GEOMETRY CONSTRAINED PROGRESSIVE LEARNING FOR LSTM-BASED SPEECH ENHANCEMENT

Xin Tang[1]*, Jun Du*[1]*, Li Chai*[1]*, Yannan Wang*[2]*, Qing Wang*[2]*, Chin-Hui Lee*[3]

[1] University of Science and Technology of China, HeFei, China
[2] Tencent Technology (Shenzhen) Co., Ltd, Shenzhen, China
[3] Georgia Institute of Technology, Atlanta, Georgia, USA

ABSTRACT

In our previous work, a progressive learning framework for long short-term memory (LSTM)-based speech enhancement was proposed to improve the performance in low SNR environment, where each LSTM layer is guided to learn an intermediate target with a specific SNR gain via the MMSE criterion. However, the constraint relationship among these targets is not considered in the objective function. In this paper, we incorporate two kinds of geometric constraints among these targets into the objective function to help LSTM achieve better training. One constraint is edge constraint and the other is the centroid constraint. In addition, we propose a method for constructing the intermediate targets online. It saves device storage space and alleviates the trouble of manually constructing intermediate targets. Experiment results demonstrate these geometric constraints can bring remarkable improvements in low SNR environments.

Index Terms— Speech enhancement, progressive learning, LSTM, geometric constraint, deep learning

1. INTRODUCTION

The goal of speech enhancement is to improve the perceptual quality and speech intelligibility by suppressing the ambient noise components present in the recorded speech. Single-channel speech enhancement has attracted much research attention due to its importance in real-world applications including mobile speech communication, hearing aids and robust automatic speech recognition. Traditional speech enhancement methods have been studied for decades including spectral subtraction [1], Wiener filtering [2], minimum mean squared error (MMSE) estimation [3] and optimally-modified log-spectral amplitude (OM-LSA) speech estimator [4]. However, these traditional methods can not well deal with highly non-stationary noise, which leads to difficulty in using these methods in real-world application scenarios.

Recently, deep learning has been successfully applied to various tasks [5] [6] [7], which greatly motivated the investigation of deep learning for speech enhancement. The investigation is mainly from the aspects of learning targets, deep neural network (DNN) structures and input features. There are two groups of learning targets: masking-based targets and mapping-based targets. Masking-based targets include ideal binary mask [8], ideal ratio mask [9], spectral magnitude mask [10], complex ideal ratio mask [11] and phase-sensitive mask [12]. They describe the time-frequency relationships between the target speech and background noise. Mapping-based targets include short-time Fourier transform (STFT) magnitude spectra, STFT log-power spectra [13] and mel spectra. They are the spectral representations of the target speech. As for the input features, most researchers operate at the spectral domain. Others operate at some higher-level features or the waveform level, training the end-to-end model [14]. In addition, many types of DNNs have been utilized in DNN-based speech enhancement, such as feed-forward DNNs [15], recurrent neural networks [16], convolutional neural networks [17] and generative adversarial networks [18]. DNN-based speech enhancement has made great progress. However, it still suffers from performance degradation in low signal-to-noise-ratio (SNR) environments regardless of its strong modeling ability. [19] proposed a preliminary progressive learning (PL) framework on DNN model to improve the speech intelligibility in low SNR environments. And each hidden layer of the DNN network is guided to learn an intermediate target with a specific SNR gain explicitly. [20] continued to study the PL with advanced long short-term memory (LSTM) network. In addition, to alleviate the possible information loss, it proposed densely connected progressive learning in which the input and the estimations of intermediate targets are spliced together to learn the next target. Compared with [19], better performance was achieved.

In this study, we continue to explore the PL from the aspect of target optimization. In the previous PL framework [19] [20], the constraint relationship among targets is not considered in the objective function. In this paper, we propose two kinds of constraint relations from the geometric point of view, namely edge constraint and centroid constraint. They are incorporated into the objective function as a regularization term. Experimental results demonstrate that performance improvement can be achieved by using these two constraints. In addtion, we also proposed the method for online calculating the target that the intermediate layer needs to learn in the frequency domain. In prior work [20], we generate these intermediate targets directly in the time domain and then store them on the hard disk. When the number of targets is large, this will take a lot of time and storage space to generate and store these intermediate targets. We also propose a method for automatically constructing the intermediate targets online via the approximate relationship in the power spectrum. In comparison, the proposed method can achieve a comparable performance.

