LETTER

Supervised Denoising Pre-Training for Robust ASR with DNN-HMM

Shin Jae KANG[†], **Kang Hyun LEE**[†], ***Nonmembers***, ***and*** **Nam Soo KIM**[†a)], ***Member***

SUMMARY In this letter, we propose a novel supervised pre-training technique for deep neural network (DNN)-hidden Markov model systems to achieve robust speech recognition in adverse environments. In the proposed approach, our aim is to initialize the DNN parameters such that they yield abstract features robust to acoustic environment variations. In order to achieve this, we first derive the abstract features from an early fine-tuned DNN model which is trained based on a clean speech database. By using the derived abstract features as the target values, the standard error back-propagation algorithm with the stochastic gradient descent method is performed to estimate the initial parameters of the DNN. The performance of the proposed algorithm was evaluated on Aurora-4 DB, and better results were observed compared to a number of conventional pre-training methods.

key words: *deep neural networks (DNNs), pre-training, denoising, back-propagation, robust speech recognition*

1. Introduction

Recently, deep neural networks (DNNs) have become one of the most popular techniques in the vast field of machine learning. Due to their powerful capability in nonlinear description between the input and the target values, the DNNs have outperformed many other conventional techniques in various tasks. This DNN's capability has also been applied to the environment-robust techniques for automatic speech recognition (ASR). Particularly, in the robust ASR area, conventional environment-robust techniques usually necessitate some specific models or formulations to account for the nonlinear relationship between the clean and noisy speech processes in an appropriate signal domain [1]–[3]. In constrast, the DNNs have the advantage that they can directly learn an arbitrary unknown relationship between the input and the target values without any specific assumption. Consequently, they have brought a better performance gain than the conventional approaches [4]. A more complicated input-target relationship can be easily learned by using wider and deeper neural network architectures with a sufficient amount of training data.

Since DNN is a highly nonlinear and non-convex model, its performance usually depends on the initial parameter setting for training. This issue has been possibly resolved through a number of unsupervised or supervised pre-training methods. For the unsupervised methods, generative pre-training algorithm for the restricted Boltzmann machines (GEPT) [5], greedy layer-wise unsupervised pre-training using autoencoder [6] and stacked denoising autoencoder (SDAE) [7] were proposed. A core idea of these algorithms is to learn a nonlinear representation of the input data one level at a time using unsupervised feature learning. In the case of SDAE, the pre-training module takes the noisy features as an input and then tries to recover the original clean features by minimizing the cross-entropy loss or the squared error loss between the reconstructed features and the original clean features.

In the class of the supervised methods, greedy layer-wise supervised pre-training (GLPT) [6] and discriminative pre-training (DPT) [8] methods were proposed. These methods first train the DNN with one hidden layer using the target labels discriminatively, then insert another hidden layer between the trained hidden layer and the output layer and again discriminatively train the network to convergence. This procedure is repeated until the desired number of hidden layers are all trained. A hybrid pre-training algorithm combining GEPT and GLPT was also introduced [9]. These pre-training techniques can potentially bring the DNN weights to a relatively good initial point for converging to a better local optimum.

The above mentioned pre-training techniques can be also applied to robust ASR. In order to initialize the DNN in adverse environments, not only the clean features but also the corrupted features are used as an input of the DNN, which is common in the robust ASR area. In a sense, this approach can be regarded as a multi-condition training technique. The parameters of the DNN are learned to describe the hidden representation of the multi-condition data set. As the depth of the DNN gets deeper, more abstract features can appear at higher layers. More abstract concepts are generally considered more robust to most local variations of the inputs. Learning these invariant features has been a long-standing goal in pattern recognition [10].

In this letter, we propose a novel supervised denoising pre-training technique for the DNN-hidden Markov model (HMM) systems for robust speech recognition in adverse environments. In the proposed approach, our aim is to initialize the DNN parameters such that they yield abstract features robust to acoustic environment variations. In order to achieve this, we first derive the abstract features from an early fine-tuned DNN model which is trained based on a clean speech database. By using the derived ab-

Manuscript received May 21, 2015.
Manuscript revised August 7, 2015.
Manuscript publicized September 7, 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.2015EDL8118

stract features as the target values, the standard error back-propagation (BP) algorithm with the stochastic gradient descent method is performed to estimate the initial parameters of the DNN. The performance of the proposed algorithm was evaluated on Aurora-4 DB, and better results were observed compared to a number of conventional pre-training methods.

