Stochastic DNN-HMM Training for Robust ASR

Kang Hyun Lee, Woo Hyun Kang, Hyeonseung Lee, and Nam Soo Kim
Department of Electrical and Computer Engineering and INMC,
Seoul National University, Seoul, Korea
E-mail: {khlee, whkang, hslee}@hi.snu.ac.kr, nkim@snu.ac.kr Tel/Fax: +82-2-884-1824

***Abstract*—Since the introduction of deep neural network (DNN)-based acoustic model to automatic speech recognition (ASR), robust ASR using DNN are being in research. However, most DNN-based techniques are performed without consideration of the reliability of the estimates and this degrades the ASR performance especially in the training-test mismatch conditions. In this paper, we propose a novel deep learning-based acoustic modeling technique which measures and takes account of the reliability using a single DNN. The proposed approach describes the mapping between the noisy input and clean features as a stochastic process. Therefore, a statistical modeling is applied to the DNN-based acoustic model in predicting the posterior distribution of the clean speech features given a distorted input data. Also, by attempting the two different probabilistic models in clean feature distribution assumption, we investigate which distribution is more proper on various environment conditions. It has been shown that the proposed technique outperforms the conventional DNN-based techniques on Aurora-4 DB and mismatched noise conditions.**

I. Introduction

In recent years, deep learning has been prevalent in signal processing and it has become an opportunity for automatic speech recognition (ASR) to progress. Especially in acoustic modeling, introduction of the deep neural network (DNN)-hidden Markov model (HMM) system which represents the relationship between the observed acoustic features and HMM states using DNN instead of Gaussian mixture model (GMM) is considered as a breakthrough [1], [2], [3]. This is attributed to the DNN's capability in automatically learning complicated non-linear mapping from the input to the target vectors. If a sufficient amount of training data is available, more complicated input-target relationship can be easily learned by using wider and deeper neural network architectures [4].

Interest in the efficient learning capability of DNN has also been expanded to the robust ASR. Traditionally, the approaches to noise robustness can be divided into two categories: front-end (FE) and back-end (BE) techniques. The goal of FE techniques is to compensate the effect of distortions on the observed input features. While the conventional FE techniques [5], [6], [7], [8] are based on some specific models or formulations to account for the complicated corruption process from the clean to distorted speech features, the DNN-based FE techniques [9], [10], [11], [12] let the networks directly map the corresponding clean targets from the distorted inputs.

Meanwhile, BE methods modify the acoustic models to match the incoming input features better. In BE technique training, it is important to make the acoustic model parameters take the environmental characteristic into account effectively. Among various DNN-based BE researches, adaptation techniques employing auxiliary features with acoustic context information have shown remarkable performances due to its easy implementation and performance [13], [14], [15]. When both FE and BE techniques are implemented by deep learning-based models, the ASR system performance can be enhanced further via joint optimization technique which concatenates the separated two networks together and fine-tuning with a single objective function (e.g., cross-entropy) [16], [17], [18], [19].

Meanwhile, in spite of the performance, the aforementioned DNN-based techniques still have an important weakness. The estimation of clean features or the phonetic targets from the aforementioned DNN-based techniques is performed in a point-wise manner, i.e., the DNN mapping from the input to the corresponding target is described as a deterministic process. However, in a realistic scenario, the test data sometimes contains unseen corruption sources in the training data and the DNN estimator cannot consider these new type of distortion patterns. Eventually, the accuracy of the estimator decreases and this degrades the overall performance of the ASR system. A promising way to compensate this problem may be to extract some information relevant to the reliability of the estimated target and then to apply this to the decoding process.

In this paper, a novel approach to DNN-based acoustic modeling which can be a solution to the aforementioned reliability issue is proposed. The proposed approach describes the relationship between the noisy input and clean features as not a deterministic but a stochastic process. We assume that the clean speech features given noisy input features follow a specific probabilistic model. Then, the parametric information of the probabilistic model is estimated via DNN mapping and directly employed to the DNN-based HMM state prediction. Therefore, we design a DNN-based acoustic model employing the parameters of estimated clean feature distributions.

