DISCRIMINATIVE PIECEWISE LINEAR TRANSFORMATION BASED ON DEEP LEARNING FOR NOISE ROBUST AUTOMATIC SPEECH RECOGNITION

Yosuke Kashiwagi[1], *Daisuke Saito*[2], *Nobuaki Minematsu*[1], *Keikichi Hirose*[3]

[1]Graduate School of Engineering,
[2]Interfaculty Initiative in Information Studies,
[3]Graduate School of Information Science and Technology,
The University of Tokyo, Japan
{kashiwagi, dsk_saito, mine, hirose}@gavo.t.u-tokyo.ac.jp

ABSTRACT

In this paper, we propose the use of deep neural networks to expand conventional methods of statistical feature enhancement based on piecewise linear transformation. Stereo-based piecewise linear compensation for environments (SPLICE), which is a powerful statistical approach for feature enhancement, models the probabilistic distribution of input noisy features as a mixture of Gaussians. However, soft assignment of an input vector to divided regions is sometimes done inadequately and the vector comes to go through inadequate conversion. Especially when conversion has to be linear, the conversion performance will be easily degraded. Feature enhancement using neural networks is another powerful approach which can directly model a non-linear relationship between noisy and clean feature spaces. In this case, however, it tends to suffer from over-fitting problems. In this paper, we attempt to mitigate this problem by reducing the number of model parameters to estimate. Our neural network is trained whose output layer is associated with the states in the clean feature space, not in the noisy feature space. This strategy makes the size of the output layer independent of the kind of a given noisy environment. Firstly, we characterize the distribution of clean features as a Gaussian mixture model and then, by using deep neural networks, estimate discriminatively the state in the clean space that an input noisy feature corresponds to. Experimental evaluations using the Aurora 2 dataset demonstrate that our proposed method has the best performance compared to conventional methods.

Index Terms: Automatic speech recognition, Noise robustness, feature enhancement, Deep learning

1. INTRODUCTION

In recent years, Automatic Speech Recognition (ASR) largely increased its performance in clean speech conditions. However, in low SNR conditions, the recognition rate drastically degrades. Therefore, it is important to reduce the mismatches between training and testing conditions [1]. Feature enhancement approaches, such as SPLICE [2], Denoising AutoEncoder (DAE) [3], and so on [4–6] are techniques that can be performed on the front-end for this aim. They reduce the mismatches by estimating clean features from observed noisy features.

SPLICE is composed of two steps. First, it models the input noisy feature space as a mixture of Gaussians (GMM) and calculates the posterior probability of the component index given a noisy feature. Next, the clean features are estimated as results of posterior-based weighted sum of linear transformations. However, division of the noisy space and that of the clean space based on GMM are often different from each other. In addition, division of the noisy space also depends on the type of noise.

To mitigate this problem, another approach was proposed, where the *clean* space is modeled as GMMs. REgularized piecewise linear mapping with DIscriminative region weighting And Long-span features (REDIAL) models the clean feature space by GMMs, and estimates their component indices that noisy features correspond to by using a discriminative approach. In REDIAL, GMM and Linear Discriminant Analysis (LDA) [7–9] are used together for discrimination. This method achieved high performance especially in the multi-condition of the Aurora 2. However, linearity of LDA is thought to limit the performance of this method because observed noisy features and clean speech states are considered to have a very complicated structure between them.

On the other hand, DAE was proposed to estimate clean features from observed noisy features non-linearly and directly by neural networks. Deep Denoising AutoEncoder (DDAE), which is stacked DAE, achieved high performance. In this case, however, it tends to suffer from well-known over-fitting problems. Therefore, another efficient approach should be investigated.

The basic idea of our method uses deep neural networks [11] and stereo (clean and noisy) data to realize a method that can estimate the states in the clean space only from observed noisy features. First, GMMs of the clean features are constructed and, by using deep neural networks, the state that an input feature corresponds to is estimated in the form of posterior probability. Next, similarly to REDIAL, linear transformations from the observed features to the clean features are trained using the above posteriors.

This paper is organized as follows. We formulate the algorithm in Section 2. In Section 3, we compare our method to the conventional methods in their performance. Experimental results are given in Section 4. Finally our paper is summarized in Section 5.