The rest of the paper is organized as follows. In Section 2, we describe the proposed geometry constrained PL. In Section 3, we present the experiments. Finally, we conclude in Section 4.

2. GEOMETRY CONSTRAINED PL

2.1. Review of densely connected PL framework

In our prior work, [20] has proposed a densely connected progresssive learning framework, which is designed to improve speech intel-

978-1-5090-6631-5/20/$31.00 ©2020 IEEE

ligibility in low SNR environments. The procedure of direct mapping from noisy to clean speech is decomposed into multiple stages with SNR increasing progressively by guiding hidden layers in the LSTM network to learn target explicitly. The densely connected PL framework with 2 targets is illustrated in Fig. 1. $\mathbf{t}_0$, $\mathbf{t}_1$ and $\mathbf{t}_2$ are denoted as the log-power spectra (LPS) of input noisy speech, the first target and the second target, respectively. All the target layers are designed to learn intermediate speech features with higher SNRs or clean speech as shown in Fig. 1. In order to alleviate the information loss and make full use of learning targets, a densely connected architecture is adopted in PL framework, namely concating the estimated target and current input to learn next layer. As for optimization procedure, a weighted MMSE criterion is designed to optimize all network parameters in the manner of back propagation through time with gradient descent [21] randomly initialized with K target layers as follows

$$E = \sum_{k=1}^{K} \beta_k E(k) \tag{1}$$

$$E(k) = \mathrm{MSE}(\hat{\mathbf{t}}_k, \mathbf{t}_k) = \|\hat{\mathbf{t}}_k - \mathbf{t}_k\|^2 \tag{2}$$

where K is the number of target. β_k is a weighting factor for the k-th target layer to balance the MSE loss of multiple targets.

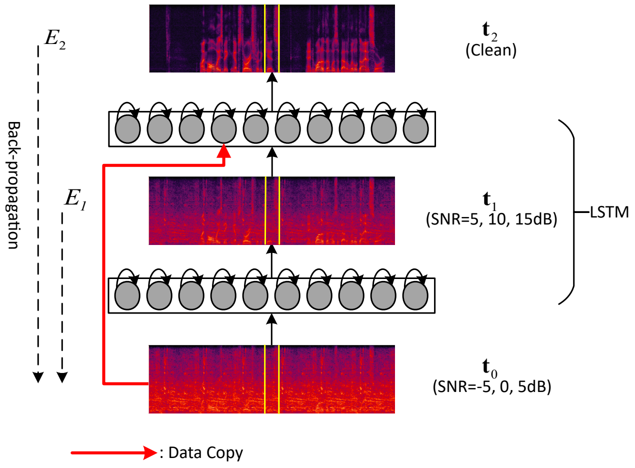


Fig. 1. The architecture of PL framework (2 targets as an example)

2.2. Motivation

In the previous intermediate optimization process, we simply fit the intermediate estimated target to the learning target, which ignores constraint relationship between targets. In other words, previous optimization method is equivalent to a point-to-point mapping, leading to inconsistent learning pace between targets. In geometry, the target is regarded as a point in D-dimensional space where D is the dimension of LPS feature. Accordingly, we propose two constraints, namely edge constraint and centroid constraint, to rectify the process of target optimization procedure, and then further improve the overall performance.

2.3. Derivation of construting intermediate targets

In time domain, background noise is an additive signal to clean speech so that the noisy speech $x_k(t)$ can be formulated as:

$$x_k(t) = s(t) + \alpha_k \cdot n(t) \tag{3}$$

where $x_k(t)$, $s(t)$, $n(t)$ denote noisy speech, clean speech and noise, respectively. Besides, α_k is an adjustable factor of utterance level employed to control the SNR level. When $k = 0, 1, 2$, the $x_k(t)$ is denoted as the time domain representation of $\mathbf{t}_0$, $\mathbf{t}_1$, $\mathbf{t}_2$ as shown in Fig. 1, respectively.

