SPECTRO-TEMPORAL NEURAL FACTORIZATION FOR SPEECH DEREVERBERATION

Jen-Tzung Chien *Kuan-Ting Kuo*

Department of Electrical and Computer Engineering, National Chiao Tung University, Taiwan

ABSTRACT

This study presents a spectro-temporal neural factorization (STNF) for speech dereverberation. Traditionally, a contextual window of spectro-temporal reverberant speech was unfolded into a one-way vector which was fed into a neural network to estimate the spectra of source speech at each time frame. Model parameters were trained by using the vectorized error backpropagation algorithm. System performance is constrained because *contextual correlations* and *common factors* in frequency and time horizons are disregarded. To compensate this weakness, a spectro-temporal factorization is incorporated to preserve the structural information in neural network training based on bi-factorized error backpropagation where the spectral and temporal factor matrices are estimated. Affine transformation in one-way neural network is generalized to the bilinear decomposition in bi-factorized neural network. The spectro-temporal features are extracted and forwarded to fully-connected layers for regression outputs. Such a STNF is further improved by merging with *long short-term memory* layer to capture the temporal features. Experiments results on 2014 REVERB Challenge demonstrate the meaningfulness of the factorized features and the merit of integrating these features for speech dereverberation.

Index Terms— spectro-temporal neural factorization, factorized error backpropagation, speech dereverberation

1. INTRODUCTION

Matrix factorization and deep neural network (DNN) have been extensively developing in the areas of signal processing and machine learning with numerous applications ranging from speech recognition [1] to computer vision, source separation [2, 3, 4, 5], music information retrieval and natural language processing. Many extensions have been proposed to discover the insights to improve system performance from different perspectives. Basically, matrix factorization performs two-way decomposition and is generalizable to tensor factorization for multiple-way observations [6]. The *front-end* processing based on signal decomposition helps extracting meaningful features in *back-end* modeling for regression or classification. In the literature, several works have been proposed to strengthen the modeling capability by integrating neural network and tensor factorization. In [7], a deep tensor neural network was constructed by cascading the double projection layer and the tensor layer so as to learn the complimentary features from two hidden vectors. The multi-way tensor weights were estimated to capture the relations between features or neurons. The inputs were still one-way vectors. In [8], the convolutional neural network (CNN) was developed to extract the spatial features through convolution layer followed by pooling layer where no factorization was performed. In [9, 10, 11], a tensor classification network was proposed for image recognition and speech recognition where the multi-way inputs were factorized and fed into a classification neural network.

This paper presents the spectro-temporal neural factorization (STNF) in a layer-wise regression model with applications for speech dereverberation. Our idea is to relax the limitation in the vectorized neural network and preserve the spectro-temporal features to learn regression outputs for source signals by using the spectro-temporal input matrices. We improve the learning representation by seamlessly combining matrix factorization and neural network. By using the spectro-temporal factorized neural network, the contextual features in hybrid frequency and time domains are extracted to learn structural abstraction in a deep model. A two-way factorized error backpropagation is proposed to realize STNF for spectral mapping and dereverberation. The gradients for minimization of regression errors are calculated by transpose factorization. There are different settings in the implementation. The STNF layers can be either cascaded with the fully connected layers in feedforward neural network or connected with the long short-term memory layer in recurrent neural network. The STNF layers can be also merged with convolutional layers to learn the shared and factorized weights for source separation. The benefit of applying STNF in different settings is shown by experiments on speech dereverberation.

2. RELATED WORKS

2.1. Spectro-temporal factorization

In signal processing, we usually represent an observed time-series signal, e.g. speech or music, using a spectro-temporal data matrix $\mathbf{X}$ which contains the log magnitude spectra calculated by short-time Fourier transform (STFT). According to Tucker decomposition, this matrix $\mathbf{X} = \{X_{ft}\} \in \mathcal{R}^{F \times T}$

with F frequency bins and T time frames can be factorized to obtain a core matrix $\mathbf{A} = \{A_{ij}\} \in \mathcal{R}^{I \times J}$ by