2. ALGORITHM FORMULATION

To gain the speech recognition performance in noisy conditions, feature enhancement approaches attempt to reconstruct clean features from input noisy features. To

address this problem in this paper, we use deep neural networks to calculate posterior probability of the clean speech state given an input noisy feature, and predict its corresponding clean feature by linear transformations using the posteriors as weights.

The reason why we use deep neural networks to estimate the state of the clean features is that the conventional piecewise linear transformation methods do not always divide a space into such subspaces that local linearity is satisfied well in conversion. In the training phase, we can use parallel data of clean and noisy features. So, any noisy feature has its clean version. By using the clean feature associated with an input noisy feature, we can find the clean state that the clean feature belongs to. If we're allowed to use a posterior probability of this clean state as oracle posterior, oracle SPLICE and its transformation matrices and biases can be obtained. With these parameters, accurate clean features can be reasonably obtained. Experimental discussion of this oracle SPLICE is done in the following section. We consider that performance degradation from the oracle SPLICE to ordinary SPLICE can be attributed to a division mismatch between the noisy feature space and the clean feature space. In other words, there seems to exist a complicated relation between both spaces. REDIAL tries to reduce the complexity by LDA, however linearity of LDA will limit the performance. On the other hand, DNNs are expected to be able to capture the complexity well.

The simplest way to use DNNs to estimate clean features is a direct mapping approach. It estimates the clean speech features directly using DNNs which are trained by back-propagation using a minimum mean square error criterion. This mapping technique can achieve high performance despite its simplicity. However, in open-noise conditions, the performance tends to decrease. This is due to the weakness of the restrictions on the proximity of the output clean speech features in the neural networks. Therefore, by estimating the state index of clean speech, it is possible to construct a neural network that holds explicitly number of classes of the output clean speech features.

Let $\{(\boldsymbol{x}_t, \boldsymbol{y}_t)\}$ denote a set of stereo data, where $\boldsymbol{x}_t$ is a clean feature at time t and $\boldsymbol{y}_t$ is a noisy feature. Both of them are N dimensional vectors. Fig. 1 shows the flowchart of our method in the training phase. The first step of our method is training a probabilistic model of $p(\boldsymbol{x})$ for clean speech features. Like REDIAL, GMMs are adopted for this purpose.

$$p(\boldsymbol{x}_t) = \sum_k p(k)p(\boldsymbol{x}_t|k) \ , \quad (1)$$

$$p(\boldsymbol{x}_t|k) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\sigma}_k^{\boldsymbol{x}}) \, , \quad (2)$$

$$p(k) = \pi_k^{\boldsymbol{x}} \, . \quad (3)$$

Posterior probability of the clean state given $\boldsymbol{x}_t$ is obtained as follows.

$$p(k|\boldsymbol{x}_t) = \frac{p(\boldsymbol{x}_t|k)p(k)}{\sum_{k'} p(\boldsymbol{x}_t|k')p(k')} \ . \quad (4)$$

In the testing phase, because we cannot use $\boldsymbol{x}_t$, $p(k|\boldsymbol{x}_t)$ cannot be applied directly. Therefore, DNNs to estimate $p(k|\boldsymbol{y}_t)$ are trained using training data of

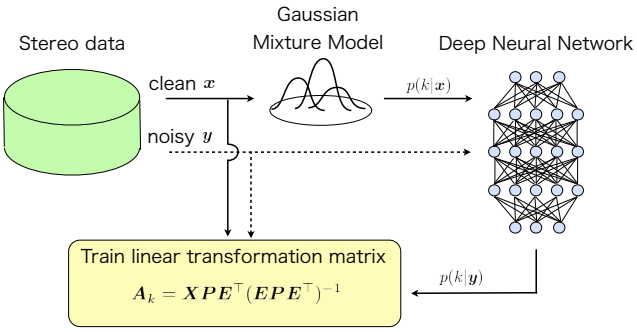

Figure 1: The flowchart of our method (Training phase).