Thus, in PL framework, suppose the number of target is K, $K \geq 1$, the SNR of k-th target is snr_k and the SNR gain between k-th and $(k-1)$-th target is g_k, namely $g_k = \mathrm{snr}_k - \mathrm{snr}_{k-1}$. In this way, the 0-th target and the K-th target refers to input noisy speech and clean speech, respectively. Considering the definition of SNR, due to $\mathrm{snr}_k - \mathrm{snr}_0 = \sum_{i=1}^{k} g_i$, we can derive the relationship between α_k and α_0 as shown as follows

$$\alpha_k = \alpha_0 / 10^{\sum_{i=1}^{k} g_i/20} \tag{4}$$

Then considering in frame level, we perform STFT for the k-th target, and then we approximately compute the corresponding power spectrum as follows

$$|X_k(d)|^2 = |S(d)|^2 + {\alpha_k}^2 \cdot |N(d)|^2 \tag{5}$$

where d is denoted as freqency bin index and here we assume that $S(d)$ and $N(d)$ is irrelevant.

According to Eq. (4) and Eq. (5), we realize that the power spectrum $|X_k(d)|^2$ can be expressed by a linear combination of $|X_0(d)|^2$ and $|X_K(d)|^2$, which is shown as follows

$$|X_k(d)|^2 = p_k |X_0(d)|^2 + q_k |X_K(d)|^2 \tag{6}$$

where $p_k = 1/10^{\sum_{i=1}^{k} g_i/10}$ and $q_k = 1 - p_k$. Because the log-power spectra is D-dimensional vector, namely $d = 1, 2, ..., D$, the $\mathbf{t}_k$ can be expressed as

$$\mathbf{t}_k = [\, \log|X_k(1)|^2,\ \log|X_k(2)|^2, ...,\ \log|X_k(D)|^2 \,]^\top \tag{7}$$

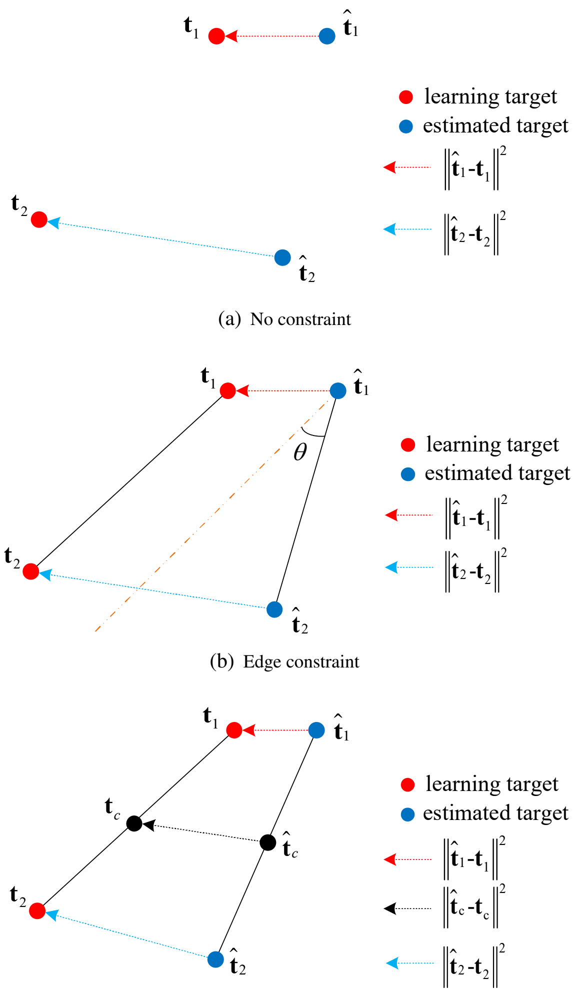
By using Eq. (6), we have the new expression in LPS domain:

$$\mathbf{t}_k = \log\left(p_k \cdot e^{\mathbf{t}_0} + q_k \cdot e^{\mathbf{t}_K}\right) \tag{8}$$

