DISCRIMINATIVELY TRAINED JOINT SPEAKER AND ENVIRONMENT REPRESENTATIONS FOR ADAPTATION OF DEEP NEURAL NETWORK ACOUSTIC MODELS

Maofan Yin[1], *Sunil Sivadas*[2], *Kai Yu*[1], *Bin Ma*[2]

[1]Key Lab. of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering SpeechLab., Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
[2]Human Language Technology Lab., Institute for Infocomm Research, Singapore

{__root__,kai.yu}@sjtu.edu.cn {sivadass,mabin}@i2r.a-star.edu.sg

ABSTRACT

A recent trend in normalization of factors extraneous to a speech recognition task has been to explicitly introduce features related to the unwanted variability in the training of Deep Neural Networks (DNN). Typically, this is done by either perturbing the training set with models of these extraneous factors such as vocal tract length and environmental noise or augmenting the conventional spectral features with auxiliary information such as i-vector, noise spectrum, etc. Another emerging approach is to derive low dimensional representations of the factors from the hidden layers of DNN and use it for normalization of the acoustic model. Almost all of these approaches focus on either speaker or environment normalization. In this paper we propose a novel approach for estimating a compact joint representation of speakers and environment by training a DNN, with a bottleneck layer, to classify the i-vector features into speaker and environment labels by Multi-Task Learning (MTL). Another novelty is to learn this compact representation while learning to map the i-vector of a noisy utterance into its corresponding clean speaker i-vector and noise-only i-vector. Experiments were conducted on an artificially noise-corrupted version of the WSJ corpus. The proposed compact joint speaker-environment representations show promising gains.

Index Terms— i-vector, deep neural network, speaker adaptation, multi-task learning, noise robustness

1. INTRODUCTION

Over the past decade, there have been tremendous advances in the accuracy of large vocabulary speech recognition systems. Even though this is largely driven by the amount of training data and increase in computational capabilities, the performance improvements are largely limited to clean and moderately noisy test conditions. Recently, there has been a lot of focus on normalization of speaker and environment variability in DNN-based acoustic modeling. fMLLR is an effective feature-transform-based approach for speaker adaptation [1]. Speaker adaptation of DNNs by training a hidden layer as a discriminative feature transform [2] and Cluster Adaptive Training [3] are structured-model-based methods. Multi-condition training and data augmentation [4] are state of the art methods where the training set incorporates possible acoustic factors that can be encountered in the testing scenario. Another strategy for normalization is to augment the DNN input with auxiliary features that carry speaker and environment information [5, 6, 7]. Effectiveness of the last approach lies in finding good speaker and environment representations.

This paper focuses on deriving a joint representation of speaker and environment that can be used to augment the DNN input. Typically, stacked spectral features are used to train a DNN with a bottleneck layer, where the output of the bottleneck layer is used as the speaker representation [8, 9]. Since the speaker-specific information is not represented in short-term spectral features, a super vector derived from the phonetic clustering of bottleneck features is examined in [10]. In this paper we use i-vectors, instead of stacked short-term spectral vectors, as input to train a DNN that estimates the Joint Speaker and Environment Representation (JSER) . We explore three forms of JSER: firstly by trying to predict the clean speaker i-vector and pure noise i-vector from the noisy utterance i-vector, which is inspired by the i-vector factorization approach [11]; secondly, using MTL to train the JSER-DNN with speaker and environment labels; and finally, the JSER-DNN is trained to predict the joint speaker-noise label.

2. SPEAKER AND ENVIRONMENT NORMALIZATION OF DNN

Our goal is to extract a low dimensional representation of speaker and environment and augment this feature with acoustic features at the input of a DNN acoustic model. The speaker and environment representations are typically derived from the signal using spectral information [7] or using