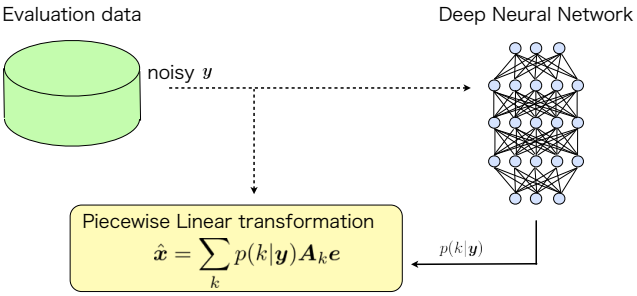

Figure 2: The flowchart of our method (Test phase).

$\{\underset{k}{\text{argmax}}\, p(k|\boldsymbol{x}_t), \boldsymbol{y}_t\}_{t=1\ldots T}$ as follows.

$$p(k|\boldsymbol{y}_t) \simeq p(k|\boldsymbol{d}_t) = \text{softmax}_k(\boldsymbol{V}\boldsymbol{h}^{(n)}(\boldsymbol{d}_t) + \boldsymbol{c}), \quad (5)$$

$$\boldsymbol{h}^{(n)}(\boldsymbol{d}_t) = \boldsymbol{\sigma}(\boldsymbol{W}^{(n)}\boldsymbol{h}^{(n-1)}(\boldsymbol{d}_t) + \boldsymbol{b}^{(n)}) \, , \quad (6)$$

$$\boldsymbol{h}^{(1)}(\boldsymbol{d}_t) = \boldsymbol{\sigma}(\boldsymbol{W}^{(1)}\boldsymbol{d}_t + \boldsymbol{b}^{(1)}) \, , \quad (7)$$

where $\boldsymbol{\sigma}$ is a vector sigmoid function, and the weight matrices $\boldsymbol{V}$ and $\boldsymbol{W}^{(n)}$ along with the bias vectors $\boldsymbol{c}$ and $\boldsymbol{b}^{(n)}$ are parameters of the neural networks. $\boldsymbol{h}^{(n)}(\boldsymbol{y})$ is an ouput vector from the n-th hidden layer. $\boldsymbol{d}_t$ is a feature vector after time context expansion as

$$\boldsymbol{d}_t = [\boldsymbol{y}_{t-s}^\top, \ldots, \boldsymbol{y}_{t-1}^\top, \boldsymbol{y}_t^\top, \boldsymbol{y}_{t+1}^\top, \ldots, \boldsymbol{y}_{t+s}^\top]^\top \, . \quad (8)$$

The proposed method adopts deep belief nets, each layer of which is trained initially by restricted Boltzmann machine (RBM), and trained subsequently by stochastic gradient descent [11].

Using the DNNs trained above, given any input noisy feature, posterior probability of its corresponding clean state $p(k|\boldsymbol{y}_t)$ can be estimated. Finally, clean speech feature $\boldsymbol{x}_t$ is predicted as $\hat{\boldsymbol{x}}_t$ through weighted sum of linear transformations using $p(k|\boldsymbol{y}_t)$ (See Fig. 2).

$$\hat{\boldsymbol{x}}_t = \sum_k p(k|\boldsymbol{y}_t)\boldsymbol{A}_k\boldsymbol{e}_t \, . \quad (9)$$

where $\boldsymbol{A}_k$ is an affine transformation matrix which corresponds to the k-th state and $\boldsymbol{e}_t$ is an expanded vector as

$$\boldsymbol{e}_t = [1, \boldsymbol{y}_{t-u}^\top, \ldots, \boldsymbol{y}_{t-1}^\top, \boldsymbol{y}_t^\top, \boldsymbol{y}_{t+1}^\top, \ldots, \boldsymbol{y}_{t+u}^\top]^\top . \quad (10)$$

Table 1: Performance gap between ordinary SPLICE and the oracle SPLICE. The difference between the two lies in how to estimate posteriors. In the former, it is done using input noisy feature $\boldsymbol{y}_t$ and in the latter it is obtained by using $\boldsymbol{x}_t$, which is paired with $\boldsymbol{y}_t$. 'Average' shows averaged results over the range of SNR20 to SNR0.

