INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

A Maximum Likelihood Approach to SNR-Progressive Learning Using Generalized Gaussian Distribution for LSTM-Based Speech Enhancement

Xiao-Qi Zhang[1], *Jun Du*[1,*], *Li Chai*[1], *Chin-Hui Lee*[2]

[1]University of Science and Technology of China, Hefei, Anhui, P.R.China
[2]Georgia Institute of Technology, Atlanta, GA, USA

xiaoqizhang,cl122@mail.ustc.edu.cn, jundu@ustc.edu.cn, chl@ece.gatech.edu

Abstract

A maximum likelihood (ML) approach to characterizing regression errors in each target layer of SNR progressive learning (PL) using long short-term memory (LSTM) networks is proposed to improve performances of speech enhancement at low SNR levels. Each LSTM layer is guided to learn an intermediate target with a specific SNR gain. In contrast to using previously proposed minimum squared error criterion (MMSE-PL-LSTM) which leads to an un-even distribution and a broad dynamic range of the prediction errors, we model the errors with a generalized Gaussian distribution (GGD) at all intermediate layers in the newly proposed ML-PL-LSTM framework. The shape factors in GGD can be automatically updated when training the LSTM networks in a layer-wise manner to estimate the network parameters progressively. Tested on the CHiME-4 simulation set for speech enhancement in unseen noise conditions, the proposed ML-PL-LSTM approach outperforms MMSE-PL-LSTM in terms of both PESQ and STOI measures. Furthermore, when evaluated on the CHiME-4 real test set for speech recognition, using ML-enhanced speech also results in less word error rates than those obtained with MMSE-enhanced speech.

Index Terms: progressive learning, maximum likelihood, minimum mean squared error, long short-term memory, layer-wise update, generalized Gaussian distribution, speech recognition.

1. Introduction

Speech enhancement has long been an open research issue [1]. In the past few decades, researchers have proposed many algorithms to solve this problem. Traditional speech enhancement algorithms including spectral subtraction (SS) [2], Wiener filtering [3, 4], minimum mean squared error (MMSE) estimation [5, 6] and optimally-modified log-spectral amplitude (OMLSA) speech estimation [7] have been extensively studied in the past. However, the challenges of speech enhancement in tracking non-stationary noises for real-world scenarios still remain [8].

In recent years, speech enhancement has shown considerable success with the rapid development of deep learning [9]. Researchers have conducted in-depth discussions on the aspects of learning targets, input features, and deep neural network (DNN) structures. In general, there are two categories of speech enhancement approaches in terms of learning targets, namely: mapping-based and masking-based methods. The former maps noisy to clean speech features [10] directly, while the latter learns a time-frequency(T-F) mask [11, 12] like ideal binary mask (IBM) from a noisy signal and obtains the enhanced features from the estimated mask. As for the input, there are many types of features that have been adopted in speech enhancement. Log-power spectra (LPS) [13], mel-frequency cepstral coefficient (MFCC) [14] are two popular choices. Moreover, many types of deep neural architectures have been utilized in speech enhancement, including feed-forward DNNs [15], convolutional neural networks (CNNs) [16, 17], recurrent neural networks (RNNs) [18, 19] and generative adversarial networks (GANs) [20].

Nonetheless, there are still many challenges in low SNR environments. A densely connected progressive learning for LSTM-based speech enhancement, called PL-LSTM, has been proposed in [21]. The main idea is to decompose the direct mapping from noisy to clean speech into multiple stages by progressively guiding each hidden layer of the neural networks to learn an intermediate target with a specific SNR gain explicitly. With a reduction of learning difficulty in a step-by-step manner, this framework has demonstrated superiority over the conventional DNN structures. Moreover, the PL framework is also widely used as the front-end of ASR. According to [22], the PL framework with intermediate outputs can directly improve the ASR performance, while the conventional speech enhancement approaches with good noise reduction even underperform the unprocessed noisy speech. Because the popular recognition system under multi-condition training is robust to noise, speech preservation is as important as noise suppression, and the intermediate outputs of the PL framework can make a tradeoff between them.

However, a key point of the PL framework is the intermediate targets are noisy speeches with different levels of noise energy. On the one hand, the distributions of the prediction errors are affected by the random noise. On the other hand, the prediction error values vary among different target layers. Therefore, it is difficult to fully use prediction error values of different target layers when training the network using MMSE criterion, which leads to over-smoothing.