$$\mathbf{X} = \mathbf{A} \times_1 \mathbf{U} \times_2 \mathbf{V} \tag{1}$$

where $\times_n$ denotes the model-n product and $\mathbf{U} = \{U_{fi}\} \in \mathcal{R}^{F \times I}$ and $\mathbf{V} = \{V_{tj}\} \in \mathcal{R}^{T \times J}$ denote the factor matrices in two horizons. I and J indicate the reduced dimensions corresponding to F and T, respectively. Each entry in core matrix is expressed by $X_{ft} = \sum_i \sum_j A_{ij} U_{fi} V_{tj}$. This decomposition can be solved by using the bilinear singular value decomposition [6]. Basically, matrix factorization is seen as a two-way realization of tensor factorization. It is important that the *inverse* of Tucker decomposition is yielded by

$$\mathbf{A} = \mathbf{X} \times_1 \mathbf{U}^{\dagger} \times_2 \mathbf{V}^{\dagger} \tag{2}$$

where $\mathbf{U}^{\dagger} = (\mathbf{U}^{\top}\mathbf{U})^{-1}\mathbf{U}^{\top}$ is the pseudo-inverse of $\mathbf{U}$. The core matrix $\mathbf{A}$ is viewed as a *factorized* feature matrix of the spectro-temporal matrix $\mathbf{X}$. This study adopts this property to fulfill the spectro-temporal neural factorization for speech dereverberation.

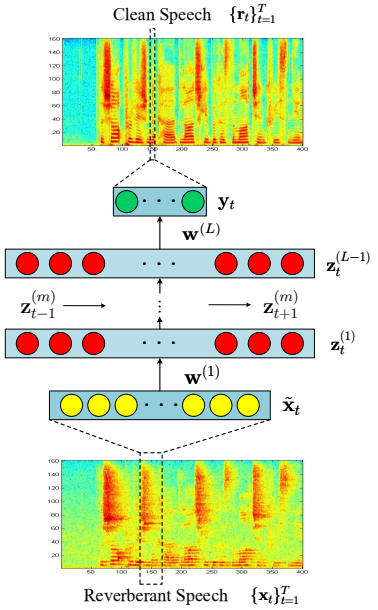

Fig. 1. Deep recurrent neural network for spectral mapping.

2.2. Dereverberation neural network

Figure 1 depicts a deep recurrent neural network (RNN) [12] for speech dereverberation which is developed for spectral mapping [13] from a reverberant speech $\mathbf{x}_t = \{X_{ft}\}$ to a clean speech $\mathbf{r}_t = \{R_{ft}\}$. The input vector $\tilde{\mathbf{x}}_t = [\mathbf{x}_{t-d}^{\top}, \ldots, \mathbf{x}_t^{\top}, \ldots, \mathbf{x}_{t+d}^{\top}]^{\top}$ consists of a window of $2d+1$ frames of reverberant speech centered at time t. Each frame $\mathbf{x}_t$ has values in F frequency bins. In the implementation, the forward and backward calculations are required to train this vector-based neural network. In forward pass, the affine transformation and nonlinear activation are calculated by $\mathbf{z}_t^{(l)} = h(\mathbf{a}_t^{(l)}) = h(\mathbf{w}^{(l)}\mathbf{z}_t^{(l-1)})$ in each fully-connected (FC) layer l. Nonlinear activation function $h(\cdot)$ can be sigmoid or ReLU. The recurrent layer m is calculated by

$$\mathbf{z}_t^{(m)} = h(\mathbf{a}_t^{(m)}) = h(\mathbf{w}^{(m)}\mathbf{z}_t^{(m-1)} + \mathbf{w}^{(mm)}\mathbf{z}_{t-1}^{(m)}) \tag{3}$$

where forward weights $\mathbf{w}^{(m)}$ and recurrent weights $\mathbf{w}^{(mm)}$ are both functioned. Temporal information is captured for speech dereverberation. Notably, the input $\tilde{\mathbf{x}}_t$, hidden features $\{\mathbf{z}_t^{(l)}, \mathbf{z}_t^{(m)}\}$ and output $\mathbf{y}_t$ are all vectors which are individually calculated in different time frames t. In backward pass, the weight parameters $\mathbf{\Theta} = \{\mathbf{w}^{(l)}, \mathbf{w}^{(m)}, \mathbf{w}^{(mm)}\}$ in different layers are estimated according to the vector-based error backpropagation algorithm where the sum-of-squares error function using T training samples $\{\mathbf{X}, \mathbf{R}\} = \{X_{ft}, R_{ft}\}$

$$E(\mathbf{\Theta}) = \sum_n E_n(\mathbf{\Theta}) = \frac{1}{2}\sum_t \sum_f (Y_{ft}(\mathbf{\Theta}) - R_{ft})^2 \tag{4}$$

is minimized. n means the index of minibatch. $\mathbf{y}_t = \{Y_{ft}\}$ is the outputs of dereverberation neural network at time t. However, RNN suffers from the problem of gradient vanishing and exploding. Long short-term memory network [14] is feasible to deal with this problem. In this spectral mapping, each frame is represented by an unfolded vector with dimension $(2d+1) \cdot F$. This one-way vector is fed into traditional neural network for training and prediction. The spatial information in a context window was partially disregarded. In [8], CNN was developed to catch spatial features by means of convolution layers and pooling layers. There was no factorized features extracted in different ways for classification or regression.