	SPLICE	SPLICE (Oracle)
clean	0.57	0.69
SNR 20	1.08	0.76
SNR 15	1.99	0.70
SNR 10	4.65	0.77
SNR 5	16.76	0.93
SNR 0	49.96	0.95
SNR -5	81.46	1.13
Average	14.89	0.82

$\boldsymbol{A}_k$ can be trained using the following weighted minimum mean square error (MMSE) criterion

$$\hat{\boldsymbol{A}}_k = \underset{\boldsymbol{A}_k}{\operatorname{argmin}} \sum_j p(k|\boldsymbol{y}_j)||\boldsymbol{x}_j - \boldsymbol{A}_k \boldsymbol{e}_j||^2. \tag{11}$$

This equation can be solved analytically as follows

$$\hat{\boldsymbol{A}}_k = \boldsymbol{X}\boldsymbol{P}\boldsymbol{E}^\top (\boldsymbol{E}\boldsymbol{P}\boldsymbol{E}^\top)^{-1}, \tag{12}$$

where $\boldsymbol{X} \in \mathcal{R}^{N\times T}$ is $[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T]$ and $\boldsymbol{E} \in \mathcal{R}^{(N(2u+1)+1)\times T}$ is $[\boldsymbol{e}_1, \ldots, \boldsymbol{e}_T]$ and $\boldsymbol{P} \in \mathcal{R}^{T\times T}$ is a diagonal matrix whose elements are $p(k|\boldsymbol{y}_t)$.

3. COMPARISON TO CONVENTIONAL STATISTICAL FEATURE MAPPING APPROACHES

This sections compares theoretically our method to conventional statistical feature mapping approaches of SPLICE, REDIAL and DAE.

3.1. SPLICE

SPLICE is a speech enhancement method, which estimates clean speech features from noisy speech features with piecewise linear transformations as

$$\hat{\boldsymbol{x}}_t = \sum_i p(i|\boldsymbol{y}_t)\boldsymbol{A}_i \begin{bmatrix} 1 \\ \boldsymbol{y}_t \end{bmatrix} . \tag{13}$$

In SPLICE, GMMs are adopted to model the probability $p(\boldsymbol{y})$ of the noisy speech features as follows

$$p(\boldsymbol{y}_t) = \sum_i \pi_i^{\boldsymbol{y}} \mathcal{N}(\boldsymbol{y}_t; \boldsymbol{\mu}_i^{\boldsymbol{y}}, \boldsymbol{\sigma}_i^{\boldsymbol{y}}) . \tag{14}$$

Therefore, posterior $p(i|\boldsymbol{y}_t)$ is calculated as

$$p(i|\boldsymbol{y}_t) = \frac{p(\boldsymbol{y}_t|i)p(i)}{\sum_{i'} p(\boldsymbol{y}_t|i')p(i')} . \tag{15}$$

Equation 14 shows that the GMMs in SPLICE are trained only with noisy features available in the training phase and used as they are in the testing phase. We can say that this is not good because those GMMs tend to be overfitted to the noisy features in training data. Therefore, it is better to model GMMs using clean features.

Table 1 shows averaged word error rates which are experimental results in the test set A (closed-noise condition) of the Aurora 2 dataset. The oracle SPLICE weights linear transformations using posterior $p(i^*|\boldsymbol{x}_t)$, where i^* is an state index in the clean space.

$$\hat{\boldsymbol{x}}_t = \sum_{i^*} p(i^*|\boldsymbol{x}_t)\boldsymbol{A}_{i^*} \begin{bmatrix} 1 \\ \boldsymbol{y}_t \end{bmatrix} , \tag{16}$$

$$p(i^*|\boldsymbol{x}_t) = \frac{p(\boldsymbol{x}_t|i^*)p(i^*)}{\sum_{i^{*\prime}} p(\boldsymbol{x}_t|i^{*\prime})p(i^{*\prime})} . \tag{17}$$

The oracle results clearly show that, if $p(i|\boldsymbol{y}_t)$ can be estimated accurately, it definitely leads to large performance improvement. The fundamental implementation of NMN-SPLICE is somewhat similar to the oracle SPLICE. NMN-SPLICE subtracts from input features, weights linear transformations with posteriors estimated from obtained quasi-clean features [2].