where $\mathbf{t}_0$, $\mathbf{t}_k$, $\mathbf{t}_K$ denote the D-dimensional LPS of input noisy speech, the k-th target and the last target, respectively. In this way, no matter how many targets, we can use Eq. (8) to calculate the intermediate targets online which not only removes the step of constructing the intermediate targets in the time domain, but also saves a lot of storage space. And the experimental results also demonstrate that this method can also achieve a comparable performance, even bring slight improvements in some cases. Moreover, according to Eq. (8), we can find there is indeed strong connection between different targets which inspires us to discover the constraint between the targets in the next subsections. In the study all PL related experiments use this method to generate online targets.

2.4. Edge constraint

In geometry, the D-dimensional learning target can be regarded as a point in D-dimensional space. For ease of analysis, we take two targets as an example in the PL framework. As shown in Fig. 2(b), $\mathbf{t}_1$,$\mathbf{t}_2$ represent two learning targets, namely D-dimensional vector,



(a) No constraint

(b) Edge constraint

(c) Centroid constraint

Fig. 2. Illustration of the interpretation for geometric constraints

as shown by red points. And $\hat{\mathbf{t}}_1$ and $\hat{\mathbf{t}}_2$ represent the corresponding estimated targets obtained by the network output during training as shown by the blue points. In standard PL framework, we just perform a point-to-point mapping by minimizing Euclidean distance between learning target and estimated target as shown by the dotted line. Whereas in the related experiments, we observe that the MSE loss of the latter target is always greater than the former one, which means the MSE loss of $\mathbf{t}_2$ is greater than $\mathbf{t}_1$. The interpretation is that the top layer increases the learning difficulty due to the large gap with the input noisy speech. That is why blue dotted line is longer than red dotted line. Here we connect blue and red dots respectively as shown by the solid black line. In order to alleviate pace inconsistency of target learning, intuitively we consider minimizing the angle between two black lines (edges), namely the angle θ, from the perspective of geometry. In other word, we expect these two black edges to be parallel so that we utilize cosine distance to measure the angle between learning targets and estimated targets which is expressed as follows

$$\min(1 - \cos < \mathbf{t}_2 - \mathbf{t}_1, \hat{\mathbf{t}}_2 - \hat{\mathbf{t}}_1 >) \tag{9}$$

$$\cos < \mathbf{t}_2 - \mathbf{t}_1, \hat{\mathbf{t}}_2 - \hat{\mathbf{t}}_1 >= \frac{(\mathbf{t}_2 - \mathbf{t}_1) \cdot (\hat{\mathbf{t}}_2 - \hat{\mathbf{t}}_1)}{\|\mathbf{t}_2 - \mathbf{t}_1\| \|\hat{\mathbf{t}}_2 - \hat{\mathbf{t}}_1\|} \tag{10}$$

Therefore, we introduce an edge constraint as a regularization term which enables optimization of all targets to keep pace with each other as far as possible. Similarly, when the number of targets increases to three or more, we can subdivide it into two targets per group to consider and then add the constraints at corresponding loss function. Combining with the particularity of the progressive learning framework structure, our final loss function with the edge constraint on the k-th target can be written as

$$L1(k) = E(k) + \sum_{i=1}^{k-1} \lambda_i (1 - \cos < (\hat{\mathbf{t}}_k - \hat{\mathbf{t}}_i), (\mathbf{t}_k - \mathbf{t}_i) >) \tag{11}$$

where λ_i is denoted as weight factor, which is used to adjust the weight of the constraint term. $E(k)$ is the same as Eq. (2). The total loss is defined as:

$$L1 = \sum_{k=1}^{K} \beta_k L1(k) \tag{12}$$

2.5. Centroid constraint for post-processing

Similar to [19], the post-processing method is to average the estimations of multiple targets, which is equivalent to make tradeoffs between noise reduction and introduced nonlinear distortions. However, if we look at the post-processing from a geometric perspective like edge constraint mentioned in Section 2.4, then the post-processing can be regarded as using the centroid of estimated targets as the final result. As shown in Fig. 2(c), the $\mathbf{t}_c$ and $\hat{\mathbf{t}}_c$ are denoted as the centroid of the learning target and estimated target, respectively, as shown by the two black points. As there are only two targets, the centroid degenerates to the midpoint of the line segment, namely the black dotted line. Accordingly, the centroid has global information about all learning targets or estimated targets. It is instinctive to minimize the $\|\hat{\mathbf{t}}_c - \mathbf{t}_c\|^2$, which means gradually reducing the Euclidean distance between the two black points from a global perspective as shown in Fig. 2(c). On the one hand, from the perspective of target optimization, it seems that we increase an additional optimization target to make the distances between learning targets and estimated targets get closer, equivalent to certain emphasis on target optimization. On the other hand, the centroid constraint is consistent with our post-processing operations, which may achieve a better tradeoff of noise reduction and speech distortion. To obtain further improvements of the overall performance, we can add centroid constraint after adding the edge constraint. In this way, the edge information and global information of the targets are both taken into account instead of superficial point-to-point mapping. Therefore, the final loss function with edge constraint and centroid constraint on the k-th target can be written as (13)

$$L2 = L1 + \lambda \text{MSE}\left(\sum_{i=1}^{K} \hat{\mathbf{t}}_i, \sum_{i=1}^{K} \mathbf{t}_i\right) \tag{13}$$

where λ is the weight factor for adjusting the centroid constraint.

3. EXPERIMENTS AND RESULT ANALYSIS

3.1. Experimental setup

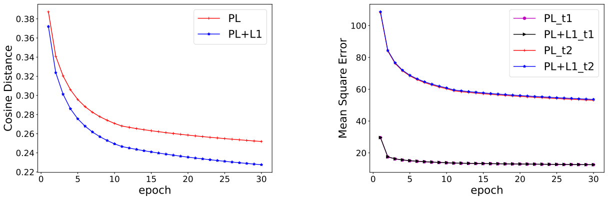
In our experiments, the clean speech data is derived from the WSJ0 corpus [22] and 115 noise types were selected as our noise database.

For training set, firstly we corrupted 7138 utterances (about 15 hours) from 83 speakers with 115 noise types [23] at three SNR levels (-5dB, 0dB, 5dB) to build a 45-hour training set composed of pairs of clean and noisy utterances. Similarly, 330 utterances from 8 other speakers, namely the Nov92 WSJ evaluation set, 6 unseen noises including buccaneer2, destroyerengine, destroyerops, factory1, pink, white from NOISEX-92 corpus [24], were used to construct the test set. Perceptual evaluation of speech quality (PESQ) [25] and short-time objective intelligibility (STOI) [26] are adopted to evaluate the intelligibility and quality of enhanced speech.

As for feature extraction, first the speech waveform was sampled at 16kHz, and the corresponding frame length was set to 32 msec (512 samples) with a frame shift of 16 msec (256 samples). A short-time Fourier analysis was employed to calculate the spectra of each overlapping windowed frame. Thus, the D-dimentional (D=257) LPS features were produced and normalized by global mean and variance before feeding them into the neural network [15].

For the training procedure, LSTM was used with 1024 units for each layer. Then we set $\beta_1 = 0.1$ and $\beta_2 = 1.0$ in Eq. 1.

3.2. Result analysis



(a) Cosine distance

(b) MSE loss on $\mathbf{t}_1$ and $\mathbf{t}_2$

Fig. 3. Learning curve of training procedure

Fig. 3 shows the learning curve of entire training phase. The curve in Fig. 3(a) gives the curve of cosine distance with respect to epochs. The blue line with '*' and red line with '+' refer to the standard PL and the standard PL with $L1$ loss function. Accordingly, we can observe that the θ indeed gradually decreases during the training phase according to the blue line with '*', whereas the **PL+L1** can obtain smaller θ compared with **PL**. Then looking at the Fig. 3(b), we find that the MSE loss on $\mathbf{t}_2$ of **PL+L1** is slightly larger than **PL** and the MSE loss on $\mathbf{t}_1$ is basically unchanged. The interpretation is that **PL+L1** sacrifices very little MSE loss to make the angle θ smaller which verify that adding edge constraints is feasible.

