FAST ADAPTATION ON DEEP MIXTURE GENERATIVE NETWORK BASED ACOUSTIC MODELING

Wen Ding, Tian Tan and Yanmin Qian[†]

Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
SpeechLab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
{wen.ding@sjtu.edu.cn, tantian@sjtu.edu.cn, yanminqian@tencent.com}

ABSTRACT

Deep neural network (DNN) has achieved the state-of-the-art performance in automatic speech recognition (ASR). However, the meaning of parameters and neurons are hard to be interpreted in DNNs, which makes the regularizations and adaptation of DNNs difficult. In this work, we aim to do effective and efficient adaptation on a more interpretable model, deep mixture generative network (DMGN). Adapted means are first proposed to perform adaptation for DMGN. The speaker-dependent means are estimated in an unsupervised adaptation mode. Moreover, discriminative linear regression (DLR) is proposed to estimate more robust speaker-dependent means when lack of adaptation data. We evaluate our proposed methods on 50-hour subset of Switchboard. Experiments reveal that all proposed methods are better than speaker independent baseline, and a slight performance improvement is obtained compared with LHUC. In addition, we project the Gaussian mean of one senone and all inputs aligned to this senone to a 2D graph. The illustration shows that after applying DLR, the mean is indeed transferred from an average point to the speaker specific center, which demonstrates the better explanation of DMGN again.

Index Terms**—** Deep mixture generative network, Speaker Adaptation, Discriminative linear regression, Acoustic Modeling

1. INTRODUCTION

In recent years, the performances of the state-of-the-art speech recognition systems have been significantly improved due to the great progress in deep learning [1, 2, 3, 4, 5]. Although DNNs have performed well in several domains, interpreting the parameters of DNN is still difficult, which makes the adaptation of DNN to acoustic conditions hard.

Traditional adaptation methods for DNN mainly focus on introducing additional features or parameters to model the acoustic conditions. Speaker-dependent (SD) features such as i-vector [6, 7, 8] and speaker code [9] provide speaker information to the networks during training and testing. Some other techniques concentrate on feature-normalization at some DNN layers. For instance, linear input network (LIN) [10, 11, 12] and linear output network (LON) [10, 13] apply a linear transformation to input features and the output layer. The learning hidden unit contributions (LHUC) [14, 15] and the parametric activation [16] scale the activation of hidden layers to transform the features into SD space. In CAT-DNN [17, 18, 19, 20], a speaker-specific weight matrix is estimated by combining the weight matrix base with SD interpolation weights. However, due to lack of understanding the meaning of neurones and parameters in DNN, previous methods would not be the most efficient way to do adaptation.

Recently, several works have been proposed to better understand DNN. In [21, 22], stimulated learning is proposed to force the neurones of different regions belong to different phonemes, making the neurones interpretable. Furthermore, structured neural networks have also been investigated. The DNN topology is explicitly modified to make some parameters in NN to model specific functions such as the deep mixture generative network (DMGN) [23, 24, 25, 26]. In this structure, the likelihood is estimated by using a GMM at the output layer of DNN. Although the behavior of the activation of DNNs is hard to understand, the parameters of GMM have clear meanings and representations, which should be helpful for doing adaptation.

This work aims to do fast and efficient adaptation on the deep mixture generative network. Assume that only one Gaussian is used for a senone, the mean of the Gaussian represents the clustering center of all sample aligned to this senone. However, this mean is estimated over all speakers, samples from different speaker should form different clustering centers. Therefore, adapted means are first proposed to do adaptation for DMGN. The SD means are estimated in an unsupervised adaptation mode. Moreover, discriminative linear regression (DLR) is proposed to estimate more robust

[†] Yanmin Qian is the corresponding author and now he is with Tencent AI Lab, Tencent, Bellevue, WA, USA.

