoder. The encoder $f_\theta()$ maps an input vector x to a hidden representation y according to:

$$y = f_\theta(x) = s(Wx + b) \tag{1}$$

where $\theta = \{W, b\}$ is the parameter set of weight matrix W and bias vector b. The function s is a non-linear transformation. The encoder is followed by a decoder $g_{\theta'}()$ which aims to reconstruct the original input from the learned representation y according to:

$$z = g_{\theta'}(y) = s'(W'y + b') \tag{2}$$

where $\theta' = \{W', b'\}$ and s' is either a linear or a non-linear transformation depending on the nature of input x. For real-valued inputs, parameters $\{\theta, \theta'\}$ are optimised according to a mean squared error (MSE) objective loss function which reflects the difference between the input and the reconstructed output.

Deeper networks have inherently greater capacity to learn highly non-linear and complex functions [24]. The depth of an AE can be increased by stacking multiple layers of encoders and decoders, thereby forming a stacked auto-encoder (SAE). However, as the network increases, it becomes increasingly difficult for the network to find global minima [25].

In order to mitigate such problems, some form of pre-training is usually employed to initialise network weights. Popular solutions include pre-training using restricted boltzman

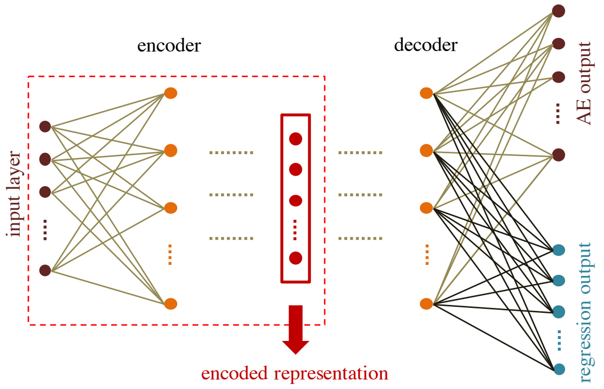

Figure 2: *A semi-supervised stacked auto-encoder.*

machines (RBMs) [25] and denoising AEs [26]. Layers are stacked after pre-training and subsequently fine-tuned. Other works have studied alternative means of network initialisation e.g. [27, 28].

3.2. Application to ABE

With a reconstruction-based objective loss function, SAEs can learn a simple mapping between the input and the reconstructed output, rather than a meaningful, high-level representation [26]. Additionally, being unsupervised, features extracted from the bottleneck layer of a conventional SAE are not expressly designed for classification or regression; they will likely be suboptimal in this respect. The partially-supervised pre-training of AEs was shown in [24] to be beneficial, especially for regression tasks.

Drawing upon this work, we have explored the semi-supervised training of SAEs in order to learn compact representations designed specifically for regression modelling and ABE. The resulting semi-supervised SAE (SSAE) architecture with 2 output layers is illustrted in Fig. 2. While one output layer is learned to reconstruct the input (AE output) as with a conventional SAE, the other output layer is learned to estimate the missing HB features (regression output). This is achieved though a joint objective loss function given by:

$$L_{total} = c * L_{reg} + (1 - c) * L_{ae}$$

where L_{reg} and L_{ae} are the objective loss functions for regression and AE outputs respectively and where $c \in [0, 1]$ weights the contribution of individual losses.

The SSAE architecture can also be used to estimate the HB components directly from the regression layer. A similar CNN based architecture designed to regularise the mapping of short i-vectors to long i-vectors for a speaker diarization task is reported in [29]. The focus here is different, i.e., to regularise/supervise dimensionality reduction so that it preserves information critical to ABE. This information is exploited by an otherwise standard regression model. In order to investigate the merit of the SSAE-based approach to dimensionality reduction, the weight matrix W_{PCA} in Fig. 1 (red boxes) is replaced by the SSAE *encoder* (red box in Fig. 2). Extracted low dimensional features are then mean and variance normalised. GMM training and estimation are performed in the same manner described in Section 2. Also reported in this paper is a variation on this approach whereby the low dimensional NB representation is derived directly from NB log power spectrum (LPS) coefficients instead of logMFE features. This is achieved quite simply by replacing logMFE features with LPS coefficients.

4. Experiments

Experiments are designed to compare the performance of the baseline ABE system using PCA dimensionality reduction M_{PCA_2} to that of the same system using SSAE dimensionality reduction M_{AE_2}. Systems M_{PCA_2} and M_{AE_2} use $\hat{X}^{NB}_{t,pca\_2}$ and $\hat{X}^{NB}_{t,ae\_2,mvn}$ features respectively. This section describes the databases used for ABE experiments, SSAE configuration details and metrics.

