Speaker Adaptation with an Auxiliary Network

Leda Sarı, Mark Hasegawa-Johnson

Department of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, US
Beckman Institute, University of Illinois Urbana-Champaign, US
lsari2@illinois.edu, jhasegaw@illinois.edu

Abstract

Deep neural network based acoustic models are prone to variability in speech signals produced by different speakers. In this work, we present a speaker adaptation scheme that does not require supervision for the test speakers. We introduce an auxiliary adaptation network to a main senone classifier and aim at extracting the speaker dependent information explicitly through our auxiliary network. By subtracting this information from the hidden layer activations of the main network, we get speaker invariant features for the main senone classification task. Experiments performed on the TIMIT dataset demonstrate that by using the speaker dependent representation from the last hidden layer of the auxiliary network, we achieve around 1% absolute improvement in phone error rate as compared to an unadapted main network.

Index Terms: speech recognition, speaker adaptation, neural networks

1. Introduction

Inter-speaker variability is one of the sources of error in automatic speech recognition (ASR) systems. One direct way to solve this problem is to learn speaker dependent models but this requires large amounts of data per speaker which might be unavailable and it is hard to generalize the models to unknown test speakers. To deal with these problems, model adaptation techniques have been introduced.

Although deep neural networks (DNNs) are successfully used in ASR applications, their performance is still affected by the variability inherent in speech due to different speakers. Techniques proposed to alleviate this problem include using speaker-informed input features to the DNNs [1, 2], introducing speaker adaptive layers to speaker independent DNN [3], adapting the weights of the DNN based on speakers without changing the DNN structure [4] or extracting speaker invariant intermediate features by multitask learning [5].

In this study, we aim at extracting speaker dependent information through an auxiliary network and subtract this component from hidden layer activations so that we obtain a speaker independent representation. In order to extract the speaker dependent component of the activations, the auxiliary network is trained to reconstruct the speaker level averages of hidden layer activations of the main network which is a senone classifier. Therefore, our training objective is a combination of classification loss of the main network and mean-squared reconstruction error of the auxiliary network.

Miao et al. [7] also use an auxiliary network but they apply the offset only to the input features rather than the hidden layers of the main network. In [5], an adversarial multitask objective is used to extract speaker-invariant deep features. Here, we also use a multitask objective but our aim is to learn speaker dependent information explicitly through our adaptation network which is then used as an offset to get speaker invariant features. Veselỳ et al. [8] use summary vectors computed over the utterance as additional input features to their main network. In our study, we use speaker or phonetic level averages to summarize speaker and phonetic content but we use them as training targets for our auxiliary network instead of using them as input.

2. Speaker Adaptation using Auxiliary Network

In this section, we introduce our network structure that consists of a main and an auxiliary network as shown in Figure 1.

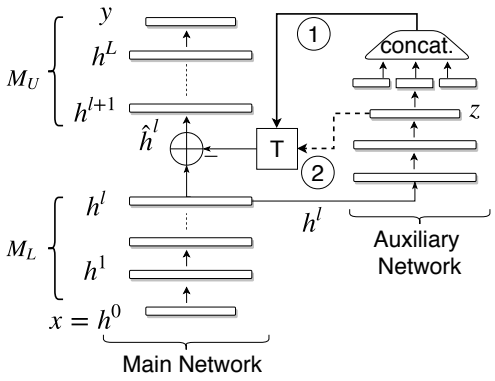

Figure 1: *Structure of the speaker adapted network*

2.1. Main Network

Let the input speech features of the network be denoted by $X = \{x_1, x_2, \ldots, x_N\}$ and the corresponding senone labels as $Y = \{y_1, y_2, \ldots, y_N\}$. First we learn a main senone classifier network with inputs X and outputs Y. Then we choose an adaptation layer l and divide the network into lower (M_L) and upper (M_U) parts. Thus, the main network outputs are written as $M_U(M_L(x_n))$. The hidden layer activations h^l at layer l are then used as input to the auxiliary side network which tries to learn the speaker dependent component of h^l.

2.2. Auxiliary Network

The aim of the auxiliary network is to extract speaker dependent information of the hidden layer activations h^l. The auxiliary speaker network has three linear output layers: one output block tries to reconstruct speaker-level average activations, the second set of output nodes tries to reconstruct the average activation at both speaker and phone level, and the third set of nodes tries to reconstruct the average activation at both speaker and senone level. These three output layers share the auxiliary network parameters at the lower layers and differ only in the final layer. By parameter sharing and joint training of these outputs, we extract speaker dependent information in the network.

Speaker dependent information for the n-th frame, z_n is computed either by concatenating the outputs of the network or by taking the last hidden layer activation of the auxiliary network. Once z_n is computed, a linear transformation T is applied to the speaker dependent features z_n. These transformed features are then subtracted from the hidden layer activations of the main network as in Eq. (1). Thus, we aim at obtaining speaker independent component of h^l such that we achieve better senone classification accuracy in this speaker invariant input space.

$$\widehat{h}_n^l = h_n^l - Tz_n, \qquad n \in \{1, 2, \ldots, N\} \tag{1}$$

If we augment our main network with this auxiliary network, the outputs of the main network are written as $M_U(\widehat{h}_n^l)$.

2.3. Training Procedure

Training procedure starts with learning the parameters of M_L and M_U of the main network, then we compute the activations $h_n^l, \forall n$. To obtain the training targets for the auxiliary network, forced alignment of the data is used to get phone and senone level alignments of the frames. Then for each speaker, (speaker, phone) pair and (speaker, senone) pair, we compute the average of the activations h^l of corresponding frames based on the alignments.