This work was partly supported by the China NSFC projects (No. U1736202 and No. 61603252), and the Shanghai Sailing Program No. 16YF1405300. Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

978-1-5386-4658-8/18/$31.00 ©2018 IEEE

SD means when lack of adaptation data. To better interpret these parameters, visualization of the mean and input features in 2-D pictures are also provided, which reveals that after applying DLR, the mean is indeed transferred from an average point to the speaker-specific center. The proposed adaptation techniques are evaluated on 50-hour subset of Switchboard speech recognition task (SWBD). The experiments show that proposed adaptation methods are better than speaker independent baseline, and a slight performance improvement is obtained compared to LHUC.

The rest of the paper is organized as follows. Section 2 briefly introduces the basic concept and training of deep mixture generative network. The adaptation techniques for deep mixture generative network are included in section 3. Experiment details and results on SWBD are included in Section 4. And finally we give our conclusion in section 5.

2. DEEP MIXTURE GENERATIVE NETWORK FOR AUTOMATIC SPEECH RECOGNITION

In order to better model the senone, Gaussian mixture model is introduced into a DNN at the output layer, which is referred to as deep mixture generative network (DMGN) in [26]. Rather than using softmax layer to predict the posterior probability $p(y|\mathbf{x})$, the likelihood $p(\mathbf{x}|y)$ is estimated at the output layer. Each senone y is modeled by a GMM. The formulation is defined as following:

$$p(\mathbf{x}|y) = \sum_{i=1}^{g} w_{y,i}\mathcal{N}(\mathbf{x}; \mu_{y,i}, \mathbf{\Sigma}_{y,i}), \tag{1}$$

where $\mathbf{x}$ is the input features, y is the senone, $\mu_{y,i}$ and $\mathbf{\Sigma}_{y,i}$ are the mean vector and covariance matrix of the i-th Gaussian of the senone y and $w_{y,i}$ is the mixing weight of each Gaussian.

The topology of the deep mixture generative network is illustrated in Figure 1. Acoustic features first pass through several non-linear transformations. Then a linear bottle-neck layer is used to reduce the dimensionality of input features and remove the correlation between features to make the diagonal covariance matrix assumption hold. So all covariance matrices used in this work are diagonal matrices. At last, the low-dimension uncorrelated vector is input to the GMM layer to get the log-likelihood $\log p(\mathbf{x}|y)$.

Cross-entropy (CE) $\mathcal{L}_{ce}$ between the ground truth label and the senone posterior is optimized to training the network. The following equations are given to calculate the posterior:

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} \tag{2}$$

$$= \frac{\exp(\log p(\mathbf{x}|y) + \log p(y))}{\sum_y \exp(\log p(\mathbf{x}|y) + \log p(y))} \tag{3}$$

$$= \mathrm{softmax}(\log p(\mathbf{x}|y) + \log p(y)) \tag{4}$$

where $p(y) = T_y/T$ is the prior probability which is estimated from training set. The key partial derivatives are the gradients of the likelihood with respect to the mean, variance and mixing weights.

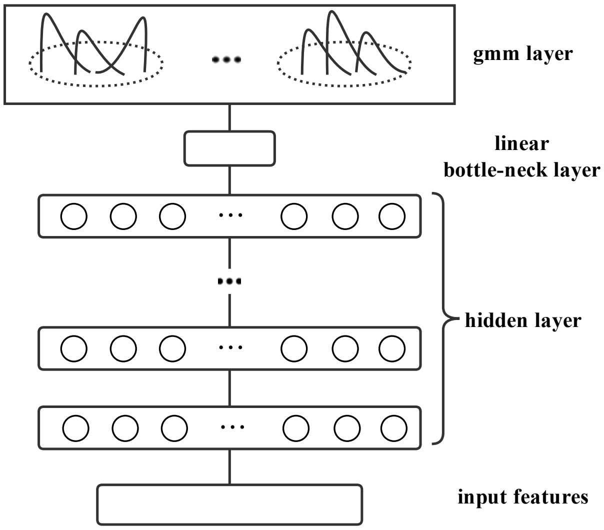

Fig. 1. Deep mixture generative network

$$\frac{\partial \mathcal{L}_{\mathrm{ce}}}{\partial \mu_{y,i,j}} = \frac{\partial \mathcal{L}_{\mathrm{ce}}}{\partial \log p(\mathbf{x}|y)} \{\pi_i(x,y)\frac{x_j - \mu_{y,i,j}}{\sigma_{y,i,j}^2}\} \tag{5}$$