2. Supervised Denoising Pre-Training

The proposed approach is called a supervised denoising pre-training technique. In the proposed technique, the initial parameters of the DNN for noisy inputs are learned so as to describe the most abstract features obtained when the corresponding clean features are applied. If this is achieved, the DNN is capable of extracting abstract representations, i.e., hidden node values robust against the interfering noises.

For this approach, we need an auxiliary DNN with the same structure, which is fully-trained based solely on a clean speech database. It can be obtained using a set of clean training data and the corresponding target labels which are the posterior probabilities of the tied-state triphones (senones) of the HMM [11]. The hidden nodes of this auxiliary DNN are considered to form abstract representations of the clean speech features which are not affected by interfering noises. Since the nodes of the top hidden layer are considered to possess the most abstract characteristics of the clean speech features, we only focus on the top hidden layer of the auxiliary DNN when creating the target abstract representation in this work.

Let $\{\mathbf{W}_c^l, \mathbf{b}_c^l | (\mathbf{o}^c, \mathbf{d}), 0 < l \leq L\}$ denote the parameters of the auxiliary DNN estimated from a clean training data $\mathbf{o}^c$ with the corresponding target labels $\mathbf{d}$. Here $\{\mathbf{W}_c^l\}$ represent the weights connecting the l-th layer with the $(l-1)$-th layer and $\{\mathbf{b}_c^l\}$ are the biases of the nodes at the l-th layer. For more detail on the basic structures and operations at each node of the DNNs, please refer to [11]. For simplicity, we denote the input layer as layer 0 and the output layer as layer L for an $(L+1)$-layer DNN. Also let $\{\mathbf{W}_m^l, \mathbf{b}_m^l | (\mathbf{o}^m, \mathbf{d}), 0 < l \leq L\}$ be the parameters of the main DNN which will be trained based on a multi-condition data $\mathbf{o}^m$ with the target labels $\mathbf{d}$. Note that $\mathbf{o}^c$ and $\mathbf{o}^m$ form a stereo database, i.e., simultaneous recordings obtained in both the clean and corrupted conditions, and the desired target labels are the same in both data. In our approach to pre-training, the parameters $\{\mathbf{W}_m^l, \mathbf{b}_m^l\}$ are initialized such that they yield the abstract representation at the $(L-1)$-th hidden layer as close as possible to those obtained at the same layer of the auxiliary DNN which was fed with clean speech feature. Providing the last hidden layer values of the auxiliary DNN as the target enables to estimate $\{\mathbf{W}_m^l, \mathbf{b}_m^l\}$ with the use of the BP algorithm. For back-propagating the errors between the activated values obtained from the main DNN and the target abstract features derived from the auxiliary DNN, we employ the mean square error (MSE) as the objective function. If the number of training samples is T, the objective function J_{MSE} is given by

$$J_{MSE} = \frac{1}{T} \sum_{t=1}^{T} \left[\frac{1}{2} \| \mathbf{v}_{m,t}^{L-1} - \mathbf{v}_{c,t}^{L-1} \|^2 \right] \quad (1)$$

where $\mathbf{v}_{m,t}^{L-1}$ and $\mathbf{v}_{c,t}^{L-1}$ respectively indicate the t-th activation vectors obtained from the main and auxiliary DNN at the $(L-1)$-th layer, and $\| \cdot \|$ means Euclidean norm. It is very important to note that $\mathbf{v}_{m,t}^{L-1}$ in (1) is derived from $\mathbf{o}^m$ while $\mathbf{v}_{c,t}^{L-1}$ is derived from the auxiliary DNN when the clean speech feature $\mathbf{o}^c$ is applied. The proposed method can be modified to reproduce all the hidden node activations of the auxiliary DNN by treating each hidden activation as the target value. From a number of preliminary experiments, we have found that using only the last hidden layer as the target values shows a slightly better performance than using all the hidden layers. After $\{\mathbf{W}_m^l, \mathbf{b}_m^l\}$ are initialized as above, a usual discriminative fine-tuning algorithm is performed [11].

3. Experiments

The performance of the proposed method was evaluated on Aurora-4 DB which is widely used in the robust speech recognition area. The proposed method was compared with the following conventional pre-training approaches: GEPT [5], GLPT [6], DPT [8] and SDAE [7]. Furthermore, the performance evaluation with the dropout technique [12] which is widely used in the DNN training was also investigated.