In this paper, we propose an ML-PL-LSTM framework to address the abovementioned issues. Considering the statistical properties of noise, we apply the ML criterion [23] to the PL framework in order to better characterize the prediction errors at the outputs of all target layers. We adopt the GGD to characterize the error distributions. The scale factors in GGD are updated in every epoch to ensure better compliance with the distribution. Then we can derive and maximize the conditional likelihood function for the whole network to optimally estimate the parameters. Furthermore, an ML training procedure in a layer-wise manner is adopted to update the network parameters step by step. Two types of test sets, simulated and real, are used in our experiments. The experiments on the simulation test set are designed to evaluate the performance on listening objective metrics. The experiments on the CHiME-4 real test set are used to evaluate the ASR performance in real-world scenarios. The experimental results show that the proposed ML-PL-LSTM method can achieve significant improvements in all metrics compared with the MMSE-PL-LSTM.

* corresponding author

http://dx.doi.org/10.21437/Interspeech.2021-922

2. The Proposed ML-PL-LSTM

2.1. Review of MMSE-PL-LSTM

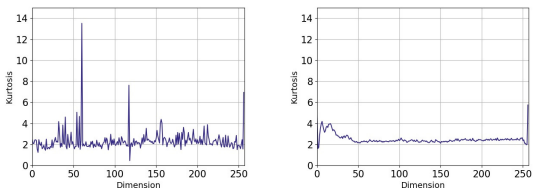
In our prior work [21], an LSTM-based densely connected progressive learning framework has been proposed to improve the performance in low SNR environments. The method decomposes the mapping from noisy to clean speech into multiple stages with SNR increasing progressively. And all the target layers are designed to learn intermediate speech features with higher SNRs or clean speech. Moreover, in order to alleviate the information loss caused by the increasing number of intermediate targets, the input and the estimations of intermediate targets are spliced together to learn the next target, forming a densely connected architecture. As for optimization procedure, a weighted MMSE criterion is designed to optimize all network parameters randomly initialized with K target layers in the manner of back propagation through time (BPTT) with gradient descent [24], and the loss function is as follow:

$$E = \sum_{k=1}^{K} \eta_k E_k \tag{1}$$

$$E_k = \frac{1}{N} \sum_{n=1}^{N} \|\mathcal{F}_k(\boldsymbol{y}_n, \hat{\boldsymbol{x}}_{n,1}, ..., \hat{\boldsymbol{x}}_{n,k-1}, \boldsymbol{W}_k) - \boldsymbol{x}_{n,k}\|_2^2 \tag{2}$$

where $\hat{\boldsymbol{x}}_{n,k}$ and $\boldsymbol{x}_{n,k}$ are the n-th D-dimensional vectors of estimated and reference target LPS feature vectors for k-th target layer, respectively, with N representing the mini-batch size. $\boldsymbol{y}_n$ is the n-th D-dimensional vector of input noisy LPS. $\boldsymbol{W}_k$ represents the LSTM parameters set of the weight matrices and bias vectors before k-th target layer. $\mathcal{F}_k(\boldsymbol{y}_n, \hat{\boldsymbol{x}}_{n,1}, ..., \hat{\boldsymbol{x}}_{n,k-1}, \boldsymbol{W}_k)$ is the neural network function for k-th target with the dense structure using the previously learned intermediate targets from $\boldsymbol{y}_n$ to $\hat{\boldsymbol{x}}_{n,k-1}$. K is the number of target layers. η_k is the weighting factor of the k-th target.

2.2. Motivation



(a) Target 1 (10dB SNR gain) (b) Target 3 (clean)

Figure 1: *(a) and (b) refer to the kurtosis curves of each dimension of the prediction error vectors of target layer 1 and 3 from the well-trained MMSE-PL-LSTM on the cross-validation set.*

Fig. 1 (a) and (b) show the kurtosis curves of each dimension of the prediction error vectors of target layer 1 and 3 from a well-trained MMSE-PL-LSTM model configured with three targets and 10dB SNR gains for each of the two intermediate targets. Compared with target layer 3, which takes clean speech as the learning goal, the kurtosis values of the prediction errors at target layer 1 fluctuate dramatically. It indicates that the prediction errors of the intermediate target layers (e.g., target layer 1) become more random due to the introduction of noise, which inspires us to adopt a general probabilistic framework to model the prediction errors of different target layers, so as to fully use them during training. Based on the above considerations, we designed the ML-PL-LSTM as shown in Fig. 2.