3. FACTORIZED NEURAL NETWORK

The factorized neural network is built by merging spectro-temporal features in dereverberation neural network.

3.1. Spectro-temporal neural factorization

Given a spectro-temporal input matrix $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^T = \{X_{ft}\}$, the latent matrix $\mathbf{A}^{(1)} = \{A_{ji}^{(1)}\}$ in the first hidden layer is obtained by the factorization through two factor matrices $\mathbf{U}^{(1)} = \{U_{if}^{(1)}\} \in \mathcal{R}^{I \times F}$ and $\mathbf{V}^{(1)} = \{V_{jt}^{(1)}\} \in \mathcal{R}^{J \times T}$ via

$$\mathbf{A}^{(1)} = \mathbf{X} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{V}^{(1)} = \sum_f \sum_t X_{ft}(\mathbf{u}_f^{(1)} \circ \mathbf{v}_t^{(1)}) \tag{5}$$

which is seen as a summation of outer products of all individual columns of factor matrices $\mathbf{u}_f^{(1)}$ and $\mathbf{v}_t^{(1)}$ along frequency and time horizons, respectively. After this bilinear transformation, we obtain a spectro-temporal feature matrix $\mathbf{Z}^{(1)} = \{Z_{ij}^{(1)}\} \in \mathcal{R}^{I \times J}$ through an activation function

$\mathbf{Z}^{(1)} = h(\mathbf{A}^{(1)})$. This feature matrix is then factorized and activated again as $\mathbf{A}^{(2)}$ and $\mathbf{Z}^{(2)}$ in the second hidden layer, respectively. The feedforward computation is run layer by layer until layer L to find dereverberant speech $\mathbf{Y} \in \mathcal{R}^{F \times T}$ corresponding to reverberant speech $\mathbf{X}$ in spectral mapping. In the last layer L, the dereverberant speech matrix is calculated by

$$\mathbf{Y} = h(\mathbf{A}^{(L)}) = h(\mathbf{Z}^{(L-1)} \times_1 \mathbf{U}^{(L)} \times_2 \mathbf{V}^{(L)}) \quad (6)$$

where $\mathbf{A}^{(L)} = \{A_{ft}^{(L)}\}$ denotes the activation matrix in layer L and $\mathbf{Z}^{(L-1)} = \{Z_{ij}^{(L-1)}\}$ denotes the output matrix of hidden units in layer $L-1$. Given the dereverberant output matrix $\mathbf{Y}$ and the clean speech matrix $\mathbf{R}$, we calculate the sum-of-squares error function $E(\mathbf{\Theta})$ in Eq. (4) for optimization to estimate the STNF parameters $\mathbf{\Theta}$ which contain the spectral parameters $\mathbf{U}^{(l)}$ and the temporal parameters $\mathbf{V}^{(l)}$ in different layer l. Different from vector-based neural network in Section 2.2 calculating the propagation of input $\mathbf{x}_t$ at each time t independently, the proposed STNF conducts the layer-wise calculation over a window of reverberant speech frames $\mathbf{X}$. The hybrid spectro-temporal features are jointly characterized.

3.2. Factorized error backpropagation

To estimate model parameters $\mathbf{\Theta} = \{\mathbf{U}^{(l)}, \mathbf{V}^{(l)}\}$, we perform the stochastic gradient descent (SGD) algorithm by calculating the gradients of E_n over a minibatch data index n with respect to individual parameters in $\mathbf{\Theta}$. Starting from the regression layer L, we calculate the local gradient

$$\frac{\partial E_n}{\partial A_{ft}^{(L)}} = \frac{\partial E_n}{\partial Y_{ft}} \frac{\partial Y_{ft}}{\partial A_{ft}^{(L)}} = (Y_{ft} - R_{ft}) h'(A_{ft}^{(L)}) \triangleq \mathcal{D}_{ft}^{(L)} \quad (7)$$