3.2. REDIAL

REDIAL is one of the approaches to estimate the clean feature state from noisy speech features. In REDIAL, this estimation is implemented by integrating GMMs with LDA. First, the dimensionality reduction matrix $\boldsymbol{L}$ of LDA is trained using $\{\{p(k|\boldsymbol{x}_t)\}_{k=1\cdots K}, \boldsymbol{d}_t\}_{t=1\cdots T}$ as soft labels,

$$\boldsymbol{L} = \underset{\boldsymbol{L}}{\operatorname{argmin}} \frac{\boldsymbol{L}^\top \boldsymbol{\Sigma}^w \boldsymbol{L}}{\boldsymbol{L}^\top \boldsymbol{\Sigma}^b \boldsymbol{L}} , \tag{18}$$

where

$$\boldsymbol{\Sigma}^w = \sum_k \sum_j \left\{ p(k|\boldsymbol{x}_t) \left(\boldsymbol{d}_t - \boldsymbol{\mu}_k^w\right) \left(\boldsymbol{d}_t - \boldsymbol{\mu}_k^w\right)^\top \right\}, \tag{19}$$

$$\boldsymbol{\Sigma}^b = \sum_k \left(\sum_j p(k|\boldsymbol{x}_t) \right) (\boldsymbol{\mu}^{\boldsymbol{d}} - \boldsymbol{\mu}_k^{\boldsymbol{d}})(\boldsymbol{\mu}^{\boldsymbol{d}} - \boldsymbol{\mu}_k^{\boldsymbol{d}})^\top , \tag{20}$$

$$\boldsymbol{\mu}^{\boldsymbol{d}} = \frac{\sum_j \boldsymbol{d}_t}{J} , \tag{21}$$

$$\boldsymbol{\mu}_k^{\boldsymbol{d}} = \frac{1}{\sum_j p(k|\boldsymbol{x}_t)} \sum_j p(k|\boldsymbol{x}_t)\boldsymbol{d}_j . \tag{22}$$

Analytical solution of (18) is obtained by solving the eigenvalue problem related to $(\boldsymbol{\Sigma}^w)^{-1}\boldsymbol{\Sigma}^b$. Next, the K^*-component GMMs of the compressed feature vectors, $\boldsymbol{v}_j = \boldsymbol{L}\boldsymbol{d}_j$, are trained as

$$p(\boldsymbol{v}_t) = \sum_{k^*=1}^{K^*} \pi_{k^*}^{\boldsymbol{v}} \mathcal{N}(\boldsymbol{v}_t; \boldsymbol{\mu}_{k^*}^{\boldsymbol{v}}, \boldsymbol{\sigma}_{k^*}^{\boldsymbol{v}}) . \tag{23}$$

Then, posterior $p(k^*|\boldsymbol{y}_t)$ is approximated as $p(k^*|\boldsymbol{v}_t)$,

$$p(k^*|\boldsymbol{y}_t) \simeq p(k^*|\boldsymbol{v}_t) = \frac{p(\boldsymbol{v}_t|k^*)p(k^*)}{\sum_{k^{*\prime}} p(\boldsymbol{v}_t|k^{*\prime})p(k^{*\prime})} . \tag{24}$$

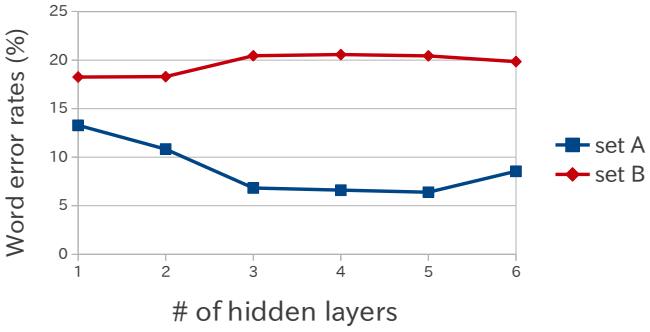

Figure 3: The performance of deep denoising autoencoder varying the number of hidden layers in Aurora 2 dataset.

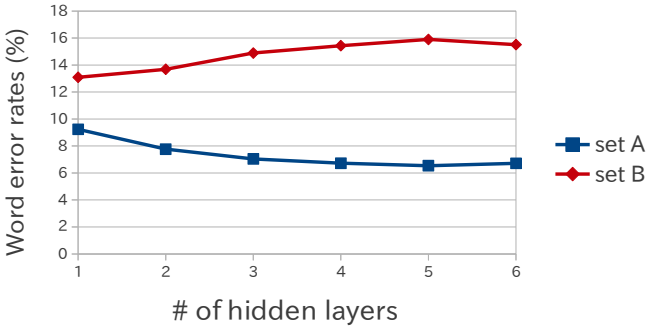

Figure 4: The performance of the proposed method varying the number of hidden layers in Aurora 2 dataset.