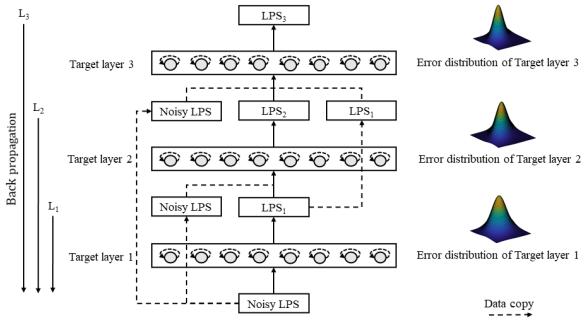


Figure 2: *The ML-PL-LSTM architecture for speech enhancement*

2.3. Derivation for ML-PL-LSTM

In ML-PL-LSTM, we redefine the objective function under the probabilistic framework to fully utilize the information of multiple target layers. In order to describe the special circumstances on the distribution due to the noise, we build the GGD error model for each target layer and assume the estimated output vectors of different target layers are independent, then we can get the joint prediction error distribution of multiple target layers with exponential weighting for each target layer, which is equivalent to the joint conditional distribution:

$$p(\boldsymbol{x}_n|\boldsymbol{y}_n, \boldsymbol{W}, \boldsymbol{\Theta}) = \prod_{k=1}^{K} [p(\boldsymbol{x}_{n,k}|\boldsymbol{y}_n, \boldsymbol{W_k}, \boldsymbol{\Theta}_k)]^{\eta_k} \tag{3}$$

where $\boldsymbol{x}_n = \{\boldsymbol{x}_{n,k}|k = 1, 2, ..., K\}$, $\boldsymbol{\Theta}_k$ represents the parameter set of error distribution at the k-th target layer. Given a set with N data pairs $(\boldsymbol{Y}, \boldsymbol{X}) = \{(\boldsymbol{y}_n, \boldsymbol{x}_n)|n = 1, 2, ..., N\}$ and assuming that they are drawn independently from the distribution in Eq. (3), the conditional likelihood function of the whole network can be expressed as:

$$p(\boldsymbol{X}|\boldsymbol{Y}, \boldsymbol{W}, \boldsymbol{\Theta}) = \prod_{n=1}^{N} \prod_{k=1}^{K} [p(\boldsymbol{x}_{n,k}|\boldsymbol{y}_n, \boldsymbol{W_k}, \boldsymbol{\Theta}_k)]^{\eta_k} \tag{4}$$

The corresponding log-likelihood function is shown as follows:

$$\ln p(\boldsymbol{X}|\boldsymbol{Y}, \boldsymbol{W}, \boldsymbol{\Theta}) = \sum_{k=1}^{K} \eta_k L_k \tag{5}$$

$$L_k = \sum_{n=1}^{N} \ln p(\boldsymbol{x}_{n,k}|\boldsymbol{y}_n, \boldsymbol{W_k}, \boldsymbol{\Theta}_k) \tag{6}$$

where L_k is the log-likelihood function of the k-th target layer.

The above is the derivation of the new objective function on the whole network. Next, we will focus on the k-th target layer and analyze the conditional distribution of the k-th target layer.

Similar to [23], the assumption is made that each dimension of the prediction error vector at each target layer is characterized as a univariate GGD with a zero mean, unknown scale factor

$\alpha_{k,d}$ and shape factor $\beta_{k,d}$:

$$p(e_{n,k,d}|\alpha_{k,d},\beta_{k,d}) = \frac{\beta_{k,d}}{2\alpha_{k,d}\Gamma(\frac{1}{\beta_{k,d}})}\exp\left(-\left(\frac{|e_{n,k,d}|}{\alpha_{k,d}}\right)^{\beta_{k,d}}\right) \tag{7}$$

