IEICE TRANS. INF. & SYST., VOL.E99–D, NO.2 FEBRUARY 2016

LETTER

DNN-Based Voice Activity Detection with Multi-Task Learning

Tae Gyoon KANG[†], *Nonmember and* **Nam Soo KIM**[†a)], *Member*

SUMMARY Recently, notable improvements in voice activity detection (VAD) problem have been achieved by adopting several machine learning techniques. Among them, the deep neural network (DNN) which learns the mapping between the noisy speech features and the corresponding voice activity status with its deep hidden structure has been one of the most popular techniques. In this letter, we propose a novel approach which enhances the robustness of DNN in mismatched noise conditions with multi-task learning (MTL) framework. In the proposed algorithm, a feature enhancement task for speech features is jointly trained with the conventional VAD task. The experimental results show that the DNN with the proposed framework outperforms the conventional DNN-based VAD algorithm.

key words: *deep neural network, voice activity detection, multi-task learning*

1. Introduction

Voice activity detection (VAD) algorithms have been widely applied to speech communication systems and front-end processing modules for the last few decades. Traditional approaches to VAD problem have usually been designed based on the assumption of stationary background noise. Recently, further improvements in VAD problem have been achieved by adopting several machine learning techniques [1]–[4]. The fundamental idea of these techniques is to consider the VAD task as a two-class classification problem and learn the mapping between the noisy speech features and the corresponding voice activity status from a huge amount of exemplars.

Compared to the traditional VAD algorithms, machine learning-based techniques are competitive in two aspects. First, machine learning-based VAD algorithms do not require any structural model which should be verified rigorously but only a set of examples of the input-output pairs is adequate. Second, they can fuse information from a variety of different features. Fusing various features allows them to estimate the voice activity status from richer information.

Among a number of machine learning techniques, the deep neural network (DNN) which learns the mapping between the noisy speech features and the corresponding voice activity status with its deep hidden structure has been one of the most popular techniques. The DNN-based VAD algorithm outperformed the traditional and other machine learning-based VAD algorithms since the DNN is efficient in learning the complicated inter-dependencies between the input variables [5].

Though the DNN-based VAD algorithm showed good performance in matched noise conditions, its performance sometimes deteriorates when the training and test environments are not matched with each other. To ameliorate this performance degradation, the mapping learned by the DNN should be general enough to cover the possible environmental mismatches.

In this letter, we propose a novel approach which enhances the robustness of DNN with the use of the multi-task learning (MTL) framework. In the MTL framework, several related tasks are jointly trained with shared hidden layers to improve the generalization power of each task [6]–[10]. In the proposed approach, the main task of VAD is jointly trained with a subsidiary task of feature enhancement.

By employing the feature enhancement task, the DNN is encouraged to denoise the noisy speech feature before estimating the voice activity status, which is useful to maintain the VAD performance in mismatched noise conditions. Experiments performed on Aurora2 database demonstrated that the DNN trained under the MTL framework is superior to the conventional DNN-based VAD algorithm.

2. DNN-Based VAD Algorithm with MTL Framework

In this section, we briefly review the conventional DNN-based VAD algorithm which considers the VAD task as a two-class classification problem [5]. The MTL framework for training the DNN will be followed with detailed description.

2.1 A Brief Review on DNN-Based VAD Algorithm

Figure 1(a) shows the structure of the conventional DNN for the VAD task. The DNN consists of an input layer, a few hidden layers and an output layer which are fully connected to their adjacent layers. For the sake of notation simplicity, the number of hidden layers is denoted as L and the input and output layers of the DNN are denoted as the 0-th and $(L+1)$-th layers of the DNN, respectively. The input vector of the DNN is usually given by the noisy speech features.

For the l-th hidden layer, the number of nodes in the layer is denoted by n_l. The n_l-dimensional activation vector $\mathbf{v}^l$ is defined as follows:

$$\mathbf{v}^l = g(\mathbf{a}^l) = g(W^l \mathbf{v}^{l-1} + \mathbf{b}^l) \quad (1)$$

Manuscript received July 31, 2015.
Manuscript publicized October 26, 2015.