In order to apply statistical model to DNN-based acoustic model, two different versions of interpretations on the reconstructed clean feature distribution are proposed. Although Gaussian or GMM are most frequently used in conventional speech signal modeling, we cannot be certain whether this is the optimal approach especially to the DNN-based acoustic models. For this reason, another well-known probabilistic model, Laplacian is adopted as the alternative to Gaussian. By employing the different acoustic modeling approaches based on two different probabilistic models, we can investigate where distribution is more proper on various environment conditions.

The performance of the proposed approach is evaluated on the Aurora-4 DB and also in some mismatched noise conditions, and the better performance was observed compared to the conventional DNN-based acoustic modeling techniques.

The organization of the paper is as follows: we first briefly review prior works on DNN-based techniques for robust ASR in Section II. The proposed technique is introduced in Section III. The experiments and results are given in Section IV. Finally, Section V concludes the paper.

II. Prior works on DNN-based technique for robust ASR

Let us denote an observed noisy feature extracted at the t-th frame, the corresponding unknown clean feature and the HMM state identity as $\mathbf{y}_t$, $\mathbf{x}_t$ and $\mathbf{q}_t$, respectively. Additionally, we define $\mathbf{x}_{m_1:m_2}$ as a subsequence of vectors $[\mathbf{x}'_{m_1}\mathbf{x}'_{m_1+1}\cdots\mathbf{x}'_{m_2}]'$ with the prime representing matrix or vector transpose.

Under the general framework of HMM-based recognition, we assume that there exists an unknown underlying function that approximates the posterior probabilities of the HMM states given as follows:

$$p(\mathbf{q}_t|\mathbf{y}_t) \cong f(\mathbf{y}_{t-\tau:t+\tau}) \tag{1}$$

where $f(\cdot)$ represents the function that maps the noisy and noise features to the corresponding HMM state identity which contains phonetic information and the subscript τ represents the temporal coverage which is required for figuring out the contextual information of the speech signal. In the multi-condition DNN-HMM which is the most basic DNN-based BE technique [13], the function $f(\cdot)$ is directly learned based on a collection of noisy data using DNN.

On the other hand, the DNN-based FE techniques map the noisy features into the corresponding clean features via a DNN and the obtained clean feature estimates are fed to the acoustic model. This can be described as

$$p(\mathbf{q}_t|\mathbf{y}_t) \cong p(\mathbf{q}_t|g(\mathbf{y}_{t-\tau:t+\tau})) \tag{2}$$

where the output of $g(\cdot)$ is a stream of clean feature estimates,

$$\hat{\mathbf{x}}_{t-\tau:t+\tau} = g(\mathbf{y}_{t-\tau:t+\tau}). \tag{3}$$

In (2) and (3), $g(\cdot)$ is a DNN dealing with the mapping from the noisy to the clean speech features. Furthermore, DNN-based FE and BE techniques can be employed together as following formulation.

$$p(\mathbf{q}_t|\mathbf{y}_t) \cong h \circ g(\mathbf{y}_{t-\tau:t+\tau}), \tag{4}$$

and

$$p(\mathbf{q}_t|\mathbf{y}_t) \cong h(\hat{\mathbf{x}}_{t-\tau:t+\tau}). \tag{5}$$

Here, $h(\cdot)$ represents a DNN predicting the phonetic target based on the clean speech feature stream. This combination of DNN-based FE and BE techniques can be further improved through joint training [16], [17], [18], [19].

Meanwhile, the DNN-based implementation of $g(\cdot)$ is usually performed by minimizing the mean squared error (MSE) function which is given by

$$J_{MSE} = \frac{1}{T}\sum_{t=1}^{T} ||\mathbf{x}_t - \hat{\mathbf{x}}_t||^2 \tag{6}$$

where T denotes the number of training samples. Since DNN-based FE technique reconstructs clean features under the assumption that the relationship between noisy input and clean target follows a deterministic process, we call this FE network as deterministic network (DE).