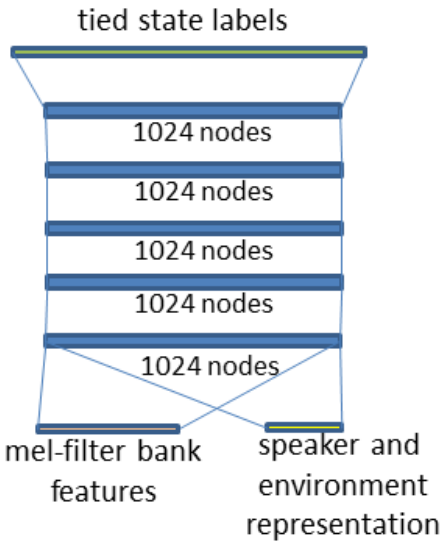

Fig. 1. Speaker and environment normalization of the DNN acoustic model.

statistical methods such as i-vector [5]. i-vector is a low dimensional representation of the acoustic variability related to speakers, environment, dialects, etc., rather than the phonetic variability [12]. Given a Gaussian Mixture Model (GMM), the corresponding speaker-specific mean super-vector $M(s)$, for speaker s, can be approximated as

$$M(s) = m + Tw(s) \quad (1)$$

where m is the mean super-vector from the GMM-UBM, T is the low-rank total variability matrix and $w(s)$ is the low-dimensional i-vector for speaker s [13].

In [14] the authors have shown that using a DNN to transform i-vectors before adding them to acoustic features can bring further improvement. In this paper we take a similar approach where i-vectors are discriminatively transformed before being augmented to the acoustic features. The following section explains the proposed method in detail.

3. PROPOSED DISCRIMINATIVE JOINT SPEAKER-ENVIRONMENT REPRESENTATION

Figure 1 shows the approach we adopt for speaker and environment normalization of the DNN acoustic model. Effectiveness of the normalization approach in Figure 1 depends on the quality of the speaker and environment representation. It is desirable that the representation can discriminate speakers and environment reliably. By blending the concepts of MTL [15, 16] and deep auto-encoder [17], we examine three different forms of Joint Speaker-Environment Representation (JSER). Figure 2 depicts the architecture of the proposed methods. Multi-task learning is an approach that aims at improving the performance on multiple tasks by jointly learning classifiers for multiple tasks. Usually better representation and common knowledge among different tasks are learned, hence it achieves higher classification accuracy than single task learning. MTL-DNN is proposed to address the multi-label classification problem in [18], where an instance may have multiple labels and the goal is to figure out all the labels of an unseen instance.

3.1. MTL-MSE-JSER: Predicting speaker and noise i-vectors

In [11], factorizing i-vectors carrying information about both speaker and environment factors into separate speaker i-vectors and noise i-vectors was shown to give better generalization for unseen environments. In the proposed method we train a transform to learn the mapping from noisy utterance i-vectors to clean speaker i-vectors and pure noise i-vectors. For the transform we use an MTL-DNN with a bottleneck layer as shown in figure 2(a). The MTL-DNN with linear output layers is trained to minimize the Mean Squared Error (MSE) between the target i-vectors and predicted i-vectors. This method can also be viewed as a method to enhance or de-noise noisy i-vectors. The low dimensional bottleneck layer encodes discriminative information about the target tasks [17].

3.2. MTL-CE-JSER: Predicting speaker and noise labels

The i-vector feature carries acoustic variability related to the speaker and the environment in which the utterance is recorded. Instead of using clean speaker and pure noise i-vectors as targets, in this method we train an MTL-DNN to classify the noisy i-vector to predict speaker and noise labels. As shown in figure 2(b), this is essentially an i-vector-based classifier for speakers and noise, where the MTL-DNN with softmax output layers is trained to minimize the cross-entropy between the target labels and the predicted labels. By choosing noise classification as an auxiliary task, the feature encoding in the bottleneck layer is discriminatively trained to represent both speakers and noise.

3.3. JTL-CE-JSER: Predicting joint speaker-noise labels

In the final method, as shown in figure 2(c), we train a DNN to predict the joint speaker-noise labels from the noisy utterance i-vector. There are $S \times N$ target classes, where S is the number of speakers and N is the number of environment types in the training data. The DNN has a single softmax output layer and the objective function is cross-entropy between target labels and predicted labels. Joint Task Learning (JTL) could be harder than MTL because the classifier has to learn all possible combinations of speakers and noise presented in the training set.