[†]The authors are with the Department of Electrical and Computer Engineering and the Institute of New Media and Communications, Seoul National University, Seoul 151–742, Korea.
a) E-mail: nkim@snu.ac.kr (Corresponding author)

DOI: 10.1587/transinf.2015EDL8168

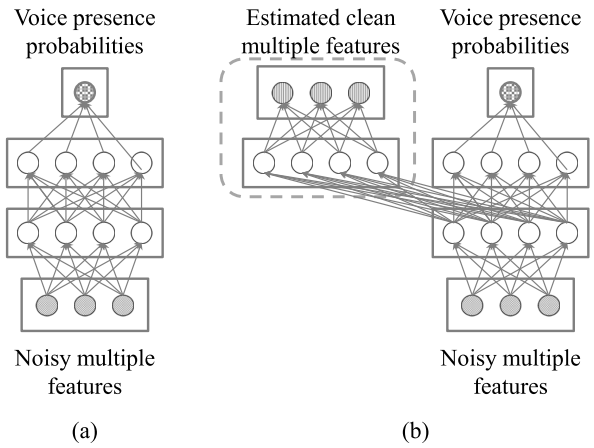

Fig. 1 Scheme of the conventional DNN (a) and the MTL-DNN (b) for the VAD task. The layers in the dotted line in (b) are discarded before the test stage.

where $\mathbf{a}^l$, W^l, and $\mathbf{b}^l$ denote the n_l-dimensional excitation vector, $n_l \times n_{l-1}$-dimensional weight matrix and n_l-dimensional bias vector, respectively, and $g(\cdot)$ represents an element-wise activation function. In this letter, all hidden layers of the DNN use the element-wise logistic sigmoid function which is defined as follows:

$$\sigma(\mathbf{a}_i^l) = \frac{1}{1 + e^{-\mathbf{a}_i^l}} \tag{2}$$

where $\mathbf{a}_i^l$ denotes the i-th element of the $\mathbf{a}^l$. Since the VAD task is a binary classification problem, the activation vector of the output layer consists of a single node $\hat{z}$ which is given by

$$\hat{z} = \sigma(W^{L+1}\mathbf{v}^L + \mathbf{b}^{L+1}). \tag{3}$$

Interested readers are referred to [11] for more detail on DNN.

The DNN-based VAD technique consists of three parts: pre-training, fine-tuning and test stages. In the pre-training stage, DNN parameters are initialized using stacked restricted Boltzmann machines trained through greedy layer-wise unsupervised learning [11]. After the pre-training stage, the fine-tuning stage which involves stochastic gradient descent and backpropagation is carried out with the cross-entropy objective function C_{VAD} which is defined as follows:

$$C_{VAD} = -z ln(\hat{z}) - (1 - z) ln(1 - \hat{z}) \tag{4}$$

where z denotes the actual target output value which equals 1 for active voice and 0 for inactive voice, respectively. In the test stage, $\hat{z}$ is estimated from the noisy input features through the standard feedforward processing and the final decision to VAD is made according to

$$H_d = \begin{cases} H_1, & \text{if } \hat{z} > \eta, \\ H_0, & \text{otherwise.} \end{cases} \tag{5}$$

where H_1 and H_0 denote active voice and noise-only hypothesis, respectively, and η is a threshold which is usually set to 0.5.

2.2 MTL Framework for DNN-Based VAD Algorithm

The performance of the DNN-based VAD algorithm with the conventional training procedure is deteriorated in some mismatched noise conditions since the mapping learned by the DNN is not general enough to cover the environmental mismatches. When the DNN is trained with the conventional training procedure, the DNN can learn the mapping between the noisy features and the corresponding voice activity status in several ways, e.g., relying on trivial characteristics or simply memorizing the training data [9]. Thus DNNs with these mappings may have difficulties in estimating voice activity status when there exists severe mismatch in noise condition.

In this letter, we introduce the MTL framework which combines the conventional VAD task with a feature enhancement task during the training stage in order to ameliorate this performance degradation. The DNN with the proposed MTL framework (MTL-DNN) denoises the noisy speech features in the shared hidden layers and learns the mapping between the denoised hidden representation and the corresponding voice activity status in the separated layers for the VAD task. The mapping which is learned by the MTL-DNN is more robust against the environmental mismatches since it represents the general denoising function for the speech features.

Figure 1(b) shows the network structure of the MTL-DNN where the conventional VAD and feature enhancement tasks share the lower hidden layers of the DNN. The right part of this network has the same structure with Fig. 1(a) which performs VAD while the left part performs feature enhancement. Both the left and right parts of the DNN share the lower hidden layers including the input layer but produce different types of outputs; left part gives the enhanced speech features while the right part outputs the voice activity status. The left part of the network is treated as a subsidiary DNN, which means that it is used only for training the DNN parameters and it is removed after training.

Similar to the conventional DNN-based VAD technique, the MTL-DNN is trained by passing through the pre-training and fine-tuning stages. In the pre-training stage, the parameters of the MTL-DNN are initialized by the same layer-wise unsupervised learning algorithm. In the fine-tuning stage, the objective function for the feature enhancement task C_{FE} is given by the Euclidean distance between the target clean feature $\mathbf{y}$ and its estimated value $\hat{\mathbf{y}}$ as follows:

$$C_{FE} = \sum_i (\hat{\mathbf{y}}_i - \mathbf{y}_i)^2. \tag{6}$$

The objective function for MTL-DNN training, C_{MTL} is derived by combining C_{VAD} and C_{FE} as given by

$$C_{MTL} = \lambda C_{VAD} + (1 - \lambda) C_{FE} \tag{7}$$

where λ is a trade-off parameter between the VAD and fea-

ture enhancement tasks.