4.1. Database

The TIMIT dataset [30] was used for training and validation. ABE solutions were trained with the 3696 utterances from the training set and 1152 utterance from the test set (excluding core test subset) using parallel WB and NB speech signals processed according to the steps described in [6]. The TIMIT core test subset (192 utterances) was used for validation and for optimisation of network parameters. Motivated by the approach to analysis presented in [31], the acoustically different TSP database [32] comprising 1378 utterances was used for testing. TSP data was downsampled to 16kHz and similarly pre-processed to obtain parallel WB and NB data.

4.2. SSAE training and configuration

The SSAE was implemented with the Keras toolkit [33]. Consistent with prior work [18], features $X_{t,conc\_2}$ at time t (obtained from the concatenation of 2 preceding and 2 proceeding frames) are fed to the input of the SSAE. Whereas the AE output is the same as the input, the regression output is set to HB features $Y^{HB}_{t,mvn}$. So as to improve the rate of convergence to global minima, the SSAE is initialised according to the approach described in [28]. Optimisation is performed according to the procedure described in [34] with a standard learning rate of 0.001, a momentum of 0.9 and with a MSE criterion.

We investigated two 6-layer symmetric SSAE structures with different numbers of units in hidden layers: 1) 512, 256, 10, 256, 512 (Arch-1); 2) 1024, 512, 10, 512, 1024 (Arch-2). Output layers consists of 50 (AE) and 10 (regression) units. Hidden layers have tanh or ReLU activation units whereas output layers have linear activation units. Dropout (*dr*) [35] and batch-normalisation [36] techniques are investigated as means of discouraging over-fitting. The learning rate is reduced by half in the case that the validation loss increases between 2 consecutive epochs. Regression and AE loss weights were both set to *c*=0.5. Networks are trained for 30 epochs.

4.3. Metrics

Performance is reported in terms of objective assessments. Objective spectral distortion measures include: the root mean square log-spectral distortion (RMS-LSD); the so-called COSH measure (symmetric version of the Ikatura-Saito distortion) [37] calculated for a frequency range 3.4-8kHz, and a WB extension to the perceptual analysis of speech quality algorithm [38]. The latter gives objective estimates of mean opinion scores (MOS-LQO$_{\text{WB}}$). The correlation of the SSAE and PCA representations with the HB features is measured via mutual information (MI) [14].

5. Results

Validation performance in terms of MSE for the two different architectures and four different combinations of dropout (dr) and batch-normalisation performed either after (bn-a) or before (bn-b) activation is shown in Tab.1. Dropout layer is used before all hidden layers. Relatively low values of MSE are achieved without dropout or batch normalisation (configuration A), although performance is poor for Arch-2 with ReLU activation. The use of dropout without batch-normalisation (configuration D) results in poorly regularized networks, especially for ReLU activation. Similar observations are reported in [31]. The use of either form of batch-normalisation without dropout gives consistently low values of MSE, with the best results being obtained with a bn-b configuration (C). All results reported in the remainder of this paper relate to this configuration.

Table 1: *Average MSE for different SSAE configurations including architecture 1 and 2 with either ReLU or tanh activation functions, with and without dropout (dr) and batch normalisation (bn) either after (a) or before (b) activation. dr value represents fraction of random hidden units being set to 0. Results are illustrated for assessments using the validation dataset.*

	dr	bn	Arch-1		Arch-2	
			ReLU	tanh	ReLU	tanh
A	-	-	0.474	0.461	1.012	0.467
B	-	a	0.460	0.461	0.461	0.467
C	-	b	**0.459**	**0.459**	**0.460**	**0.460**
D	0.2	-	1.012	0.504	1.012	0.509

Objective performance measures obtained from the testing set and for both the baseline M_{PCA_2} and SSAE-based approach M_{AE_2} to ABE are illustrated Table 2. With only one exception, spectral distortion metrics results show lower values for SSAE than for the baseline. MOS-LQO$_{\text{WB}}$ scores for SSAE systems are consistently higher. The Arch-2 SSAE system with a tanh activation performs best. Unfortunately, despite convincing improvements in objective performance metrics, informal listening tests showed little discernible differences between the quality of speech signal produced by the baseline and SSAE systems.