4. EXPERIMENTS

Experiments were conducted on a corrupted WSJ database. We used the 84 speaker WSJ0 subset for training the acoustic

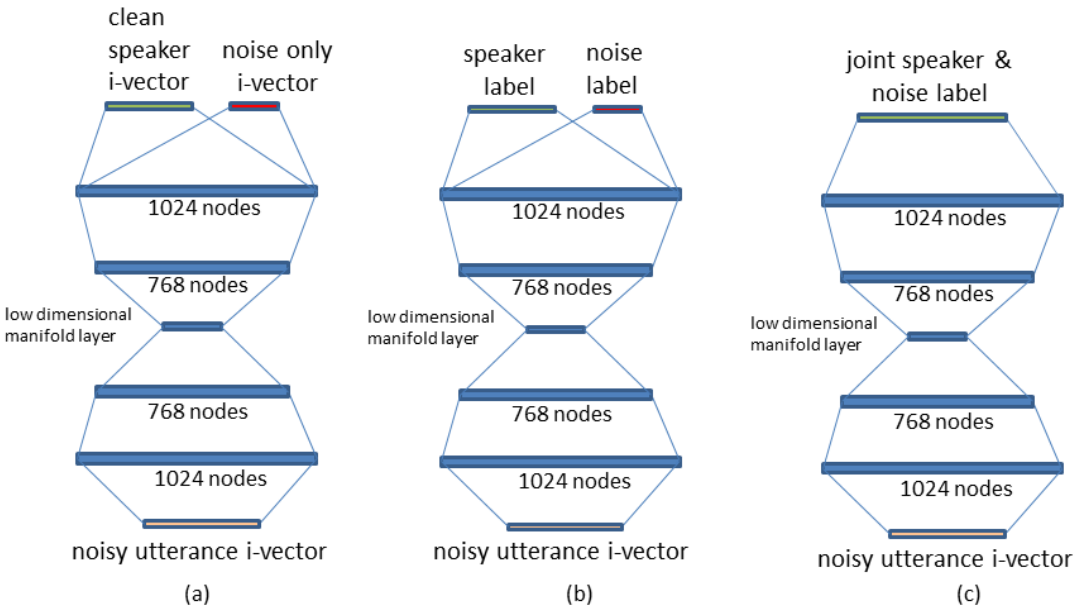

Fig. 2. Discriminatively trained low dimensional subspace from noisy utterance i-vectors. (a) MTL-MSE-JSER: Learning the mapping from noisy utterance i-vectors to corresponding speaker and noise i-vectors by MTL. (b) MTL-CE-JSER: Training a classifier to predict speaker and noise labels from noisy utterance i-vectors by MTL. (c) JTL-CE-JSER: Training a classifier to predict the joint speaker and noise labels from noisy utterance i-vectors.

model and both WSJ0 and WSJ1 for training the JSER transforms. To simulate the background noise, 8 different types of noise (restaurant, street, supermarket, food-court, living room, mall, taxi and gym) were added to the clean waveforms at different SNRs. Each noise recording was about half an hour long. Each clean waveform in the training set was mixed with a random noise segment equal to the duration of the waveform. We created two different noise corrupted databases, one for i-vector extraction and one for acoustic model training. Trained models were evaluated on corrupted *eval92, dev93* and *eval93* 5K closed vocabulary test sets. The same 8 noise types at random SNRs between 5 dB and 20 dB were added to the clean test sets. A trigram language model was used in decoding.

4.1. Speaker and noise i-vector extraction

For every noise type we corrupted the WSJ0 and WSJ1 training set produced by 283 speakers, at 8 different SNRs, from 5 dB to 20 dB in steps of 2 dB. This resulted in a noisy database that is 64 times the size of clean database. Gender dependent UBM-GMMs with 2048 mixture components were trained on this set. UBMs were trained on 13 MFCC coefficients appended with delta and delta-delta coefficients. The features were normalized to zero mean and unit variance over each utterance. Utterance i-vectors and speaker i-vectors were extracted using all the utterances. For the noise model, a UBM-GMM with 512 mixture components was trained. For computing the pure noise i-vectors, the long noise recordings were randomly segmented into many 20-second chunks and MFCC features were extracted. Using the noise and speaker labels for each utterance i, we got triplets of $\{w(i), w(s_i), w(n_i)\}$ and $\{w(i), s_i, n_i\}$ where $w(i)$ is the utterance i-vector, $w(s_i)$ is the speaker i-vector, $w(n_i)$ is the pure noise i-vector, s_i is the speaker label and n_i is the noise label. These features were used for training the JSER-DNNs as shown in figure 2.