One important characteristic of the MTL framework is that it only increases the training complexity. After the fine-tuning stage, the layers for the conventional VAD task are preserved while those parts that are relevant to only the subsidiary task are discarded. In the test stage, the same feedforward algorithm and decision rule to the conventional DNN-based VAD algorithm are applied to estimate the voice activity status.

3. Experiments

3.1 Experiments in Matched Noise Conditions

In order to evaluate the performance of the proposed algorithm, we conducted a set of VAD experiments. In the experiments, the {Airport, Babble, Car, Restaurant, Street, Subway, Train} noisy speech data was taken from the Aurora2 database [12]. Each waveform was sampled at 8 kHz and the frame length was 25 ms with a frame-shift of 10 ms. The list of features for the DNN input used in the experiments is shown in Table 1. We compared the frame level accuracies of VAD obtained from the proposed algorithm with those from the conventional DNN-based VAD algorithm [5].

To train the DNNs, a set of noisy speech utterances with SNRs from -5 to 10 dB were used. 1001 utterances for each SNR and each noise type were randomly split into 300 utterances of training set, 300 utterances of validation set and 401 utterances of test set, respectively. The input features of the DNNs were normalized to have zero mean and unit variance. The DNNs were implemented using the Theano neural network toolkit [13].

The DNN with conventional training procedure was constructed by stacking 2 hidden layers of 1024 nodes. We ran 30 epochs for pre-training of each hidden layer to train the DNN. For Gaussian-Bernoulli RBMs, we fixed the learning rate to 0.001 while for Bernoulli-Bernoulli RBMs we fixed the learning rate to 0.01. For the fine-tuning stage, the learning rate started at 0.1. At the end of each epoch, if the frame accuracy on the development set decreased, the parameters of the DNN were returned to their values at the beginning of the epoch and the learning rate was exponentially decayed with a decaying factor of 0.8. This procedure was continued until the learning rate fell below 0.001. For both stages, we fixed the mini-batch size to 100.

The MTL-DNN was constructed by stacking one shared hidden layer and one separated hidden layer for each task with 1024 nodes each. The clean features for the feature enhancement task were normalized to have zero mean and unit variance. The MTL-DNN was trained with the same training configuration to that of the conventional DNN except the objective function in the fine-tuning stage was changed to (7). During the fine-tuning stage, we fixed λ to 0.9.

Tables 2 and 3 show the frame accuracies of the DNNs with or without the MTL framework in matched noise conditions. From the results, we can see that the proposed algorithm showed slightly better performance than the conventional DNN-based VAD. The performance difference between the two DNNs in matched noise condition was not significant since the DNN can learn the mapping between the noisy speech features and the corresponding voice activity status without any denoising function when the background noises match.

Table 1 Feature structures extracted from noisy and clean speech waveform.

Feature	Dimension	Feature	Dimension
Pitch	1	$MFCC_{16}$	20
DFT	16	LPC	12
DFT_8	16	RASTA-PLP	17
DFT_{16}	16	AMS	135
MFCC	20	**Total**	**273**
$MFCC_8$	20		

Table 2 Frame Accuracies (%) of the conventional DNN-based VAD in matched noise conditions.

	SNR (dB)				
	-5	0	5	10	Average
Street	73.45	81.57	87.22	90.45	83.17
Airport	76.19	84.17	89.89	93.38	85.91
Car	79.18	86.84	90.91	93.74	87.67
Babble	73.57	83.24	89.16	93.00	84.74
Train	76.22	83.98	89.92	93.09	85.80
Restaurant	69.93	80.87	87.78	92.15	82.68
Subway	69.77	79.51	87.39	91.62	82.07
Average	74.04	82.88	88.9	94.49	**84.58**

Table 3 Frame Accuracies (%) of the MTL-DNN-based VAD algorithm in matched noise conditions.