Table 1 lists the average PESQ and STOI results of Target1 and Target2 at different systems across six unseen noise types at -5dB, 0dB, 5dB. **PL_t1** and **PL_t2** refer to the standard PL framework by using $\mathbf{t}_1$ and $\mathbf{t}_2$ for enhancement in Fig. 1, respectively. **PL+L1** refers to the standard PL with the edge constraint, the corresponding weight factor λ is set to 20. So that **PL+L1_t1** and **PL+L1_t2** is denoted as the enhancement results of $\mathbf{t}_1$ and $\mathbf{t}_2$, respectively. Compare with **PL_t2**, the results of **PL+L1_t2** could achieve remarkable improvements for both STOI and PESQ, e.g., STOI increasing from 0.587 to 0.619 and PESQ increasing from 1.486 to 1.602 at SNR=-5dB. If we observe all SNR conditions, the gap between **PL+L1_t2** and **PL_t2** decreases as the SNR level increases. Besides, we notice that **PL_t1** can achieve higher STOI but lower STOI than **PL_t2**. The reason is that $\mathbf{t}_2$ removes noise excessively, resulting in very serious speech distortion. By adding the edge constraint, we decrease the gap of STOI between $\mathbf{t}_1$ and $\mathbf{t}_2$ when comparing **PL+L1_t1** and **PL+L1_t2** at all SNR level. Besides, as shown in Table 1, **PL_std_t1** and **PL_std_t2** refers to the $\mathbf{t}_1$ and $\mathbf{t}_2$ of standard PL in which we construct intermediate target in the time domain manually. Accordingly, we can find that the performance is comparable when comparing **PL_std_t1** with **PL_t1** or comparing **PL_std_t2** with **PL_t2**.

Table 1. The average PESQ and STOI comparison of all targets in different systems across 6 unseen noises at -5dB, 0dB, 5dB

	PESQ			STOI		
System	-5dB	0dB	5dB	-5dB	0dB	5dB
Noisy	1.31	1.60	1.94	0.606	0.731	0.842
PL_std_t1	1.41	1.80	2.21	0.648	0.779	0.874
PL_t1	1.36	1.78	2.21	0.647	0.778	0.874
PL+L1_t1	1.43	1.85	2.24	**0.657**	**0.785**	**0.879**
PL_std_t2	1.50	2.02	2.46	0.591	0.742	0.845
PL_t2	1.49	2.04	2.47	0.587	0.741	0.845
PL+L1_t2	**1.60**	**2.11**	**2.52**	0.619	0.759	0.855

Table 2. The average PESQ and STOI comparison of different systems after post-processing across 6 unseen noises at -5dB, 0dB, 5dB

	PESQ			STOI		
System	-5dB	0dB	5dB	-5dB	0dB	5dB
Noisy	1.31	1.60	1.94	0.606	0.731	0.842
PL+PP	1.48	2.00	2.43	0.636	0.776	0.870
PL+L2+PP	**1.59**	**2.09**	**2.50**	**0.654**	**0.789**	**0.879**

As for the centroid constraint, the average PESQ and STOI comparison across 6 unseen noises is shown as Table 2. **PL+L2+PP** refers to the enhancement results of standard PL with $L2$ loss after post-processing. In **PL+L2+PP**, we incorporate two kinds of geometric constraints, namely edge constraint and centroid constraint. The weight factors of edge constraint and centroid constraint are 20 and 1.0, respectively. As centroid constraint is closely related to post-processing, we directly gave the results of the post-processing. Compared with **PL+PP**, **PL+L2+PP** can achieve better results, especially in STOI. For instance, the STOI gain is 0.018 at SNR=-5dB. Thus, according to the analysis above, both the edge constraint and centroid constraint can achieve overall performance improvements in low SNR environments.

4. CONCLUSION

In this study, firstly we have derived a method for constructing intermediate learning target online. More importantly, two geometric-based constraints, namely edge constraints and centroid constraints, are proposed to guide the learning of each target of PL framework more effectively and correctly. The cosine edge constraints mainly keep optimization direction of each target consistent, while the centroid constraints reduce the distance between the learning target and the estimated target globally. Experiments demonstrate that the proposed geometry constrained PL can achieves good PESQ and STOI improvements at low SNR environments.