4.2. Training speaker and environment representations

Figure 2 shows the topology of JSER-DNNs. The DNNs differ only in the design of the output layer and the number of nodes in the linear bottleneck layer. We fixed the dimensionality of the input noisy utterance i-vectors at 100, that of clean speaker i-vectors at 50 and of pure noise i-vectors at 10. For the MTL-CE-JSER, there are 283 speaker labels and 8 noise labels and for the JTL-CE-JSER there are $283 \times 8 = 2264$ output nodes. Note that JSER-DNNs only need utterance-level features (i.e., noisy i-vectors), which implies the amount of training data is significantly less compared to many other models that use frame-level input, such as spectral and MFCC features [8, 9]. Table 1 gives the performance of JSER-DNNs in terms of MSE and classification accuracy for speakers and noise. We use 3% of the data set for cross-validation. The DNN is initialized using the RBM pre-training method [17] and fine-tuned using back-propagation. It can be seen that the JSER-DNNs are able to classify the speakers and environment types with high accuracy from noisy i-vector features. We hypothesize that activations of the bottleneck layer of a well trained JSER-DNN encode discriminative information about speaker and environment factors.

Multi-Task Learning	Speaker		Environment	
	Train	CV	Train	CV
MTL-MSE-JSER (60)	0.0501	0.0633	0.0931	0.1337
MTL-CE-JSER (60)	99.28	97.39	93.94	89.50
Joint-Task Learning	Spk. × Env.			
	Train		CV	
JTL-CE-JSER (60)	93.02		80.62	

Table 1. Speaker and noise classification performance of JSER-DNNs. For MTL-MSE-JSER, the numbers are MSE values and for the rest they are classification accuracies in percentage. The number in brackets is the dimensionality of the bottleneck layer.

	dev93	eval92	eval93
multi-condition	14.08	8.31	11.14
i-vector (25)	13.90	**7.73**	11.40
i-vector (100)	14.38	8.09	11.22
MTL-MSE-JSER (60)	13.72	8.07	11.06
MTL-CE-JSER (60)	**13.34**	8.37	**9.89**
JTL-CE-JSER (60)	15.36	9.47	11.89

Table 2. Word error rates for various speaker and environment representations. The number in brackets is the dimensionality of the representation.

4.3. DNN acoustic model

To train the DNN acoustic model, a WSJ0 subset with 84 speakers is corrupted with all 8 noise types at random SNRs between 10 dB and 20 dB to create a multi-condition training set. Including clean condition, this results in 9 environment conditions. The corrupted training set has the same distribution of environment conditions for every speaker and has the same size as the original clean training set. Note that training and test sets have the same noise types but different SNR ranges. Since the pure noise recordings are much longer than the clean waveforms, it is highly unlikely to encounter the same noise segment in training and test sets. 13 MFCC, delta and delta-delta features normalized by mean and variance over the utterance are computed. 11 frames of temporal context were used at the input of DNN with the topology shown in figure 1. The tied-state labels are obtained from an MMI trained GMM-HMM. DNN is initialized using RBM pre-training method [17] and fine-tuned using back-propagation. DNN acoustic modelling is performed using the Kaldi toolkit [19].