Table 2: *Objective performance metric results. RMS-LSD and d*$_{COSH}$ *are mean spectral distortion measures in dB (lower values indicate better performance) whereas MOS-LQO*$_{WB}$ *values reflect quality (higher values indicate better performance).*

	Arch-1C		Arch-2C		Baseline
	ReLU	tanh	ReLU	tanh	
$d_{\text{RMS-LSD}}$	7.28	7.12	7.38	**7.11**	7.34
d_{COSH}	1.48	1.44	1.49	**1.43**	1.52
MOS-LQO$_{\text{WB}}$	2.99	3.06	2.99	**3.07**	2.96

Objective performance measures for the two best performing SSAE configurations, Arch-1C and Arch-2C both with tanh activations, trained using LPS inputs instead of logMFE features are illustrated in Table 3. Distortion measures are consistently lower, whereas MOS-LQO$_{\text{WB}}$ scores are consistently higher than results for all other SSAE-based systems. In contrast to findings for the SSAE systems that operate using logMFE features, informal listening test show discernible improvements to speech quality compared to speech produced using the baseline ABE system. Examples of bandwidth-extended speech produced by both baseline and SSAE systems operating on both logMFE and LPS inputs are available online[1].

Table 3: *Objective assessment results for SSAE using raw log power spectrum (LPS) inputs in place of log-Mel filter energy (logMFE).*

	$d_{\text{RMS-LSD}}$	d_{COSH}	MOS- LQO$_{\text{WB}}$
Arch-1C, tanh	6.90	1.37	3.16
Arch-2C, tanh	**6.88**	**1.34**	**3.17**

A final set of results aims to further validate the findings of both objective and informal listening tests. This is achieved by observing improvements to the mutual information (MI) between the learned NB representation and true HB representation measured using the testing set. A 128-component full-covariance GMM trained with joint vectors formed by learned NB and true HB features is used for the MI estimation as described in [18]. MI results presented in Table 4 show that the Arch-2C SSAE system with tanh activations trained using LPS inputs gives a relative increase in MI of 23% over the baseline system. This result corroborates the findings presented above, namely that semi-supervised techniques which operate on raw spectral inputs are capable of learning better representations that can be exploited to deliver improved ABE performance.

Table 4: *Mutual information assessment results.* $I(X;Y)$ *denotes the MI between features X and Y.*

$I(\hat{X}^{NB}_{pca\_2}; Y_t)$, Baseline	1.55
$I(\hat{X}^{NB}_{ae\_2}; Y_t)$, Arch-1C (logMFE)	1.69
$I(\hat{X}^{NB}_{ae\_2}; Y_t)$, Arch-2C (logMFE)	1.71
$I(\hat{X}^{NB}_{ae\_2}; Y_t)$, Arch-1C (LPS)	1.84
$I(\hat{X}^{NB}_{ae\_2}; Y_t)$, Arch-2C (LPS)	**1.90**

6. Conclusions

This paper presents a non-linear, semi-supervised approach to dimensionality reduction for artificial bandwidth extension. The ability of stacked auto-encoders to learn higher-level representation is exploited further to learn compact narrowband features directly from raw spectra. The merit of the approach is demonstrated with different objective metrics and is confirmed by the findings of informal listening tests. The usefulness of newly learned features is confirmed by information theoretic analysis. Features extracted from raw spectra in a data-driven manner can be used by a standard regression model without augmenting complexity. Exploiting potential spectral modelling transforms and their further optimisation to learn features for ABE should be our focus in future. Further work should also investigate the combination of semi-supervised auto-encoders with unsupervised or partially supervised pre-training methods. These may offer even greater potential to improve the quality of artificially bandwidth-extended speech.

[1] http://audio.eurecom.fr/content/media

7. References

[1] Y. Cheng, D. O'Shaughnessy, and P. Mermelstein, "Statistical recovery of wideband speech from narrowband speech," *IEEE Trans. on Speech and Audio Processing*, vol. 2, no. 4, pp. 544–548, 1994.

[2] K.-Y. Park and H. Kim, "Narrowband to wideband conversion of speech using GMM based transformation," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, 2000, pp. 1843–1846.

[3] P. Jax and P. Vary, "On artificial bandwidth extension of telephone speech," *Signal Processing*, vol. 83, no. 8, pp. 1707–1719, 2003.

[4] K. Li and C.-H. Lee, "A deep neural network approach to speech bandwidth expansion," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4395–4399.

[5] R. Peharz, G. Kapeller, P. Mowlaee, and F. Pernkopf, "Modeling speech with sum-product networks: Application to bandwidth extension," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2014, pp. 3699–3703.