However, despite their success in robust ASR, the performance of these approaches usually degrades when there exist some mismatches between the training and test data. While the training data set is limited to rather narrow environments, the test data may undergo distortions not observed in the training data.

III. Proposed technique

In order to supplement the problem in Section II, the proposed DNN is constructed by concatenating two individually fine-tuned DNNs and training the unified DNN jointly as shown in Fig. 1. The first DNN is applied to estimate the statistical parameters. We call this DNN the stochastic network (SN) since it extracts stochastic information of the probabilistic model. The second DNN which is called the prediction network (PN), deals with modeling the relationship between the output of SN and the phonetic target.

A. Stochastic Network

SN estimates the parametric information of the clean feature distribution given the noisy input feature $p(\mathbf{x}_t|\mathbf{y}_t)$. In order to accomplish this, SN is trained to maximize the likelihood of the estimated clean distribution. Therefore, the objective function of SN J_{SN} can be formulated as follows:

$$J_{SN} = \frac{1}{T}\sum_{t=1}^{T}\log p(\mathbf{x}_t|\mathbf{y}_t). \tag{7}$$

Here, J_{SN} can be different depending on the distribution assumption with respect to $p(\mathbf{x}_t|\mathbf{y}_t)$. In this paper, two different versions of SN are proposed: Gaussian stochastic network (GSN) and Laplacian stochastic network (LSN).

In training GSN, $p(\mathbf{x}_t|\mathbf{y}_t)$ is given by Gaussian pdfs where each component of $\mathbf{x}_t$ is uncorrelated as in (8).

$$p(\mathbf{x}_t|\mathbf{y}_t) = \mathcal{N}(\mathbf{x}_t; \mu_{\mathbf{x}_t}(\mathbf{y}_t), \Sigma_{\mathbf{x}_t}(\mathbf{y}_t)) \tag{8}$$

with

$$\Sigma_{\mathbf{x}_t} = \begin{bmatrix} \sigma^2_{\mathbf{x}_{t,1}} & 0 & \dots & 0 \\ 0 & \sigma^2_{\mathbf{x}_{t,2}} & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \sigma^2_{\mathbf{x}_{t,D_\mathbf{x}}} \end{bmatrix}. \tag{9}$$

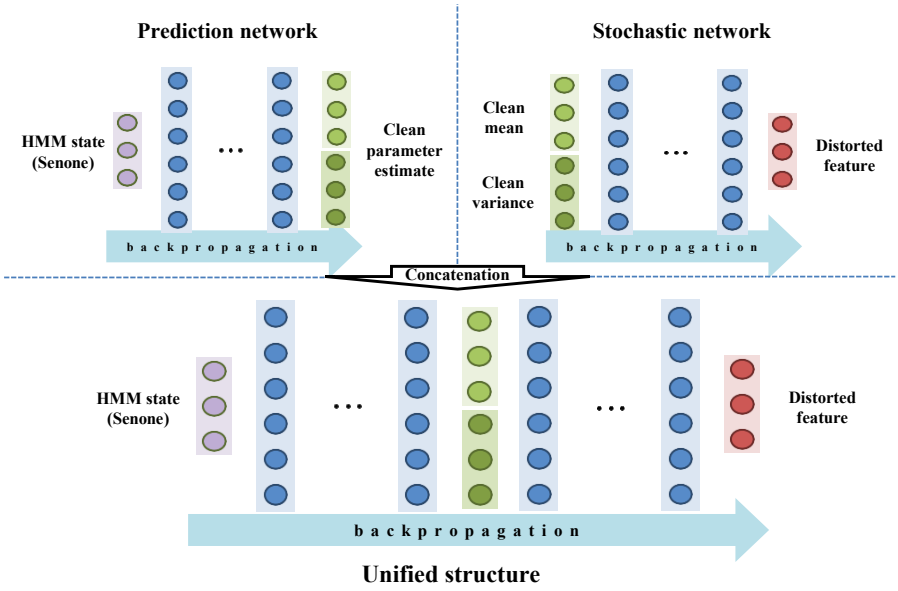

Fig. 1. The structure of stochastic network.