of an output neuron at frequency f and time t and then find the gradients

$$\begin{aligned} \frac{\partial E_n}{\partial U_{if}^{(L)}} &= \sum_t \frac{\partial E_n}{\partial A_{ft}^{(L)}} \frac{\partial A_{ft}^{(L)}}{\partial U_{if}^{(L)}} = \sum_t \mathcal{D}_{ft}^{(L)} \left(\sum_k Z_{ki}^{(L-1)} V_{kt}^{(L)} \right) \\ &= \langle \boldsymbol{\mathcal{D}}_{f:}^{(L)}, \mathbf{Z}_{:i}^{(L-1)} \times_2 \mathbf{V}^{(L)} \rangle \\ \frac{\partial E_n}{\partial V_{jt}^{(L)}} &= \langle \boldsymbol{\mathcal{D}}_{:t}^{(L)}, \mathbf{Z}_{j:}^{(L-1)} \times_1 \mathbf{U}^{(L)} \rangle \end{aligned}$$

for updating parameters $\mathbf{U}^{(L)}$ and $\mathbf{V}^{(L)}$, respectively. We assume $\mathbf{Z}^{(L-1)} = \{Z_{ji}^{(L-1)}\} \in \mathcal{R}^{J \times I}$. After updating $\{\mathbf{U}^{(L)}, \mathbf{V}^{(L)}\}$, we propagate local gradients from $\boldsymbol{\mathcal{D}}^{(L)} = \{\mathcal{D}_{ft}^{(L)}\}$ in layer L back to $\boldsymbol{\mathcal{D}}^{(L-1)} = \{\mathcal{D}_{ji}^{(L-1)}\}$ in layer $L-1$ through

$$\frac{\partial E_n}{\partial A_{ji}^{(L-1)}} = \sum_f \sum_t \frac{\partial E_n}{\partial A_{ft}^{(L)}} \frac{\partial A_{ft}^{(L)}}{\partial Z_{ji}^{(L-1)}} \frac{\partial Z_{ji}^{(L-1)}}{\partial A_{ji}^{(L-1)}} \triangleq \mathcal{D}_{ji}^{(L-1)}. \quad (8)$$

which can be written as matrix form $\boldsymbol{\mathcal{D}}^{(L-1)} = h'(\mathbf{A}^{(L-1)}) \odot (\boldsymbol{\mathcal{D}}^{(L)} \times_1 (\mathbf{U}^{(L)})^\top \times_2 (\mathbf{V}^{(L)})^\top)$ where $\odot$ denotes the element-wise product. These local gradients $\boldsymbol{\mathcal{D}}^{(L-1)}$ will be used to calculate gradients for updating the spectral and temporal factor matrices $\{\mathbf{U}^{(L-1)}, \mathbf{V}^{(L-1)}\}$ in layer $L-1$. Notably, the local gradient $\boldsymbol{\mathcal{D}}^{(L-1)}$ in layer $L-1$ is calculated in a form of *transpose factorization* by using $\{\boldsymbol{\mathcal{D}}^{(L)}, (\mathbf{U}^{(L)})^\top, (\mathbf{V}^{(L)})^\top\}$ in layer L. It is meaningful that the gradients with respect to spectral factor $U_{if}^{(L)}$ at frequency f and temporal factor $V_{jt}^{(L)}$ at time t are calculated by summing up all information over time frames t and frequency bins f, respectively.

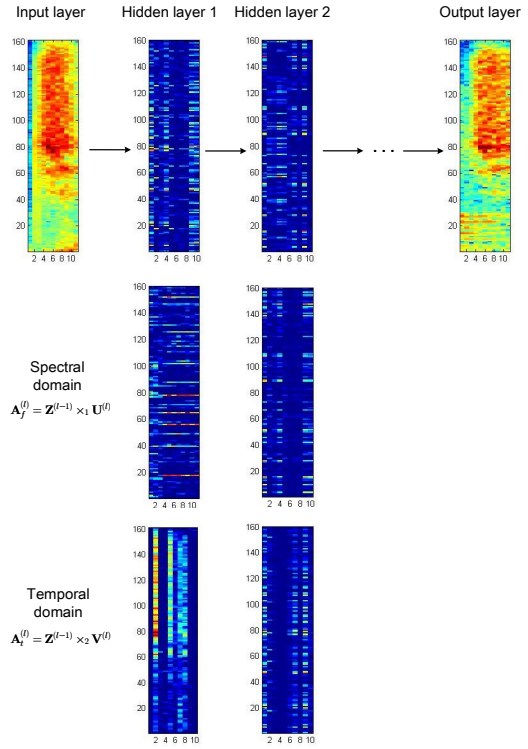

Fig. 2. Analysis of hidden layers of speech dereverberation in spectral and temporal domains.