REDIAL estimates the clean speech features by weighted linear transformations as

$$\hat{\boldsymbol{x}}_t = \sum_{k^*} p(k^*|\boldsymbol{y}_t)\boldsymbol{A}_{k^*}\boldsymbol{e}_t\,. \quad (25)$$

$$\boldsymbol{e}_t = [1, \boldsymbol{y}_{t-u}^\top, \ldots, \boldsymbol{y}_{t-1}^\top, \boldsymbol{y}_t^\top, \boldsymbol{y}_{t+1}^\top, \ldots, \boldsymbol{y}_{t+u}^\top]^\top. \quad (26)$$

Since $\boldsymbol{A}_{k^*}$ has a large number of parameters, REDIAL calculates it based on the weighted MMSE criterion using regularization.

In theoretical comparison between our method and REDIAL, the difference is found only in how to estimate posterior of the component index from noisy features. Because LDA is linear transformation, linearity of LDA will not be adequate to capture the complex relationship between the clean states and the noisy features.

3.3. DAE

DAE is a neural network which attempts to reconstruct clean features from input nosy features directly. DDAE has a multilayer structure and it can estimate clean features as

$$\hat{\boldsymbol{x}}_t = \boldsymbol{U}\boldsymbol{h}^{(n)}(\boldsymbol{d}_t) + \boldsymbol{c}\,, \quad (27)$$

$$\boldsymbol{h}^{(n)}(\boldsymbol{d}_t) = \boldsymbol{\sigma}(\boldsymbol{W}^{(n)}\boldsymbol{h}^{(n-1)}(\boldsymbol{d}_t) + \boldsymbol{b}^{(n)})\,, \quad (28)$$

$$\boldsymbol{h}^{(1)}(\boldsymbol{d}_t) = \boldsymbol{\sigma}(\boldsymbol{W}^{(1)}\boldsymbol{d}_t + \boldsymbol{b}^{(1)})\,, \quad (29)$$

where $\boldsymbol{U}$ is a weight matrix. In this paper, we compared our approach to DDAE which is pre-trained with RBM in all layers and fine-tuned by back-propagation based on the MMSE criterion. Fig. 3 shows word error rates (WERs) as a function of the number of hidden layers. Those results were obtained using the Aurora 2 dataset. Set A and B were closed-noise and open-noise settings, respectively. The number of hidden nodes in each layer was fixed to 1024. From the results, the structure of multilayer improved the recognition performance in the closed-noise condition, however it was not effective in the open-noise condition.

Since DDAE has a large number of model parameters, it tends to be over-fitted to training data used. In contrast in our approach, the number of model parameters to estimate can be kept as constant against variety of the kind of environmental noises. This is because our neural network is trained whose output layer is associated with the states in the clean feature space, not in the noisy feature space. This strategy makes the size of the output layer independent of the kind of a given noisy environment.

Recently, a complex denoising autoencoder using deep recurrent neural networks was proposed in [12]. Although It seems that the topology of DNNs is a key topic to optimize DNN-based methods, in the following section, we do not use recurrent networks. We consider that the network topology is independent of the main theme of this paper, where DNN is tested as posterior estimator and compared to a GMM-based estimator.

4. EXPERIMENTAL EVALUATION

The performance of the proposed method was evaluated using Aurora 2 database under the task of continuous digits recognition in noisy conditions. The database contains connected digits recorded in a clean environment and some types of noises are added to the utterances. Therefore, parallel data sets can be used for training. The database defined two training conditions (clean condition and multi condition) for acoustic models. In the clean condition, the HMMs were trained with only clean data. Further, the HMMs trained using clean data and noisy data were also provided and they are referred to as multi-conditioned HMMs. The database also defined three sets of utterances for testing, sets A, B, and C, according to the type of noise. Three test sets are defined against noise types (sets A, B and C). Set A contains the utterances in the noise conditions which were used in recording training utterances. Set B contains the noise conditions which are not found in training utterances. In set A and set B, the same microphone and channel were used in recording training and testing utterances. In set C, a different channel condition is introduced. Continuous digits recognition experiments were carried out with the complex backend scripts with HTK 3.4 [13] . MFCC and the first and second derivations were used as basic features. For training DNNs, the KALDI toolkit was used [14].