One thing that should be noticed is the shape factors of GGD are artificially set and fixed during training in our previous work. However, it can be observed from Fig. 1 that the kurtosis values of the target layer 1 in all dimensions are not stable. In addition, when there are multiple targets, it is not easy to artificially adjust the values of shape factors for different target layers to find the best combination method. Therefore, we adopt the method in [25], which updates shape factors automatically through the one-to-one mapping between shape factor and the kurtosis in GGD:

$$\text{Kurt}[e_{k,d}] = \text{E}\left[\left(\frac{e_{k,d}-\mu_{k,d}}{\sigma_{k,d}}\right)^4\right] = \frac{\text{E}[(e_{k,d}-\mu_{k,d})^4]}{(\text{E}[(e_{k,d}-\mu_{k,d})^2])^2} \tag{8}$$

$$\text{Kurt}[e_{k,d}] = \frac{\Gamma(5/\beta_{k,d})\Gamma(1/\beta_{k,d})}{\Gamma(3/\beta_{k,d})^2} - 3 \tag{9}$$

where $e_{k,d}$ is the d-th dimension of prediction error vector at the k-th target layer. $\mu_{k,d}$ and $\sigma_{k,d}$ are the mean and standard deviation of $e_{k,d}$. In the training process, we calculate the kurtosis of each dimension in the prediction error vector by Eq. (8), then we get the value of the new shape factor by looking up the table calculated by Eq. (9).

Under the assumption that distribution in each dimension is mutually independent, the joint distribution of all dimensions on the k-th target layer at sample index n is as follows:

$$\begin{aligned} &p(\boldsymbol{x}_{n,k}|\boldsymbol{y}_n,\boldsymbol{W}_k,\boldsymbol{\alpha}_k,\boldsymbol{\beta}_k) \\ &= \prod_{d=1}^{D}\frac{\beta_{k,d}}{2\alpha_{k,d}\Gamma(\frac{1}{\beta_{k,d}})}\exp\left(-\left(\frac{|x_{n,k,d}-\hat{x}_{n,k,d}|}{\alpha_{k,d}}\right)^{\beta_{k,d}}\right) \end{aligned} \tag{10}$$

where $\boldsymbol{\beta}_k = \{\beta_{k,d}|d = 1,2,...,D\}$, $\boldsymbol{\alpha}_k = \{\alpha_{k,d}|d = 1,2,...,D\}$. Finally, based on Eq. (10), the log-likelihood function in Eq. (5) is specified as:

$$\begin{aligned} \ln p(\boldsymbol{X}|\boldsymbol{Y},\boldsymbol{W},\boldsymbol{\alpha},\boldsymbol{\beta}) = &\sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{d=1}^{D}\eta_k \times \ln\left(\frac{\beta_{\text{k,d}}}{2\alpha_{\text{k,d}}\Gamma(\frac{1}{\beta_{\text{k,d}}})}\right) \\ &-\sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{d=1}^{D}\eta_k \times \left(\frac{|x_{n,k,d}-\hat{x}_{n,k,d}|}{\alpha_{k,d}}\right)^{\beta_{k,d}} \end{aligned} \tag{11}$$

In the training phase, we maximize Eq. (11) with an alternate two-step updating algorithm. First, a closed solution of $\boldsymbol{\alpha}$ can be obtained by fixing $\boldsymbol{W}$:

$$\alpha_{k,d} = \left(\frac{\beta_{k,d}}{N}\sum_{n=1}^{N}|x_{n,k,d}-\hat{x}_{n,k,d}|^{\beta_{k,d}}\right)^{\frac{1}{\beta_{k,d}}} \tag{12}$$

Second, by fixing $\boldsymbol{\alpha}$, $\boldsymbol{W}$ can be optimized by the manner of BPTT with gradient descent.

Moreover, we update the parameters in a layer-wise manner, which divides the parameter updating process into several steps. The total number of steps is consistent with the number of targets (K). At the s-th step ($1 \le s \le K$), the parameters before the s-th layer are updated by maximizing the weighted log-likelihood function of the layers before the s-th layer, namely using Eq. (5) with the replacement of K by s.

3. Experiments

3.1. Experimental conditions

Four noise types (BUS, CAF, PED, STR) in the CHiME-4 challenge were selected as our noise database. 7138 utterances came from the WSJ0 corpus [26] were used as the clean speech, which were corrupted with the above-mentioned noise types at three levels of SNRs (-5dB, 0dB, 5dB) to build 36-hour noisy speech training set. 500 utterances randomly selected from the training set were used to construct the cross-validation set.

Two types of test sets, simulated and real, were used in our experiments. The simulation test set was constructed by mixing 330 clean speech utterances from the CHiME-4 simulation test set with 15 unseen noise types from the NOISEX-92 [27] corpus at three SNR levels (-5dB, 0dB, 5dB). The experiments on the simulation test set were designed to evaluate the intelligibility and quality of enhanced speech by objective metrics including perceptual evaluation of speech quality (PESQ) [28] and short-time objective intelligibility (STOI) [29]. The real test set was offered by the CHiME-4 challenge, and we used it to evaluate the recognition performance of different enhancement methods in real-world scenarios.