Figure 2 illustrates how an input matrix $\mathbf{X}$ of a short segment of reverberant speech is transformed to find the activation matrices $\mathbf{A}^{(l)}$ in two hidden layers $l = 1, 2$ and the output matrix of dereverberant speech $\mathbf{Y}$. The second and third rows show the visualization of spectral domain $\mathbf{A}_f^{(l)}$ and temporal domain $\mathbf{A}_t^{(l)}$, respectively. The factorized features in spectral and temporal domains are well reflected. The features in second hidden layer are more abstract than those in first layer. The smeared spectrogram is enhanced by using STNF layers.

4. EXPERIMENTS

4.1. Experimental setup

We evaluated the proposed method by using 2014 REVERB Challenge dataset [15], which contained the reverberation and stationary noise [16]. There were eight different acoustic con-

ditions of which six were simulated by convolving the WSJ-CAM0 corpus with three room impulse responses at near (50 cm) and far microphone distances (200 cm), and adding the stationary noise recordings from the same rooms at signal-to-noise-ratio (SNR) of 20dB (SimData). There were 1484 and 2176 utterances from 20 and 28 speakers collected as the development and evaluation data, respectively. The other two conditions were real recordings (RealData) in a reverberant meeting room at two microphone distances (near at 100 cm and far at 250 cm) with stationary noise, taken from the MC-WSJ-AV corpus [17]. 179 and 372 utterances from five and ten speakers were collected as development and evaluation data, respectively. Reverberation time T_{60} ranged from 0.25s to 0.7s, but was unknown at test time. Training data had 7862 utterances from 92 speakers. Dereverberation performance was evaluated by the speech-to-reverberation modulation energy ratio (SRMR) [18] (higher is better) and the perceptual evaluation of speech quality (PESQ) (higher is better). The results of SimData and RealData were averaged over the corresponding conditions.

In the implementation, 320-point STFT was calculated. At each time frame, an input matrix consisting of five neighboring frames with dimension 160×11 was formed and regressed into a target frame 160×1 in dereverberant speech. STNF layers were empirically configured with a fixed size 90×8. Hidden and output layers were implemented by using ReLU and sigmoid activations, respectively. The baseline DNN system was built by using two, three or four FC layers (FC2, FC3, FC4). Using the proposed method, there were two, three or four STNF layers followed by one FC layer to implement the STNF2-FC, STNF3-FC, STNF4-FC. The topologies of three LSTM layers (LSTM3), three STNF layers (STNF3) and two or three STNF layers followed by LSTM layer (STNF2-LSTM, STNF3-LSTM) were also examined. CNN was implemented by referring [9]. SGD algorithm was run using a mini-batch size of 128 frames with ℓ_2 regularization where regularization parameter was selected from validation data. Adam algorithm was applied. Weights were randomly initialized by an uniform distribution. We delayed the LSTM outputs by five frames to make prediction of a future frame. LSTM with 256 memory cells was implemented. The size of hidden matrices in STNF and STNF-LSTM was comparable with the size of hidden vectors in FC.

4.2. Experimental results

Table 1 reports the results of SRMR and PESQ in speech dereverberation by using different neural network methods under the conditions of SimData and RealData. Number of parameters is included in the comparison. SRMR and PESQ are improved by increasing the number of FC layers in hidden structure of DNN. Using various methods, the improvement of SRMR and PESQ in RealData condition is much higher than that in SimData condition. LSTM and CNN do increase the values of SRMR and PESQ when compared with DNN. Temporal and convolutional information is helpful for speech dereverberation. The stand-alone STNF layers do not work well. However, the improvement of combing STNF layers with FC layer is clearly obtained. The highest SRMR and PESQ are achieved by cascading STNF layers with LSTM layer. This result is considerably better than that of CNN. In terms of total number of parameters, STNF layer is much more efficient than FC layer because STNF parameters are counted by *adding* the dimensions of factor matrices in spectral and temporal domains. But, the parameters of FC layer are counted by *multiplying* the dimensions of one-way matrix between two layers. STNF layer is superior to FC layer with higher SRMR and PESQ and lower number of parameters.