[6] P. Bachhav, M. Todisco, M. Mossi, C. Beaugeant, and N. Evans, "Artificial bandwidth extension using the constant Q transform," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5550–5554.

[7] C. Yağli and E. Erzin, "Artificial bandwidth extension of spectral envelope with temporal clustering," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 5096–5099.

[8] I. Katsir, D. Malah, and I. Cohen, "Evaluation of a speech bandwidth extension algorithm based on vocal tract shape estimation," in *Proc. of Int. Workshop on Acoustic Signal Enhancement (IWAENC)*. VDE, 2012, pp. 1–4.

[9] Y. Wang, S. Zhao, D. Qu, and J. Kuang, "Using conditional restricted boltzmann machines for spectral envelope modeling in speech bandwidth extension," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016, pp. 5930–5934.

[10] Y. Gu, Z.-H. Ling, and L.-R. Dai, "Speech bandwidth extension using bottleneck features and deep recurrent neural networks." in *Proc. of INTERSPEECH*, 2016, pp. 297–301.

[11] Y. Wang, S. Zhao, J. Li, J. Kuang, and Q. Zhu, "Recurrent neural network for spectral mapping in speech bandwidth extension," in *Proc. of IEEE Global Conf. on Signal and Information Processing (GlobalSIP)*, 2016, pp. 242–246.

[12] B. Liu, J. Tao, Z. Wen, Y. Li, and D. Bukhari, "A novel method of artificial bandwidth extension using deep architecture," in *Sixteenth Annual Conf. of the Int. Speech Communication Association*, 2015.

[13] J. Abel, M. Strake, and T. Fingscheidt, "Artificial bandwidth extension using deep neural networks for spectral envelope estimation," in *Proc. of Int. Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2016, pp. 1–5.

[14] P. Jax and P. Vary, "Feature selection for improved bandwidth extension of speech signals," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2004, pp. I–697.

[15] A. Nour-Eldin, T. Shabestary, and P. Kabal, "The effect of memory inclusion on mutual information between speech frequency bands," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, 2006, pp. III–III.

[16] A. Nour-Eldin and P. Kabal, "Objective analysis of the effect of memory inclusion on bandwidth extension of narrowband speech," in *Proc. of INTERSPEECH*, 2007, pp. 2489–2492.

[17] ——, "Mel-frequency cepstral coefficient-based bandwidth extension of narrowband speech,," in *Proc. of INTERSPEECH*, 2008, pp. 53–56.

[18] P. Bachhav, M. Todisco, and N. Evans, "Exploiting explicit memory inclusion for artificial bandwidth extension," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5459–5463.

[19] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 3377–3381.

[20] T. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4153–4156.

[21] D. Yu and M. Seltzer, "Improved bottleneck features using pretrained deep neural networks," in *Twelfth Annual Conf. of the Int. Speech Communication Association*, 2011.

[22] S. Takaki and J. Yamagishi, "A deep auto-encoder based low-dimensional feature extraction from fft spectral envelopes for statistical parametric speech synthesis," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5535–5539.

[23] J. Markel and A. Gray, *Linear prediction of speech*. Springer Science & Business Media, 2013, vol. 12.

[24] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in neural information processing systems*, 2007, pp. 153–160.

[25] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[26] 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, no. Dec, pp. 3371–3408, 2010.

[27] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. of the Thirteenth Int. Conf. on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. of the IEEE int. conf. on computer vision*, 2015, pp. 1026–1034.

[29] J. Guo, U. A. Nookala, and A. Alwan, "CNN-based joint mapping of short and long utterance i-vectors for speaker verification using short utterances," *Proc. of INTERSPEECH*, pp. 3712–3716, 2017.

[30] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, and D. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon technical report N*, vol. 93, 1993.

[31] J. Abel and T. Fingscheidt, "Artificial speech bandwidth extension using deep neural networks for wideband spectral envelope estimation," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 71–83, 2018.

[32] P. Kabal, "TSP speech database," *McGill University, Database Version : 1.0*, pp. 02–10, 2002.

[33] F. Chollet *et al.*, "Keras," https://github.com/keras-team/keras, 2015.

[34] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

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

[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Int. conf. on machine learning*, 2015, pp. 448–456.

[37] R. Gray, A. Buzo, A. Gray, and Y. Matsuyama, "Distortion measures for speech processing," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 367–376, 1980.

[38] "ITU-T Recommendation P.862.2 : Wideband extension to Recommendation P.862 for the assessment of wideband telephone networks and speech codecs," *ITU*, 2005.