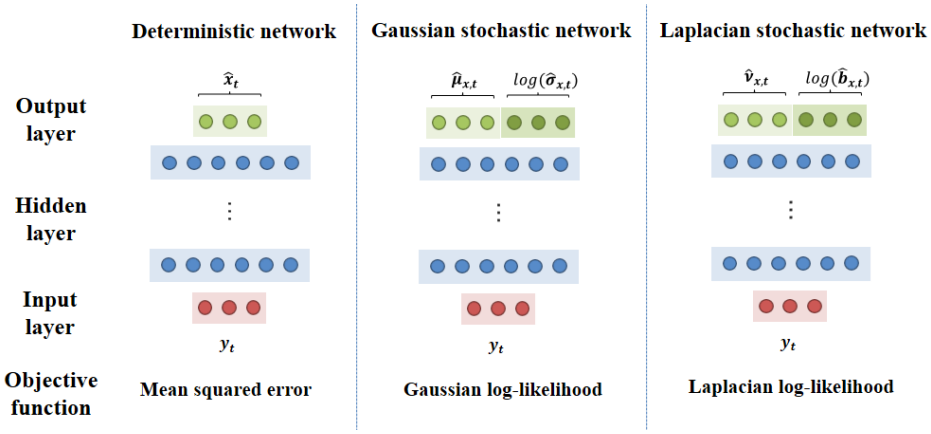

Fig. 2. The structures of deterministic, Gaussian and Laplacian stochastic networks.

Therefore, the objective function of Gaussian stochastic network (GSN) J_{GSN} is as follows:

$$J_{GSN} = \frac{1}{T}\sum_{t=1}^{T}\sum_{d=1}^{D_{\mathbf{x}}} -\log\left(\sigma_{\widehat{\mathbf{x}}_{t,d}}\sqrt{2\pi}\right) - \frac{(\mathbf{x}_{t,d} - \mu_{\widehat{\mathbf{x}}_{t,d}})^2}{2{\sigma_{\widehat{\mathbf{x}}_{t,d}}}^2} \quad (10)$$

where $\mathbf{x}_{t,d}$, $\mu_{\widehat{\mathbf{x}}_{t,d}}$ and $\sigma_{\widehat{\mathbf{x}}_{t,d}}$ are the d-th elements of $\mathbf{x}_t$, $\mu_{\widehat{\mathbf{x}}_t}$ and $\sigma_{\widehat{\mathbf{x}}_t}$, mean and standard deviation of clean feature estimate, respectively. The output vector $\mathbf{o}_t^{GSN}$ of GSN is given by:

$$\mathbf{o}_t^{GSN} = [{\mu_{\widehat{\mathbf{x}}_t}}', \log({\sigma_{\widehat{\mathbf{x}}_t}})']' \quad (11)$$

where

$$\sigma_{\widehat{\mathbf{x}}_t} = [{\sigma_{\widehat{\mathbf{x}}_{t,1}}}', {\sigma_{\widehat{\mathbf{x}}_{t,2}}}', ..., {\sigma_{\widehat{\mathbf{x}}_{t,D_{\mathbf{x}}}}}']'. \quad (12)$$

Meanwhile, LSN defines $p(\mathbf{x}_t|\mathbf{y}_t)$ as Laplacian pdfs where the each components of $\mathbf{x}_t$ is uncorrelated as in (13).

$$p(\mathbf{x}_t|\mathbf{y}_t) = \mathcal{L}(\mathbf{x}_t; \nu_{\mathbf{x}_t}(\mathbf{y}_t), b_{\mathbf{x}_t}(\mathbf{y}_t)) \quad (13)$$