Model	SimData		RealData		# Par
	SRMR	PESQ	SRMR	PESQ	
Unprocessed	3.70	2.18	3.81	2.89	–
FC2	3.79	2.32	7.78	5.93	1.90
FC3	3.85	2.83	8.43	7.19	2.42
FC4	3.81	2.73	8.45	7.14	2.94
LSTM3	3.90	2.92	8.95	7.94	2.81
CNN	3.92	2.99	8.62	7.68	2.65
STNF3	3.78	2.29	7.21	6.88	0.05
STNF2-FC	3.89	2.84	8.98	7.19	0.66
STNF3-FC	4.09	3.19	9.93	8.13	0.67
STNF4-FC	3.97	3.10	8.83	7.99	0.68
STNF2-LSTM	**4.20**	**3.33**	9.83	**9.53**	0.78
STNF3-LSTM	4.13	3.29	**10.09**	9.13	**0.79**

Table 1. Comparison of SRMR (in dB), PESQ (in dB) and number of parameters (in millions) under the conditions of SimData and RealData by using different models.

5. CONCLUSIONS

We have presented a spectro-temporal neural factorization to build a deep recurrent neural network for speech dereverberation. The factorized features were extracted to capture the abstraction in spectral and temporal spaces and enhance the blurred spectrogram in a distorted speech due to the room reverberation. The forward calculation in layer-wise neural network was run as a matrix factorization while the backward calculation was performed to propagate the local gradient layer by layer which was also seen as an operation of matrix factorization. The resulting STNF layers outperformed the fully-connected layers in the experiments of speech dereverberation in terms of quality measures using SRMR and PESQ. STNF layers used smaller number of parameters than FC layers. The topology of cascading STNF layers with LSTM layer obtained the best measures among different topologies. Future work will be extended to incorporating convolutional neural network in STNF neural network. Tensor factorized neural network will be implemented using multi-way data.

6. REFERENCES

[1] D. Yu, G. Hinton, N. Morgan, J.-T. Chien, and S. Sagayama, "Introduction to the special section on deep learning for speech and language processing," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 4–6, 2012.

[2] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations - Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*, Wiley, 2009.

[3] J.-T. Chien and P.-K. Yang, "Bayesian factorization and learning for monaural source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 185–195, 2016.

[4] C.-C. Hsu, T.-S. Chi, and J.-T. Chien, "Discriminative layered nonnegative matrix factorization for speech separation," in *Proc. of Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2016, pp. 560–564.

[5] K.-W. Tsou and J.-T. Chien, "Memory augmented neural network for source separation," in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing*, 2017, pp. 1–6.

[6] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[7] B. Hutchinson, L. Deng, and D. Yu, "Tensor deep stacking networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1944–1957, 2013.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, pp. 2278–2324, 1998.

[9] J.-T. Chien and Y.-T. Bao, "Tensor-factorized neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

[10] Y.-T. Bao and J.-T. Chien, "Tensor classification network," in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing*, 2015, pp. 1–6.

[11] J.-T. Chien and C. Shen, "Deep neural factorization for speech recognition," in *Proc. of Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2017, pp. 3682–3686.

[12] G.-X. Wang, C.-C. Hsu, and J.-T. Chien, "Discriminative deep recurrent neural networks for monaural speech separation," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 2544–2548.

[13] K. Han, Y. Wang, D. Wang, W. S. Woods, I. Merks, and T. Zhang, "Learning spectral mapping for speech dereverberation and denoising," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 6, pp. 982–992, 2015.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, A. Sehr, W. Kellermann, and R. Maas, "The REVERB challenge: a common evaluation framework for dereverberation and recognition of reverberant speech," in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2013, pp. 1–4.

[16] J.-T. Chien and Y.-C. Chang, "Bayesian learning for speech dereverberation," in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2016, pp. 1–6.

[17] M. Lincoln, I. McCowan, J. Vepa, and H. K. Maganti, "The multi-channel Wall Street Journal audio visual corpus (MC-WSJ-AV): specification and initial experiments," in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding*, 2005, pp. 357–362.

[18] T. H. Falk, C. Zheng, and W.-Y. Chan, "A non-intrusive quality and intelligibility measure of reverberant and dereverberated speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1766–1774, 2010.