The waveforms were sampled at 16kHz. We set the frame length to 512 samples and the frame shift to 256 samples. The 257-dimensional LPS features normalized by mean and variance were used to train neural networks. The configurations of the basic PL framework were fixed at 3 target layers and 1048 cells for each LSTM layer. The SNR gain between the target layers was set to 10dB, and the last target was clean speech. The training process was divided into three steps and each step contained 10 epochs. The total number of epochs was 30. The learning rate for fine-tuning was set to 0.001 and declined at a rate of 80% when the epoch was 6, 12, 16, 24 with the batch size of 8. The shape factors of ML-PL-LSTM were initialized from the well-trained MMSE-PL-LSTM and updated for each epoch. The weighting factor of each target layer was set to 1. The ASR system was provided in [30]. The acoustic model is a DNN-HMM discriminatively trained with the sMBR criterion [31]. The language models are 5-gram with Kneser-Ney (KN) smoothing [32] for the first-pass decoding and the simple RNN-based language model [33, 34, 35] for rescoring.

3.2. Evaluations on speech enhancement

Table 1 lists the average PESQ and STOI results of different systems across the unseen noise types at -5dB, 0dB, 5dB. "T1", "T2", and "T3" denote using the outputs of target layer 1, 2, and 3 for enhancement, respectively. "PP" refers to the enhancement results after post-processing, which is to average the estimations of multiple targets [21]. Compared with MMSE-PL-LSTM, the results of ML-PL-LSTM can achieve remarkable improvements on all target layers, especially in STOI. For instance, the STOI increases from 0.737 to 0.776 at target layer 3 and from 0.771 to 0.792 at target layer 2 in average. If we take all SNR conditions into consideration, we can observe a more significant improvement in low SNR environments.

Another interesting observation is for both MMSE-PL-LSTM and ML-PL-LSTM, T1 and T2 usually generate better results than T3. The reason is that in low SNR environments, aggressive denoising (e.g., T3) can lead to large speech distortions, especially for unseen noise types. By using the ML-PL-LSTM, we decrease the performance gap among T1, T2, and T3 for both STOI and PESQ metrics, e.g., the STOI gap of 0.034 between MMSE-PL-LSTM-T3 and MMSE-PL-LSTM-

T2, while the STOI gap of 0.016 between ML-PL-LSTM-T3 and ML-PL-LSTM-T2 in average.

Moreover, post-processing is used in the testing stage to make full use of the information of different learning targets when there are multiple targets in the PL framework. Under our experiment configurations that the basic PL framework was fixed at 3 target layers, the superiority of post-processing methods can not be reflected in the MMSE-PL-LSTM. However, when using the ML-PL-LSTM, the post-processing can achieve comparable or better results than any single target layer output, which implies there exists a stronger complementarity among different targets in ML framework.

Fig. 3 shows the spectrogram comparison between MMSE-PL-LSTM and ML-PL-LSTM with different target outputs. Compared with ML-PL-LSTM, there exist large speech distortions and even removal of some speech segments in MMSE-PL-LSTM as shown in the blue box areas of Fig. 3, which illustrates that the ML-PL-LSTM method can alleviate the over-smoothing problem caused by the inability to balance the differences among multiple layers in MMSE-PL-LSTM.

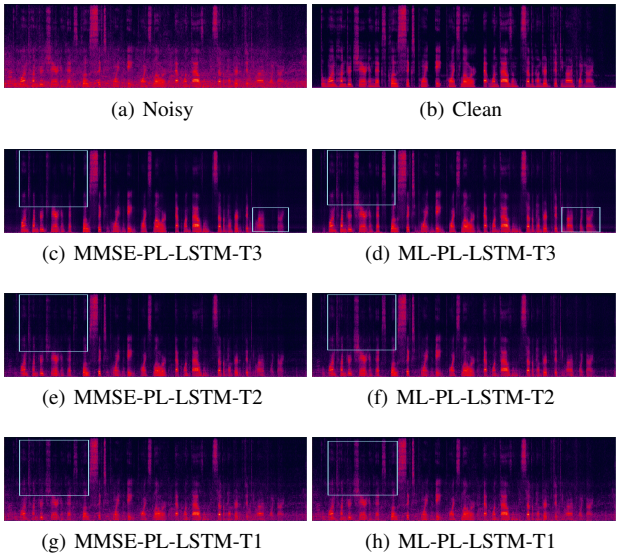
Table 1: *The average PESQ and STOI results of different systems across the unseen noise types at -5dB, 0dB, 5dB.*