First, the recognition performance of the proposed method against the parameters was investigated. Fig. 4 shows WERs. The window length of temporal context expansion s in (8) and u in (10) were both fixed to 3. The number of hidden nodes in each layer is 1024. The num-

Table 2: Performance comparison among our proposal and conventional methods (word error rates %).

	clean condition (WER. %)				multi condition (WER. %)			
	set A	set B	set C	average	set A	set B	set C	average
Baseline	48.93	55.80	39.23	47.98	10.57	11.89	14.33	12.27
SPLICE	14.89	19.31	21.59	18.60	9.20	14.50	15.22	12.97
REDIAL	16.70	20.59	21.14	19.48	8.98	**13.26**	**12.45**	11.56
DDAE	**6.39**	20.44	17.20	14.68	5.97	18.50	14.67	13.04
PROPOSED	7.04	**14.93**	**15.54**	**12.51**	**5.64**	15.20	13.29	**11.38**

Table 3: The word error rates (%) of DDAE in each noisy condition (clean condition).

	closed-noise condition (Set A)					open-noise condition (Set B)				
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average
Clean	0.58	0.60	0.81	0.46	0.61	0.58	0.60	0.81	0.46	0.61
SNR 20	1.29	0.85	0.78	0.96	0.97	1.11	2.21	1.82	1.14	1.57
SNR 15	1.54	1.45	1.16	1.79	1.49	2.21	6.32	3.91	3.46	3.98
SNR 10	2.27	3.02	2.00	2.81	2.53	6.05	18.02	10.11	8.67	10.71
SNR 5	4.64	8.01	4.47	6.70	5.96	18.33	40.39	25.95	24.38	27.26
SNR 0	14.77	31.29	18.85	19.10	21.00	51.30	69.35	58.75	55.29	58.67
SNR -5	46.58	72.64	63.11	49.27	57.90	97.91	90.93	96.42	86.52	92.95
Average	4.90	8.92	5.45	6.27	6.39	15.80	27.26	20.11	18.59	20.44

Table 4: The word error rates (%) of the proposed method in each noisy condition (clean condition).

	closed-noise condition (Set A)					open-noise condition (Set B)				
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average
Clean	0.49	0.57	0.60	0.65	0.58	0.49	0.57	0.60	0.65	0.58
SNR 20	1.01	0.91	0.66	0.86	0.86	0.71	1.45	1.04	1.05	1.06
SNR 15	1.72	1.21	1.22	1.57	1.43	1.35	2.96	2.00	1.85	2.04
SNR 10	2.36	2.60	2.12	2.81	2.47	3.22	9.98	5.58	4.72	5.88
SNR 5	5.25	7.62	6.80	7.41	6.77	10.96	29.29	17.36	16.17	18.45
SNR 0	16.40	31.59	26.04	20.77	23.70	35.40	62.36	43.48	47.73	47.24
SNR -5	49.86	72.13	69.31	54.46	61.44	78.94	88.09	83.84	84.11	83.75
Average	5.35	8.79	7.37	6.68	7.05	10.33	21.21	13.89	14.30	14.93

ber of output classes, which equals to that of the states of the clean feature space, was set to 1024. The number of training utterances was 8,440 in the corpus. They were divided in this experiment into a development set of 844 utterances and a training set of 7,596 utterances, which were used in the step of fine tuning. In the experiment, input utterances with various types of noises added were tested. In each case of the noises, regularization of linear transformation matrices was performed similarly to REDIAL. To estimate parameters of regularization, 3-fold cross validation was done.

According to the results in Fig. 4, the deep architecture was also effective to estimate the state of the clean features. In addition, the most interesting point is that the effect of the deep architecture in the open-noise condition is similar to that of DDAE.

Next, we compare the performance of the proposed method to that of SPLICE, REDIAL, and DDAE. For SPLICE, the number of noisy states was set to 1024. For REDIAL, that of clean states was also set to 1024, and the dimensionality transformed by LDA was set to 64. The number of hidden layers was set to 5 in DDAE. As for context expansion, the same window length were used for READIAL, DDAE, and our method.