Therefore, the objective function of LSN J_{LSN} is as follows:

$$J_{LSN} = \frac{1}{T}\sum_{t=1}^{T}\sum_{d=1}^{D_{\mathbf{x}}} -\log\left(2b_{\widehat{\mathbf{x}}_{t,d}}\right) - \frac{|\mathbf{x}_{t,d} - \nu_{\widehat{\mathbf{x}}_{t,d}}|}{b_{\widehat{\mathbf{x}}_{t,d}}} \quad (14)$$

where $\nu_{\widehat{\mathbf{x}}_{t,d}}$ and $b_{\widehat{\mathbf{x}}_{t,d}}$ are the d-th elements of $\nu_{\widehat{\mathbf{x}}_t}$ and $b_{\widehat{\mathbf{x}}_t}$, respectively. The output vector $\mathbf{o}_t^{LSN}$ of LSN is given by:

$$\mathbf{o}_t^{LSN} = [{\nu_{\widehat{\mathbf{x}}_t}}', \log({b_{\widehat{\mathbf{x}}_t}})']' \quad (15)$$

where

$$b_{\widehat{\mathbf{x}}_t} = [{b_{\widehat{\mathbf{x}}_{t,1}}}', {b_{\widehat{\mathbf{x}}_{t,2}}}', ..., {b_{\widehat{\mathbf{x}}_{t,D_{\mathbf{x}}}}}']'. \quad (16)$$

Comparison about network structures and objective function of two stochastic networks and DN is represented in Fig. 2.

B. Prediction Network and Joint Training

Once the training of stochastic network is completed, we can implement the acoustic modeling which considers the reliability of the clean feature estimates. In the stage of prediction network training, the network learns the mapping between the output vector of the GSN or LSN and the corresponding one-hot encoding label which contains information of the HMM states. Through the mapping, the prediction of the posterior probabilities of the HMM states considering the reliability of clean feature estimates can be performed.

After the prediction network is optimized, the stochastic and prediction networks are concatenated together to form a single unified DNN. Then, the unified network is trained jointly according to the cross-entropy criterion. Specifically, the error signal between the output of the unified DNN and the corresponding phonetic target flows back to the prediction and stochastic networks, and consequently trains all the parameters. With this series of processes, learning the relationship between the noisy features and the corresponding HMM state can be enhanced by guiding the DNN through the intermediate level features, i.e., the parametric information of the clean estimates.

By applying this training scheme, it is expected that the parameters of stochastic network contribute to more sophisticated HMM state estimation given observed input. Especially, the variance related terms such as $\log(\sigma_{\widehat{\mathbf{x}}_t})$ and $\log(b_{\widehat{\mathbf{x}}_t})$ in (11) and (15) may take roles of auxiliary features providing reliability information of estimated mean terms like $(\mu_{\widehat{\mathbf{x}}_t})$ and $(\nu_{\widehat{\mathbf{x}}_t})$.

IV. Experiments

To evaluate the speech recognition performance of the proposed approach, we performed a series of experiments in Aurora-4 DB [20]. In order to verify the performance of the proposed technique, conventional DNN-based acoustic modeling techniques were implemented and their performances were compared with that of the proposed approach. In addition to the ASR performance evaluations of all the DNN-based techniques in training-test matched conditions on noisy type, the evaluations in mismatched conditions were conducted.

A. Aurora-4 DB and GMM-HMM system

Aurora-4 DB[20] was made based on the Wall Street Journal (WSJ) DB with 5k-word vocabulary. The corpus has two training sets: clean- and multi-condition. Both clean- and multi-condition sets are composed of the same 7138 utterances. While clean-condition set deals with only a speech without any distortion, multi-condition set includes a combination of clean speech and speech corrupted by one of six different types of noises (car, babble, restaurant, street, airport and train station) at a range of signal-to-noise ratios (SNRs) between 10 and 20 dB.

The test sets including 330 utterances from 8 speakers. The sets were corrupted by the same six noises used in the training set at SNRs between 5 and 15 dB, creating a total of 14 test sets. These 14 sets were then grouped into 4 subsets based on the type of distortions: none (clean speech), additive noise only, channel distortion only and noise + channel distortion. For convenience, we denote these subsets by A, B, C and D, respectively. It is notable that the types of noises are common across training and test sets but the SNRs of the data are not.