3.1 Aurora-4 DB

Aurora-4 DB [13] was made using 5k-word vocabulary based on the Wall Street Journal (WSJ) DB. The WSJ data were recorded with a primary Sennheiser microphone and with a secondary microphone in parallel. The corpus has two training sets: clean- and multi-condition. Both clean- and multi-condition sets consist of the same 7138 utterances from 83 speakers. The clean-condition set consists of only the primary Sennheiser microphone data. One half of the utterances in the multi-condition set were recorded by the primary Sennheiser microphone and the other half were recorded using one of 18 different secondary microphones. Both halves include 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 evaluation was conducted on the test set consisting of 330 utterances from 8 speakers. This test set was recorded by the primary microphone and a number of secondary microphones. These two sets were then each 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 Set_A, Set_B, Set_C and Set_D, respectively. Note that the types of noises are common across training and test sets but the SNRs of the data are not.

For the validation test, we used the development set in Aurora-4 DB consisting of 330 utterances from 10 speakers not included in the training and test set speakers. A total of 14 sets with the same conditions as the test set were configured.

3.2 Feature Extraction and GMM-HMM System

We used the Kaldi speech recognition toolkit [14] for feature extraction, acoustic modeling of ASR, DNN training and ASR decoding. The feature was extracted with the default configuration of Kaldi. According to that configuration, 23-dimensional log mel filterbank (LMFB) features were calculated and 13-dimensional mel-frequency cepstral coefficients (including C_0) with their first and second derivatives were extracted for the Gaussian mixture model (GMM)-HMM recognizer. The cepstral mean normalization algorithm was applied for each speaker.

In order to provide the target alignment information for the discriminative DNN training, we built a clean-condition GMM-HMM system with 2009 senones and 15028 Gaussian mixtures in total. The target senone labels of the DNN-HMM system were obtained over the clean-condition training data. As for the language model, we applied the standard 5k open bi-gram model for decoding.

3.3 DNN Structures

For the auxiliary and main DNN training, we applied five hidden layers with 2048 nodes. As for the input features of the DNNs, we used the LMFB features due to their good performance demonstrated in the previous studies. The input features consisted of 11 frames (5 frames on each side of the current frame) context window of 23 dimensional LMFB features with their first and second order derivatives, which resulted in the input dimension of 759. The input features of the DNNs were normalized to have zero mean and unit variance.

For training the auxiliary DNN using the clean-condition training data, GEPT was carried out to initialize the DNN parameters as described in [15]. For the supervised fine-tuning, the initial learning rate of 0.008 with the same 256 minibatch size as the pre-training was used for the DNN training. The errors between the DNN outputs and the target senone labels were calculated according to the cross-entropy criterion [11].

For initializing the main DNN parameters using the proposed method, GEPT was first conducted using the multi-condition training data for the main DNN and then supervised fine-tuning was performed using the abstract features derived from the auxiliary DNN as the target values with the initial learning rate of 0.0005. The errors between the last hidden node activations of the main DNN and the target abstract features derived from the auxiliary DNN were calculated as in (1). After initializing the parameters of the main DNN and adding an output layer on the top of the network, the discriminative fine-tuning with the senone targets was performed with the initial learning rate of 0.008. In order to speed up training, we applied the learning rate scheduling scheme and stop criteria presented in [15].

GLPT, DPT and SDAE methods used the same multi-condition training data as the proposed pre-training techniques. GLPT and DPT were implemented using the senone target labels in a layer-wise manner until the desired number of hidden layers was reached with initial learning rate of 0.008 and 0.001, respectively. DPT is similar to GLPT but differs in that the latter only updates the newly added hidden layer while in the former all layers are jointly updated each time when a new hidden layer is added. For SDAE, the initial parameters of the DNN were obtained by minimizing the squared error loss between the reconstructed features from the multi-condition training data and those from the clean-condition training data, which was performed in the layer-wise greedy training mode with initial learning rate of 0.0001 [7]. After initializing the parameters of each DNN using GLPT, DPT and SDAE, the usual discriminative fine-tuning algorithm was performed with initial learning rate of 0.008. The same learning rate scheduling scheme and stop criteria mentioned above were applied.