5. ACKNOWLEDGEMENT

This work was supported in part by the National Key R&D Program of China under contract No. 2017YFB1002202, the National Natural Science Foundation of China under Grants No. 61671422 and U1613211, the Key Science and Technology Project of Anhui Province under Grant No. 17030901005. This work was also funded by Tencent.

6. REFERENCES

[1] Steven Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 27, no. 2, pp. 113–120, 1979.

[2] Jae Soo Lim and Alan V Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.

[3] Yariv Ephraim and David Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE transactions on acoustics, speech, and signal processing*, vol. 33, no. 2, pp. 443–445, 1985.

[4] Israel Cohen and Baruch Berdugo, "Speech enhancement for non-stationary noise environments," *Signal processing*, vol. 81, no. 11, pp. 2403–2418, 2001.

[5] Zixing Zhang, Jürgen Geiger, Jouni Pohjalainen, Amr El-Desoky Mousa, Wenyu Jin, and Björn Schuller, "Deep learning for environmentally robust speech recognition: An overview of recent developments," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 5, pp. 49, 2018.

[6] Johan Rohdin, Anna Silnova, Mireia Diez, Oldřch Plchot, Pavel Matějka, and Lukáš Burget, "End-to-end dnn based speaker recognition inspired by i-vector and plda," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4874–4878.

[7] DeLiang Wang and Jitong Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[8] DeLiang Wang, "On ideal binary mask as the computational goal of auditory scene analysis," in *Speech separation by humans and machines*, pp. 181–197. Springer, 2005.

[9] Arun Narayanan and DeLiang Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7092–7096.

[10] Gibak Kim and Philipos C Loizou, "Improving speech intelligibility in noise using a binary mask that is based on magnitude spectrum constraints," *IEEE Signal Processing Letters*, vol. 17, no. 12, pp. 1010–1013, 2010.

[11] Donald S Williamson, Yuxuan Wang, and DeLiang Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 3, pp. 483–492, 2016.

[12] Hakan Erdogan, John R Hershey, Shinji Watanabe, and Jonathan Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 708–712.

[13] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal processing letters*, vol. 21, no. 1, pp. 65–68, 2014.

[14] Szu Wei Fu, Tao Wei Wang, Tsao Yu, Xugang Lu, and Hisashi Kawai, "End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks," *IEEE/ACM Transactions on Audio Speech & Language Processing*, vol. PP, no. 99, pp. 1–1, 2017.

[15] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2015.

[16] Lei Sun, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "Multiple-target deep learning for lstm-rnn based speech enhancement," in *2017 Hands-free Speech Communications and Microphone Arrays (HSCMA)*. IEEE, 2017, pp. 136–140.

[17] Szu-Wei Fu, Yu Tsao, and Xugang Lu, "Snr-aware convolutional neural network modeling for speech enhancement.," in *Interspeech*, 2016, pp. 3768–3772.

[18] Santiago Pascual, Antonio Bonafonte, and Joan Serra, "Segan: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[19] Tian Gao, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "Snr-based progressive learning of deep neural network for speech enhancement.," in *INTERSPEECH*, 2016, pp. 3713–3717.

[20] Tian Gao, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "Densely connected progressive learning for lstm-based speech enhancement," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5054–5058.

[21] Paul J Werbos et al., "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[22] John Garofalo, David Graff, Doug Paul, and David Pallett, "Csr-i (wsj0) complete," *Linguistic Data Consortium, Philadelphia*, 2007.

[23] G Hu, "100 nonspeech environmental sounds,"[online] available: http://web. cse. ohio-state. edu/pnl/corpus/hunonspeech," *HuCorpus. html*, 2004.

[24] Andrew Varga and Herman JM Steeneken, "Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech communication*, vol. 12, no. 3, pp. 247–251, 1993.

[25] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*. IEEE, 2001, vol. 2, pp. 749–752.

[26] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.