In these experiments, we used multi-condition training data for training all the DNN-based techniques and the GMM-HMM system. The number of utterances used for HMM training was 7138. The input features for GMM-HMM were 39-dimensional MFCC features (static plus first and second order delta features) and cepstral mean normalization was performed. The multi-condition GMM-HMM system was trained with 2006 senones and 15026 Gaussian mixtures in total. We used the Kaldi speech recognition toolkit [21] for feature extraction, GMM-HMM training, alignment, and ASR decoding.

B. Stucture and training of DNNs

All the deep learning-based techniques were implemented by Keras [22] and trained using the ADADELTA optimization technique [23]. Also, dropout [24] with a fraction of 0.2 and L2 regularization with a weight of 0.00002 were applied for training all the networks. For training all the DNN-based acoustic models, log mel filterbank (FBANK) feature of 24-dimension was used. As in the case of MFCC feature above, both the first and second-order derivative of FBANK features were used.

In order to evaluate the performance of the proposed approach, four different methods of DNN-based acoustic modeling were trained. The compared techniques are

- *Baseline*: Multi-condition DNN-HMM,
- *Deterministic*: Conventional DNN-based acoustic modeling using the clean feature estimates obtained from the DE as intermediate features [16],
- *Gaussian*: GSN-based DNN-HMM,
- *Laplacian*: LSN-based DNN-HMM.

The input layer of all the techniques identically had a total of 792 visible units obtained by windowing 11 consecutive LMFB features, i.e., τ was set to be 5. Also, all the techniques had 7 hidden layers and a softmax output layer of 2006 units corresponding to senones, respectively. Each hidden layer of *Baseline* consisted of 2048 rectified linear units (ReLUs).

All the techniques except for *Baseline* aim to guide the mapping from the observed input to the corresponding HMM state via each of the intermediate feature layers. When it comes to the techniques including *Deterministic*, *Gaussian* and *Laplacian*, these techniques have their own enhancement networks and corresponding PNs. *Gaussian* and *Laplacian* exploits the mean and variance terms of $\mathbf{x}_t$ as the intermediate features at the fourth hidden layer, i.e., GSN and LSN have 3 hidden layers with 2048 ReLU nodes, respectively. Then, two different stochastic networks respectively concatenate with

their PNs, which have 3 hidden layers with 2048 ReLU nodes and 2006-dimension softmax output layers, and are trained jointly in unified networks. Meanwhile, in a practical implementation on the output representations of GSN and LSN, we modified $\mathbf{o}_t^{GSN}$ and $\mathbf{o}_t^{LSN}$ for considering the contextual coverage of the observed input $\mathbf{y}_{t-\tau:t+\tau}$. The modified $\mathbf{o}_t^{GSN}$ and $\mathbf{o}_t^{LSN}$ are represented as follows:

$$\mathbf{o}_t^{GSN} = [\mu_{\widehat{\mathbf{x}}_{t-\tau:t+\tau}}{}', \log(\sigma_{\widehat{\mathbf{x}}_{t-\tau:t+\tau}})']' \quad (17)$$

$$\mathbf{o}_t^{LSN} = [\nu_{\widehat{\mathbf{x}}_{t-\tau:t+\tau}}{}', \log(b_{\widehat{\mathbf{x}}_{t-\tau:t+\tau}})']' \quad (18)$$

Therefore, GSN and LSN of *Gaussian* and *Laplacian* has output layers with a total of 1584 linear units including mean- and variance-related terms of 792 dimensions, respectively.

Deterministic exploits the clean feature estimates, i.e, the output of the DN $\widehat{\mathbf{x}}_{t-\tau:t+\tau}$, as the intermediate features at the fourth hidden layer. Therefore, It may safely be said that *Deterministic* is constructed by replacing GSN or LSN in *Gaussian* or *Laplacian* with DN. Mini-batch size for the ADADELTA algorithm was set to 512 for all the techniques. The learning rate was set to be 1 for training all the networks, except for the cases of joint training where the learning rate was set to be 0.1. Training of each network was stopped after 20 epochs.