$$\frac{\partial \mathcal{L}_{\mathrm{ce}}}{\partial \sigma_{y,i,j}} = \frac{\partial \mathcal{L}_{\mathrm{ce}}}{\partial \log p(\mathbf{x}|y)} \pi_i(x,y)\{(\frac{x_j - \mu_{y,i,j}}{\sigma_{y,i,j}^2})^2 - 1\} \tag{6}$$

$$\frac{\partial \mathcal{L}_{\mathrm{ce}}}{\partial w_{y,i}} = \frac{\partial \mathcal{L}_{\mathrm{ce}}}{\partial \log p(\mathbf{x}|y)} \{\pi_i(x,y) - w_{y,i}\} \tag{7}$$

where

$$\pi_i(x,y) = \frac{w_{y,i}\mathcal{N}(x; \mu_{y,i}, \Sigma_{y,i})}{\sum_{l=1}^{g} w_{y,l}\mathcal{N}(x; \mu_{y,l}, \Sigma_{y,l})} \tag{8}$$

and the $\frac{\partial \mathcal{L}_{ce}}{\partial \log p(\mathbf{x}|y)}$ is the partial derivative with respect to $\log p(\mathbf{x}|y)$, which is the error propagating back to the GMM. The index i is the number of GMM components and index j is related to the dimension, so $i = 1, ..., g$ and $j = 1, ..., d$. With the update equations above, the deep mixture generative network can be trained accordingly.

3. FAST ADAPTATION ON DEEP MIXTURE GENERATIVE NETWORK AM

In this section, several adaptation techniques are investigated for deep mixture generative network. We begin with the learning hidden unit contributions (LHUC) which is well known for DNN. Then we propose adapted means and discriminative linear regression for adapting DMGN. Since the mean of each Gaussian is the cluster center of a given state, it is expectable that adapting the mean should be more efficient.

3.1. LHUC

LHUC [14] is a typical approach to adapt DNN, in which a speaker-dependent (SD) transformation is applied after the activations of the hidden layer for each speaker s by

$$\mathbf{h}_s^l = a(\mathbf{r}_s^l) \cdot \phi(\mathbf{W}^l \mathbf{h}_s^{l-1} + \mathbf{b}^l) \tag{9}$$

where $\mathbf{h}_s^l$ is the adapted hidden output of layer l, $\mathbf{r}_s^l$ is a speaker specific vector for the l-th hidden layer and $\cdot$ is an element-wise multiplication. This method can be applied to DMGN directly since the first several layers of DMGN are normal operation like DNNs.

3.2. Adapted Means

Although LHUC is a good way to do adaptation for DNNs, it is not the most appropriate for the DMGN because LHUC do not consider the meanings of neurones. There is no prior knowledge of the activation of each layers. Based on this consideration, adapted means are proposed. An unsupervised adaptation mode is used, hypotheses are first generated using the speaker-independent deep mixture generative network (SI-DMGN) system to get state level alignments. Then the mean of each Gaussian will be adapted to a speaker-specific mean. The adaptation criterion is to minimize the cross entropy between the state posterior and the label generated by hypothesis. After that, different speakers will have different means. Thus, the mean of each Gaussian will move much closer to the true clustering center of a given speaker rather than an average of all speakers.

3.3. Discriminative Linear Regression

However, due to the lack of adaptation data, it is impossible to estimate the correct adapted means for all senones. Discriminative linear regression (DLR) is proposed to do more robust adaption for DMGN.