Table 2 shows the results. It was found that the proposed method realizes significant improvement in the clean condition. In contrast, REDIAL achieved lower word error rates in the multi condition. The reason might be that the linear transformation of LDA in REDIAL can keep the topology of the classes (states) in the clean space. Table 3 and 4 shows the detailed results of DDAE and the proposed method in the clean condition. Although the proposed method shows a slightly higher WER on average in the closed-nose condition, it achieves a much lower WER in the open-noise condition.

5. CONCLUSION

We have presented a stochastic mapping technique for robust speech recognition that uses stereo data. Our novel approach models the clean features by GMMs and applies deep neural network to estimate the clean speech features and linear transform the input features to clean features weighted by the posterior. We demonstrate the method is competitive with existing feature denoising approaches on the Aurora 2 task, then our method outperforms them.

One interesting extension of the proposed method is to implement piecewise linear transformation based on DNN with neural network format, then fine tune the whole parameters. our approach has a possibility to connect the conventional statistical feature mapping and the deep learning approaches in a proper manner.

6. REFERENCES

[1] Gales, Mark JF, "Model-based approaches to handling uncertainty," *Robust Speech Recognition of Uncertain or Missing Data-Theory and Applications. Springer, Berlin, Germany*, pp. 101–125, 2011.

[2] Droppo, Jasha and Deng, Li and Acero, Alex, "Evaluation of SPLICE on the Aurora 2 and 3 tasks," *International Conference on Spoken Language Processing*, pp. 29–32, 2002.

[3] Vincent, Pascal and Larochelle, Hugo and Bengio, Yoshua and Manzagol, Pierre-Antoine, "Extracting and composing robust features with denoising autoencoders," *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

[4] Li, Jinyu and Seltzer, Michael L and Gong, Yifan, "Improvements to VTS feature enhancement," *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4677–4680, 2012.

[5] Afify, Mohamed and Cui, Xiaodong and Gao, Yuqing, "Stereo-based stochastic mapping for robust speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 7, pp. 1325–1334, 2009.

[6] Gemmeke, Jort F and Virtanen, Tuomas and Hurmalainen, Antti, "Exemplar-based sparse representations for noise robust automatic speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 7, pp. 2067–2080, 2011.

[7] Senior, Andrew and Cho, Youngmin and Weston, Jason "Learning improved linear transforms for speech recognition.," Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, pp. 1957–1960, 2012.

[8] Suzuki, Masayuki and Yoshioka, Takuya and Watanabe, Shinji and Minematsu, Nobuaki and Hirose, Keikichi "MFCC enhancement using joint corrupted and noise feature space for highly non-stationary noise environments," Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, pp. 4109–4112, 2012.

[9] Suzuki Masayuki, Yoshioka Takuya, Watanabe Shinji, Minematsu Nobuaki, and Hirose Keikichi "Feature Enhancement With Joint Use of Consecutive Corrupted and Noise Feature Vectors With Discriminative Region Weighting," IEEE TASLP, (to appear).

[10] http://eurospeech2001.org/ese/NoiseRobust/index.html, http://www.elda.fr/proj/aurora1.html, http://www.elda.fr/proj/aurora2.html

[11] Hinton, Geoffrey E and Osindero, Simon and Teh, Yee-Whye "A fast learning algorithm for deep belief nets," Neural computation, vol. 18, no. 7, pp. 1527–1554, 2006.

[12] Maas, Andrew L and Le, Quoc V and O'Neil, Tyler M and Vinyals, Oriol and Nguyen, Patrick and Ng, Andrew Y, "Recurrent Neural Networks for Noise Reduction in Robust ASR," INTERSPEECH, 2012.

[13] http://htk.eng.cam.ac.uk/

[14] Povey, Daniel and Ghoshal, Arnab and Boulianne, Gilles and Burget, Lukas and Glembek, Ondrej and Goel, Nagendra and Hannemann, Mirko and Motlicek, Petr and Qian, Yanmin and Schwarz, Petr and others "The kaldi speech recognition toolkit," IEEE 2011 workshop on automatic speech recognition and understanding, 2011.