C. Performance evaluations on Aurora-4 DB

We evaluated the performance on Aurora-4 DB. The word error rates (WERs) of the four techniques are shown in Table I. We can see that both *Laplacian* and *Gaussian* outperformed the conventional techniques including *Baseline* and *Deterministic* in almost every condition. In case of *Laplacian*, the average relative error rate reductions (RERRs) over *Baseline* and *Deterministic* are 13.30% and 6.10%, respectively. It demonstrates that the variance-related terms of clean estimate distribution obtained from the proposed technique obviously helps the DNN-based acoustic models to supplement the reliability issue of the estimation. Also, comparing *Laplacian* with *Gaussian*, the performance of *Laplacian* in noisy subset including B and D was slightly better than those of *Gaussian*. The average relative error rate reductions (RERRs) of *Laplacian* over *Gaussian* was 3.46%.

D. Performance evaluations on mismatched noise conditions

To evaluate the proposed technique in the training-test mismatched noise conditions, we made the noise-mismatched test sets by mixing the clean speech of test set with six noises included in 100 non-speech environmental sounds [25]. Four types of noise were chosen from 100 noise types : clap, cough, crowd, machine, siren and phone dialing. Applying the same configurations of Aurora-4 DB, each noises were added to the test sets at SNRs between 5 and 15 dB with an equal rate. From the results in Table II, we can see that the proposed approach is also effective in the mismatched noise conditions. Although the gap between the proposed and conventional techniques is not that huge comparing with that in matched noise condition, the average RERRs of *Laplacian* and *Gaussian* over *Deterministic* were 3.33% and 4.31%, respectively.

TABLE I
WERs (%) ON THE COMPARED ACOUSTIC MODELING TECHNIQUES FOR TEST DATA ON AURORA-4 DB.

Method	A	B	C	D	Avrg.
Baseline	3.12	7.43	7.33	17.84	11.58
Deterministic	2.97	6.60	6.13	16.81	10.69
Laplacian	2.75	**6.18**	5.59	**15.81**	**10.04**
Gaussian	**2.75**	6.30	**6.02**	16.50	10.40

TABLE II
WERs (%) ON THE COMPARED ACOUSTIC MODELING TECHNIQUES FOR NOISE-MISMATCHED TEST DATA.

Method	Clap	Cough	Crowd	Machine	Phone	Siren	Avrg.
Baseline	14.12	21.32	17.37	18.89	18.06	14.38	17.36
Deterministic	12.58	19.91	15.22	16.24	15.78	12.26	15.33
Laplacian	12.04	18.93	**14.52**	16.01	15.39	12.01	14.82
Gaussian	**11.96**	**18.35**	14.59	**15.97**	**15.27**	**11.86**	**14.67**

V. CONCLUSIONS

In this paper, a novel deep learning-based acoustic modeling technique for estimation reliability problem was proposed. In order to consider the estimation inaccuracy in the training-test environment mismatch condition, the proposed technique designed DNN-based acoustic modeling which describes the mapping between the noisy observed features and the phonetic target as the stochastic process. The proposed technique estimates the clean estimate parameters of two well-known statistical models: Gaussian and Laplacian. According to the maximum likelihood (ML) criterion which was driven from each of the probabilistic models, the network outputs mean and variance-related terms and applies those to acoustic modeling. Through a series of experiments on Aurora-4 DB and mismatched noise conditions, we have found that the proposed technique outperforms the conventional acoustic modeling in word accuracy on both matched and mismatched conditions. Future study will deal with techniques considering other statistical models such as Gamma distribution.

ACKNOWLEDGMENT

This work was supported by the Technology Innovation Program (10076583, Development of free-running speech recognition technologies for embedded robot system) funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea).