4.4. ASR results with normalization

Table 2 presents the word error rates (WER) of various systems on all test sets. Multi-condition DNN does not have any speaker and noise normalization. Utterance i-vector is appended to the baseline features to create the baseline speaker and environment system. The reason for using utterance-level i-vector adaptation instead of speaker-level adaptation is because under our corrupted WSJ experiment settings, it is found that speaker-level i-vector adaptation has worse performance. This may be due to the difference in distribution of noises and SNRs per speaker in training and test sets. We found that 25-dimensional i-vectors can provide better performance than 100-dimensional i-vectors in adaptation. This is consistent with the findings in [6]. As shown in the table, MTL-MSE-JSER outperforms the 100-dimensional baseline and the multi-condition baseline (without adaptation) in all three test sets. MTL-CE-JSER is even better on *dev93* and *eval93*. Although the 25-dimensional baseline has a better result on the *eval92* set than all others, MTL-CE-JSER has much better WERs on *dev93* and *eval93*, and therefore becomes the best in terms of the averaged WER of 10.53% on three test sets, whereas the 25-dimensional baseline has the averaged WER of 11.01%. JTL-CE-JSER causes degradation on all test sets. It may be due to the fact that it could be hard for the JTL-DNN to learn all combinations of speaker and noise labels present in the training set, and as seen in Table 1, the accuracy on the CV set is much worse than the one on training set, whereas MTL-CE-JSER seems to learn better according to its high accuracy on CV for both tasks. We experimented with dimensionality of the proposed representations, but observed that even though the frame accuracy of the DNN acoustic model was better on training and cross-validation sets, it did not translate into better word error rates.

5. CONCLUSIONS AND FUTURE WORK

We presented three novel methods for training discriminative joint speaker-environment representations from i-vectors. Firstly, we investigated multi-task learning to learn the representations. An MTL-DNN is trained to learn the mapping from noisy utterance i-vectors to clean speaker i-vectors and pure noise i-vectors. Secondly, an MTL-DNN for predicting speaker and noise labels from the noisy i-vector input is trained. Finally, a DNN is trained to predict the joint speaker-noise labels. All DNNs have a linear bottleneck layer. The proposed joint speaker-noise representations are the activation of the linear bottleneck layer. Except the JTL-CE-JSER, appending these representations as an additional feature at the input of the DNN acoustic model for speaker and environment normalization was found to be promising. In future we will explore additional auxiliary tasks relevant to multi-task learning, test on larger tasks and explore its application to noise robust speaker verification.

6. REFERENCES

[1] M.J.F. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer Speech and Language*, vol. 12.

[2] Kaisheng Yao, Dong Yu, Frank Seide, Hang Su, Li Deng, and Yifan Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 366–369.

[3] Tian Tan, Yanmin Qian, Maofan Yin, Yimeng Zhuang, and Kai Yu, "Cluster adaptive training for deep neural network," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4325–4329.

[4] Navdeep Jaitly and Geoffrey E Hinton, "Vocal tract length perturbation (vtlp) improves speech recognition," in *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, 2013.

[5] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2013, pp. 55–59.

[6] Andrew Senior and Ignacio Lopez-Moreno, "Improving dnn speaker independence with i-vector inputs," in *Proc. ICASSP*, 2014.

[7] Michael L Seltzer, Dong Yu, and Yongqiang Wang, "An investigation of deep neural networks for noise robust speech recognition," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 7398–7402.

[8] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Jorge Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4052–4056.

[9] Yuan Liu, Yanmin Qian, Nanxin Chen, Tianfan Fu, Ya Zhang, and Kai Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[10] Hengguan Huang and Khe Chai Sim, "An investigation of augmenting speaker representations to improve speaker normalisation for dnn-based speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4610–4613.

[11] Penny Karanasou, Yongqiang Wang, Mark JF Gales, and Philip C Woodland, "Adaptation of deep neural network acoustic models using factorised i-vectors," in *Proc. Interspeech*, 2014.

[12] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[13] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[14] Yajie Miao, Hao Zhang, and Florian Metze, "Speaker adaptive training of deep neural network acoustic models using i-vectors," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 11, pp. 1938–1949, 2015.

[15] Rich Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[16] Dongpeng Chen, Brian Mak, Cheung-Chi Leung, and Sunil Sivadas, "Joint acoustic modeling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 5592–5596.

[17] Geoffrey E Hinton and Ruslan R Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[18] Yan Huang, Wei Wang, Liang Wang, and Tieniu Tan, "Multi-task deep neural network for multi-label learning," in *2013 20th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2013, pp. 2897–2900.

[19] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *Proc. of IEEE ASRU*, 2011.