	SNR		-5	0	5	AVE
STOI	Noisy		0.648	0.752	0.846	0.749
	MMSE-PL-LSTM	T1	0.686	0.791	0.871	0.783
		T2	0.67	0.781	0.861	0.771
		T3	0.613	0.75	0.848	0.737
		PP	0.665	0.784	0.87	0.773
	ML-PL-LSTM	T1	0.696	0.801	0.879	0.792
		T2	0.693	0.804	0.878	0.792
		T3	0.659	0.792	0.876	0.776
		PP	0.693	0.81	0.888	0.797
PESQ	SNR		-5	0	5	AVE
	Noisy		1.448	1.744	2.073	1.755
	MMSE-PL-LSTM	T1	1.673	2.039	2.407	2.040
		T2	1.834	2.265	2.636	2.245
		T3	1.545	2.016	2.487	2.016
		PP	1.8	2.231	2.624	2.218
	ML-PL-LSTM	T1	1.696	2.07	2.434	2.067
		T2	1.883	2.319	2.683	2.295
		T3	1.698	2.219	2.653	2.190
		PP	1.869	2.317	2.689	2.292

3.3. Evaluations on speech recognition after enhancement

In Section 3.2, we only evaluated our framework on the simulation test set by using objective metrics. Considering the superiority of the PL framework in ASR, in this section, we conducted the speech recognition experiments on the real test set to explore the ASR performance of our proposed approach in realistic scenarios.

Table 2 shows word error rate (WER) comparison between the ML-PL-LSTM and the MMSE-PL-LSTM on the real test set across four environments of CHiME-4 test set. First, compared with MMSE-PL-LSTM, the ML-PL-LSTM can achieve remarkable WER reductions on the outputs of all target layers in all situations. Second, MMSE-PL-LSTM-T1 and ML-PL-LSTM-T1, the intermediate target with +10 dB SNR gain, improve the ASR performance comparing to Noisy, and our proposed ML-PL-LSTM can achieve further improvement, e.g., the relative WER reduction of 4.66% and 11.28% from Noisy to MMSE-PL-LSTM-T1 and ML-PL-LSTM-T1 in average. Finally, we observe there is a significant ASR performance difference among the different target outputs of MMSE-PL-LSTM, while the ML-PL-LSTM can decrease this gap, which implies



(a) Noisy (b) Clean

(c) MMSE-PL-LSTM-T3 (d) ML-PL-LSTM-T3

(e) MMSE-PL-LSTM-T2 (f) ML-PL-LSTM-T2

(g) MMSE-PL-LSTM-T1 (h) ML-PL-LSTM-T1

Figure 3: *The spectrogram comparison between MMSE-PL-LSTM and ML-PL-LSTM with different target outputs. Test on speech babble noise at SNR = 5dB.*

that our proposed method can effectively alleviate the instability of different target outputs in ASR performance by controlling dynamic range of prediction error values.

Table 2: *The WER(%) comparison between the ML-PL-LSTM and the MMSE-PL-LSTM on the real test set.*

		BUS	CAF	PED	STR	AVE
Noisy		36.55	24.73	19.92	14.16	23.84
MMSE-PL-LSTM	T1	39.32	22.51	16.87	12.25	22.73
	T2	47.39	30.95	23.73	16.03	29.52
	T3	55.39	49.33	37.57	21.24	40.88
ML-PL-LSTM	T1	35.74	20.64	16.33	11.88	21.15
	T2	40.49	29.7	22.4	14.05	26.66
	T3	46.63	45.95	34.36	19.33	36.56

4. Conclusion

In this paper, we propose an ML-PL-LSTM approach to speech enhancement by modeling the regression errors of the outputs at the intermediate target layers in the SNR progressive learning framework. Compared with MMSE-PL-LSTM, the ML-PL-LSTM alleviates the problems of randomness in the prediction error and reduces the dynamic range of prediction errors among multiple target layers. To verify the generalization ability of the proposed method, we conducted experiments on the CHiME-4 simulation and real test sets. The experimental results demonstrate that the proposed ML-PL-LSTM framework achieves improved speech quality in PESQ and STOI and reduces word error rates. In the future, we will explore powerful deep architectures for ML-PL-LSTM and its joint optimization with the ASR back-end.