	SNR (dB)				
	-5	0	5	10	Average
Street	73.93	81.71	86.95	90.57	83.29
Airport	77.35	84.68	90.10	93.46	86.40
Car	79.60	86.79	91.04	93.92	87.84
Babble	74.72	84.07	89.78	93.28	85.46
Train	75.05	82.89	89.55	93.17	85.17
Restaurant	70.61	81.20	88.14	92.36	83.08
Subway	69.09	78.84	86.91	91.53	81.59
Average	74.33	82.89	88.92	92.61	**84.69**

3.2 Experiments in Mismatched Noise Conditions

We also evaluated the performance of the DNNs when the noises were mismatched between the training and test phases. In this experiment, the DNNs were trained with {Airport, Babble, Car, Train} noisy speech data and tested with {Street, Restaurant, Subway} noisy speech data. For each SNR and each noise in {Airport, Babble, Car, Train} data, 600 utterances were assigned to the training set and 401 utterances were assigned to the validation set. For each SNR and each noise in {Street, Restaurant, Subway} data, 401 utterances were used as the test data.

Tables 4 and 5 show the frame accuracies of the DNNs with or without MTL framework in mismatched noise conditions. From the results, we can see that the proposed algorithm outperformed the conventional DNN-based VAD al-

Table 4 Frame Accuracies (%) of the conventional DNN-based VAD algorithm in mismatched noise conditions.

	SNR (dB)				
	-5	0	5	10	Average
Street	68.84	80.06	87.39	91.37	81.92
Restaurant	62.58	72.43	82.21	89.96	76.80
Subway	57.25	58.83	63.09	70.54	62.43
Average	62.89	70.44	77.56	83.96	**73.71**

Table 5 Frame Accuracies (%) of the MTL-DNN-based VAD algorithm in mismatched noise conditions.

	SNR (dB)				
	-5	0	5	10	Average
Street	71.50	81.87	88.40	91.90	83.42
Restaurant	63.18	73.90	84.37	90.94	78.10
Subway	57.66	60.80	67.96	77.85	66.07
Average	64.11	72.19	80.24	86.90	**75.86**

gorithm. These results show that the MTL framework improves the robustness of the DNN especially in mismatched noise conditions.

4. Conclusions

In this letter, we have proposed an MTL framework for robust DNN-based VAD algorithm in mismatched noise conditions. From the results, it has been shown that the proposed algorithm outperformed the conventional algorithm in mismatched noise conditions. The future work will focus on employing the MTL framework with a set of tasks which represent other types of speech information, such as speaker and phonetic identities.

Acknowledgments

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012R1A2A2A01045874), and by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2015-H8501-15-1016) supervised by the IITP(Institute for Information & communications Technology Promotion).

References

[1] J.W. Shin, J.-H. Chang, and N.S. Kim, "Voice activity detection based on statistical models and machine learning approaches," Comput. Speech Lang., vol.24, no.3, pp.515–530, July 2010.
[2] D. Enqing, L. Guizhong, Z. Yatong, and Z. Xiaodi, "Applying support vector machines to voice activity detection," Proc. Int. Conf. Signal Process., Beijing, China, vol.2, pp.1124–1127, Aug. 2002.
[3] J. Wu and X.-L. Zhang, "Efficient multiple kernel support vector machine based voice activity detection," IEEE Signal Process. Lett., vol.18, no.8, pp.466–469, Aug. 2011.
[4] D. Ying, Y. Yan, J. Dang, and F.K. Soong, "Voice activity detection based on an unsupervised learning framework," IEEE Trans. Audio, Speech, Lang. Process., vol.19, no.8, pp.2624–2633, Nov. 2011.
[5] X.-L. Zhang and J. Wu, "Deep belief networks based voice activity detection," IEEE Trans. Audio, Speech, Lang. Process., vol.21, no.4, pp.697–710, April 2013.
[6] R.A. Caruana, "Multitask connectionist learning," Proc. 1993 Connectionist Models Summer School, pp.372–379, 1993.
[7] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in Neural Networks: Tricks of the Trade, eds. G. Montavon, G.B. Orr, and K.-R. Müller, pp.437–478, Springer-Verlag, Berlin, 2012.
[8] P. Bell and S. Renals, "Regularization of context-dependent deep neural networks with context-independent multi-task training," Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process., Brisbane, Australia, pp.4290–4294, April 2015.
[9] R. Giri, M.L. Seltzer, J. Droppo, and D. Yu, "Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning," Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process., Brisbane, Australia, pp.5014–5018, April 2015.
[10] D. Chen and B.K.-W. Mak, "Multitask learning of deep neural networks for low-resource speech recognition," IEEE/ACM Trans. Audio, Speech, Lang. Process., vol.23, no.7, pp.1172–1183, July 2015.
[11] G.E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural Comput., vol.18, no.7, pp.1527–1554, July 2006.
[12] D. Pearce and H.-G. Hirsch, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," Proc. ISCA ITRW ASR, pp.29–32, Paris, France, Sept. 2000.
[13] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: A CPU and GPU math compiler in Python," Proc. Scientific Comput. with Python Conf. (SciPy), pp.3–9, Austin, Texas, July 2010.