As one of the well-known regularization techniques, dropout was also applied. Dropout is a method that improves the generalization ability of the DNNs. It can easily be implemented by randomly dropping the input and hidden neuron units. As pointed out by Hinton et al. [12], dropout can be considered as a bagging technique that averages over a large amount of models with shared parameters of the DNN. Dropout was applied to the fine-tuning stages. In all the experiments, we used the dropout retention rate of 0.95 at both the input and hidden layers, which showed the best performance in our experiments.

Table 1 WERs (%) on the auxiliary DNN-HMM system.

Method	Set_A	Set_B	Set_C	Set_D	Average
GEPT	7.12	47.55	42.91	65.98	52.23

3.4 Performance Evaluation

We compared our proposed method with the conventional pre-training approaches on Aurora-4 DB. For convenience, the proposed method is denoted by SDPT when demonstrating the experimental results. Table 1 shows the word error rates (WERs) obtained with the auxiliary DNN-HMM system which was used to generate the target abstract features. Table 2 shows the WERs of the main DNN-HMM system built with various pre-training techniques. In Table 2, 'Random' denotes no pre-training with which the parameters were randomly drawn from a Gaussian $\mathcal{N}(0, 0.01)$. Furthermore, WERs obtained with the use of the dropout are shown in Table 3. In our experiments, random initialization showed lower performance than any other pre-training

Table 2 WERs (%) on the main DNN-HMM system according to several pre-training methods.

Method	Set_A	Set_B	Set_C	Set_D	Average
Random	8.82	14.09	14.81	26.91	19.26
SDAE	7.77	11.88	12.42	23.72	16.70
DPT	8.00	11.92	12.68	23.29	16.57
GLPT	7.83	11.73	11.97	23.02	16.31
GEPT	7.81	11.71	12.27	22.71	16.18
SDPT	7.42	10.82	11.69	22.49	**15.64**

Table 3 WERs (%) on the main DNN-HMM system according to several pre-training methods with dropout.

Method	Set_A	Set_B	Set_C	Set_D	Average
Random+dropout	8.28	12.89	13.08	25.02	17.77
SDAE+dropout	7.96	11.81	11.71	22.84	16.45
DPT+dropout	7.83	11.88	11.92	23.18	16.43
GLPT+dropout	7.60	11.63	11.58	22.33	15.98
GEPT+dropout	7.75	11.59	11.34	22.47	15.96
SDPT+dropout	7.19	10.89	10.89	21.74	**15.27**

methods. After applying the dropout, the performance of all the pre-training techniques was improved consistently. From the results, we can see that SDPT outperformed all the other pre-training techniques in all the tested conditions and the dropout gave additional gains.

4. Conclusion

In this letter, we have proposed a novel supervised denoising pre-training technique for the DNN robust to noisy input variations. From the experimental results, we have found that the proposed method was effective for enhancing the recognition performance in adverse conditions. Future study may focus on reducing the training time of the proposed method.

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] L. Deng, J. Droppo, and A. Acero, "Recursive estimation of nonstationary noise using iterative stochastic approximation for robust speech recognition," IEEE Trans. Audio, Speech, Language Process., vol.11, no.6, pp.568–580, Nov. 2003.
[2] N.S. Kim, T.G. Kang, S.J. Kang, C.W. Han, and D.H. Hong, "Speech feature mapping based on switching linear dynamic system," IEEE Trans. Audio, Speech, Language Process., vol.20, no.2, pp.620–631, Feb. 2012.
[3] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, "HMM adaptation using vector Taylor series for noisy speech recognition," Proc. ICSLP, pp.869–872, Beijing, China, Oct. 2000.
[4] M. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," Proc. ICASSP, Vancouver, Canada, pp.7398–7402, May 2013.
[5] 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.
[6] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," Proc. NIPS, pp.153–160, Vancouver, Canada, Dec. 2006.
[7] X.-L. Zhang and J. Wu, "Denoising deep neural networks based voice activity detection," Proc. ICASSP, Vancouver, Canada, pp.853–857, May 2013.
[8] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," Proc. ASRU, Hawaii, USA, pp.24–29, Dec. 2011.
[9] H. Larochelle and Y. Bengio, "Classification using discriminative restricted Boltzmann machines," Proc. ICML, pp.536–543, 2008.
[10] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," IEEE Trans. Pattern Anal. Mach. Intell., vol.35, no.8, pp.1798–1828, Aug. 2013.
[11] 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.
[12] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," CoRR, vol.abs/1207.0580, 2012.
[13] 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.
[14] 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.
[15] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," Proc. Interspeech, pp.2345–2349, Lyon, France, Aug. 2013.