5. Acknowledgements

This work was supported by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDC08050200.

6. References

[1] J. Benesty, S. Makino, and J. Chen, *Speech enhancement*. Springer Science & Business Media, 2005.

[2] S. F. Boll, "Supression of acoustic noise in speech using spectral subtraction, ieee transaction on assp, vol," *ASSP-2 No*, vol. 2, 1979.

[3] J. Lim and A. Oppenheim, "All-pole modeling of degraded speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 3, pp. 197–210, 1978.

[4] Lim, S. J., Oppenheim, and V. A., "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, 1979.

[5] Y. Ephraim and D. 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.

[6] T. Gerkmann and R. C. Hendriks, "Unbiased mmse-based noise power estimation with low complexity and low tracking delay," *IEEE Transactions on Audio Speech & Language Processing*, vol. 20, no. 4, pp. 1383–1393, 2012.

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

[8] P. C. Loizou and G. Kim, "Reasons why current speech-enhancement algorithms do not improve speech intelligibility and suggested solutions," *IEEE transactions on audio, speech, and language processing*, vol. 19, no. 1, pp. 47–56, 2010.

[9] Y. Bengio, *Learning deep architectures for AI*. Now Publishers Inc, 2009.

[10] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 1, pp. 7–19, 2015.

[11] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

[12] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1381–1390, 2013.

[13] J. Du and Q. Huo, "A speech enhancement approach using piecewise linear approximation of an explicit model of environmental distortions," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[14] Y. Wang, K. Han, and D. Wang, "Exploring monaural features for classification-based speech segregation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 2, pp. 270–279, 2012.

[15] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal processing letters*, vol. 21, no. 1, pp. 65–68, 2013.

[16] A. Pandey and D. Wang, "Tcnn: Temporal convolutional neural network for real-time speech enhancement in the time domain," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6875–6879.

[17] S.-W. Fu, Y. Tsao, and X. Lu, "Snr-aware convolutional neural network modeling for speech enhancement." in *Interspeech*, 2016, pp. 3768–3772.

[18] M. Liu, Y. Wang, J. Wang, J. Wang, and X. Xie, "Speech enhancement method based on lstm neural network for speech recognition," in *2018 14th IEEE International Conference on Signal Processing (ICSP)*. IEEE, 2018, pp. 245–249.

[19] C. Zheng, X. Zhang, M. Sun, Y. Xing, and H. Shi, "Throat microphone speech enhancement via progressive learning of spectral mapping based on lstm-rnn," in *2018 IEEE 18th International Conference on Communication Technology (ICCT)*. IEEE, 2018, pp. 1002–1006.

[20] P. S. Nidadavolu, J. Villalba, and N. Dehak, "Cycle-gans for domain adaptation of acoustic features for speaker recognition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6206–6210.

[21] T. Gao, J. Du, L.-R. Dai, and C.-H. 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.

[22] Y.-H. Tu, J. Du, T. Gao, and C.-H. Lee, "A multi-target snr-progressive learning approach to regression based speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1608–1619, 2020.

[23] L. Chai, J. Du, Q.-F. Liu, and C.-H. Lee, "Using generalized gaussian distributions to improve regression error modeling for deep learning-based speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 1919–1931, 2019.

[24] Werbos and J. P., "Backpropagation through time: what it does and how to do it," *Proc IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[25] S.-T. Niu, J. Du, L. Chai, and C.-H. Lee, "A maximum likelihood approach to multi-objective learning using generalized gaussian distributions for dnn-based speech enhancement," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6229–6233.

[26] J. Garofalo, D. Graff, D. Paul, and D. Pallett, "Csr-i (wsj0) complete," *Linguistic Data Consortium, Philadelphia*, 2007.

[27] A. Varga and H. J. 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.

[28] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. 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)*, vol. 2. IEEE, 2001, pp. 749–752.

[29] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. 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.

[30] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech & Language*, vol. 46, pp. 535–557, 2017.

[31] K. Veselỳ, M. Hannemann, and L. Burget, "Semi-supervised training of deep neural networks," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 267–272.

[32] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *1995 international conference on acoustics, speech, and signal processing*, vol. 1. IEEE, 1995, pp. 181–184.

[33] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[34] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 1562–1566.

[35] P. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2136–2147, 2015.