Our auxiliary network training objective is based on multitasking where we want to minimize the mean-squared reconstruction errors of average activations in the auxiliary network while minimizing the cross-entropy of the senone classifier.

3. Experiments

We performed our experiments on the TIMIT dataset. Forced alignments are obtained by a DNN-HMM acoustic model trained using Kaldi [10] TIMIT s5 recipe.

Main network and auxiliary speaker networks are trained using Tensorflow [11]. The main network is a fully-connected network with three layers where each layer has 300 nodes with rectified linear units (relu) activation. Input features of the network are mel-frequency cepstral coefficient features transformed by linear discriminant analysis and maximum likelihood linear transformation and spliced over a neighborhood of nine frames. This results in 360-dimensional input features. The number of senones is 1947.

The auxiliary network consists of three hidden layers with 500, 1000 and 500 nodes with relu activation. Adaptation layers $l \in \{1, 2, 3\}$ were tested; Tables 1 and 2 show the best results, achieved when $l = 1$.

Table 1 shows the senone classification accuracy of the main network and the speaker adapted network when $l = 1$, z is the concatenation of the auxiliary network outputs, shown by path 1 in Figure 1, or when it is the last hidden layer activation of the auxiliary network, shown by path 2 in Figure 1. After subtracting the speaker dependent component of h^l and then feeding them into the upper part of the network, M_U, we obtain a more speaker independent representation of the data and get higher classification accuracy for both training, development and test data. When z is hidden features of the auxiliary network, we achieve 2.8% relative increase in the classification accuracy on the test data as compared to the network without any adaptation.

Table 2 shows the phone error rate (PER) of the systems discussed above. Using the auxiliary network lowers PER, and achieves the best results when the speaker dependent representation z is taken from the hidden layer activations of the auxiliary network resulting in about 1% absolute improvement.

Table 1: *Senone classification accuracy*

	Train	Dev	Test
Main network	69.45	55.65	54.69
Adapted network - concatenated	70.30	56.18	55.28
Adapted network - hidden	73.08	57.05	56.26

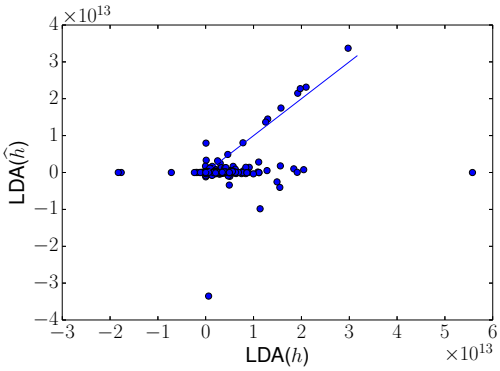

Figure 2: *LDA scores for $\widehat{h}$ versus h*

Table 2: *PER of the TIMIT dataset*

	Train	Test
Main network	19.25	30.31
Adapted network - concatenated	19.10	29.91
Adapted network - hidden	18.81	29.33

When l is chosen to be 2 or 3 and the best performing structure, that is taking the hidden layer of the auxiliary network as z, is used, we got 29.72% and 30.31% PER on the test data.

To show that we achieve speaker invariance after applying Eq. (1), we plotted LDA scores for h and $\widehat{h}$ in Figure 2. We define LDA score to be the trace of inverse within class covariance times the between covariance matrix. When the auxiliary network learns to reconstruct the expectation $E[h|s_n, t_n]$, $E[\widehat{h}|s_n, t_n]$ for speaker-senone pair (s_n, t_n) becomes 0, and hence the between class covariance becomes 0 which results in an LDA score of zero. As shown in Figure 2, for most of the senones, LDA score for $\widehat{h}$ is zero while they are nonzero for h.

4. Conclusions

In this paper, we introduced an auxiliary adaptation network that learns to extract speaker-dependent component of the hidden layer activations of a senone classifier main network. Training targets of the auxiliary network are chosen to be speaker, phone and senone level average activations and the network is trained using a multitask learning objective where we minimize a linear combination of the classification loss in the main network and the reconstruction errors in the auxiliary network. Speaker dependent representation is computed either from concatenation of the output layers or from the last hidden layer activations of the auxiliary network. Experimental results on the TIMIT dataset showed that the PER of the system where we use the hidden layer representation is about 1% absolutely lower than the unadapted system showing that the deep features are more representative of the speaker information as compared to using the reconstructed average speaker dependent activations.

5. References

[1] S. P. Rath, D. Povey, K. Veselỳ, and J. Cernockỳ, "Improved feature processing for deep neural networks." in *Interspeech*, 2013, pp. 109–113.

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

[3] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 24–29.

[4] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 171–176.

[5] Z. Meng, J. Li, Z. Chen, Y. Zhao, V. Mazalov, Y. Gong *et al.*, "Speaker-invariant training via adversarial learning," *arXiv preprint arXiv:1804.00732*, 2018.

[6] J. Neto, L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson, "Speaker-adaptation for hybrid hmm-ann continuous speech recognition system," in *Fourth European Conference on Speech Communication and Technology*, 1995.

[7] Y. Miao, H. Zhang, and F. 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.

[8] K. Veselỳ, S. Watanabe, K. Žmolíková, M. Karafiát, L. Burget, and J. H. Černockỳ, "Sequence summarizing neural network for speaker adaptation," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5315–5319.

[9] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid nn/hmm model for speech recognition based on discriminative learning of speaker code," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7942–7946.

[10] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[11] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.