REFERENCES

[1] A. Mohamed, G. E. Dahl, and G. E. Hinton, "Acoustic modeling using deep beliefs networks," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 14-22, Jan. 2012.

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

[3] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum Bayes risk training of deep neural network acoustic models using distributed Hessian-free optimization," in *Proc. Interspeech*, 2012, pp. 10-13.

[4] D. Yu, M. L. Seltzer, J. Li, J. Huang, and F. Seide, "Feature learning in deep neural networks - A study on speech recognition tasks," *CORR*, vol. abs/1301.3605, 2013.

[5] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Process*., vol. 32, no. 6, pp. 1109-1121, Dec. 1984.

[6] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, and A. Acero, "Robust speech recognition using a cepstral minimum-meansquare-error-motivated noise suppressor," *IEEE Trans. Audio, Speech, Language Process*., vol. 16, no. 5, pp. 1061-1070, Jul. 2008.

[7] N. S. Kim, "IMM-based estimation for slowly evolving environments," *IEEE Signal Process. Lett*., vol. 5, no. 6, pp. 146-149, Jun. 1998.

[8] L. Deng, J. Droppo, and A. Acero, "Recursive estimation of non-stationary noise using iterative stochastic approximation for robust speech recognition," *IEEE Trans. Speech, Audio, Process*., vol. 11, no. 6, pp. 568-580, Nov. 2003.

[9] A. Narayanan, and D. Wang, "Investigation of speech separation as a front-end for noise robust speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 4, pp. 826-835, Apr. 2014.

[10] W. Li, L. Wang, Y. Zhou, J. Dines, M. Magimai.-Doss, H. Bourlard, and Q. Liao, "Feature mapping of multiple beamformed sources for robust overlapping speech recognition using a microphone array ," *IEEE/ACM Trans. Audio, Speech, Language Process*., vol. 22, no. 12, pp. 2244-2255, Dec. 2014.

[11] X. Feng, Y. Zhang, and J. Glass, "Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition," in *Proc. ICASSP*, 2014, pp. 1759-1763.

[12] M. Mimura, S. Sakai, and T. Kawahara, "Exploring deep neural networks and deep autoencoders in reverberant speech recognition," in *HSCMA*, 2014, pp. 197-201.

[13] M. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. ICASSP*, 2013, pp. 7398-7402.

[14] G. Saon, H. Nahamoo, D. Nahamoo and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proc. ASRU*, 2013, pp. 55-59.

[15] M. Delcroix, K. Kinoshita, T. Hori, and T. Nakatami, "Context adaptive deep neural networks for fast acoustic model adaptation in noisy conditions," in *Proc. ICASSP*, 2016, pp. 5270–5274.

[16] T. Gao, J. Du, L.-R. Dai and C.-H. Lee, "Joint training of front-end and back-end deep neural networks for robust speech recognition,"in *Proc. ICASSP*, 2015, pp. 4375-4379.

[17] K. H. Lee, W. H. Kang, T. G. Kang, and N. S. Kim, "Integrated DNN-based model adaptation technique for noise-robust speech recognition," in *Proc. ICASSP*, 2017, pp. 5245–5249.

[18] A. Narayanan, and D. Wang, "Improving robustness of deep neural network acoustic models via speech separation and joint adaptive training," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 1, pp. 92–101, Jan. 2015.

[19] Z. Wang, and D. Wang, "A joint training framework for robust autiomatic speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 4, pp. 796–806, Apr. 2016.

[20] G. Hirsch, "Experimental framework for the performance evaluation of speech recognition front-ends on a large vocabulary task, version 2.0," ETSI STQ-Aurora DSR Working Group, 2002.

[21] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," Proc. ASRU, Hawaii, USA, Dec. 2011.

[22] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[23] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[24] N. Srivastava, et al., "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, Jun. 2014.

[25] G. Hu. (2004) 100 nonspeech environmental sounds. [Online]. Available: http://web.cse.ohiostate.edu/pnl/corpus/HuNonspeech/HuCorpus.html