DLR is intended for transformations of means of Gaussian mixtures learned from limited adaptation data. At the output layer of DMGN, the means of each Gaussian is transformed directly by

$$\mu_{y,i}^s = \mathbf{W}_s \mu_{y,i} \qquad \forall y, i \tag{10}$$

where $\mathbf{W}_s$ is the transformation matrix of speaker s to transform the mean $\mu_{y,i}$ for all different senones to a speaker specific mean $\mu_{y,i}^s$. Different speakers use different $\mathbf{W}_s$. After transformation, the GMM in DMGN outputs the likelihood using the new transformed parameters:

$$\begin{aligned}\log p(\mathbf{x}|y,s) &= \sum_{i=1}^{g} w_{y,i} \mathcal{N}(\mathbf{x}; \mu_{y,i}^s, \mathbf{\Sigma}_{y,i}) && (11)\\ &= \sum_{i=1}^{g} w_{y,i} \mathcal{N}(\mathbf{x}; \mathbf{W}_s \mu_{y,i}, \mathbf{\Sigma}_{y,i}) && (12)\end{aligned}$$

The transformation matrix is also optimized by unsupervised adaptation mode. Only transformation matrix $\mathbf{W}$ is updated during adaptation and all other parameters are frozen. The training criterion is CE and the update can use the common parameter training scheme such as stochastic gradient descent (SGD). Since a bottle-neck layer is used before the GMM output layer, e.g. 50-dim bottle-neck, the size of the transformation matrix is quite small to make the adaptation efficient and effective.

4. EXPERIMENTS AND RESULTS

4.1. Dataset description

A 50 hour subset of the Switchboard dataset is used for evaluation in this paper. There are 810 speakers including in the training set. We used two test sets including: the Fisher and the Switchboard part of the Rich Transcription 2003 evaluation, which is referred to as *fsh* and *swbd* in rest experiments. Test sets include 144 speakers and 8422 utterances.

4.2. Experimental set-up

The deep mixture generative networks and all proposed adaptation methods were implemented using CNTK [27] and a GMM-HMM model containing 2723 tied tri-phone states was first trained to generate the alignments for DNN training. Kaldi [28] was used to train GMM-HMM and decode.

36-dimensional log mel-frequency filter bank (FBANK) along with their first and second order derivatives were extracted as features and CMN for each speaker is applied. 11 consecutive frames (one frame with its left and right 5 frames) were used as input and the label of each frame is the forced alignment generated by GMM-HMM system. Sigmoid was chosen as the activation function and CE was used as the training criterion. SGD was used to train models, with an initial learning rate of 1.0. The learning rate would be reduced by half if the CE of cross validate set (CV) does not decrease after an epoch and the batch size is 256. A trigram language model trained on the Switchboard transcripts was used for decoding. The adaptation schemes are evaluated in an unsupervised fashion: a speaker-independent system is used to generated the hypotheses and the state level alignment, then these alignments would be used to estimate the SD parameters such as mean or transformation matrix for each speaker.

4.3. Baseline

The baseline DNN contained 5 hidden layers with 2048 nodes at each layer and a linear bottle-neck layer with 50 nodes before the softmax layer. The performance of the baseline DNN is shown at the first line in Table 1. Then the baseline DNN with the bottle-neck layer was used to initialize the DMGN system. The output softmax layer was removed and replaced by a GMM layer. The $\mathbf{\Sigma}$ was frozen as an identity matrix in our experiment for simplicity. So the parameters in GMM layer are the means $\mu_{y,i}$ and the mixing weights $w_{y,i}$ Using a DNN as an initialization is of great importance because the network could not train well when initialized randomly from scratch in our experience. After the initialization, parameters in the GMM layer were updated for one epoch. The performance of DMGN with different number of Gaussian components including 1, 2 and 4 are compared in the Table 1. The single Gaussian performs the best compared to that with 2 and 4 Gaussian mixture components. With a single Gaussian component, the DMGN performs slightly better than common DNN with bottle-neck layer, while DMGN does not perform better with the number of Gaussian mixture components increases. It can result from the strong assumption we make that the covariance is always an identity. In all rest experiments, only single Gaussian is used.

Table 1. WER (%) comparison of the DNN baseline and DMGN with different numbers of Gaussian.

Models	*swbd*	*fsh*
DNN	37.9	27.4
1-gmm DMGN	**37.6**	**27.4**
2-gmm DMGN	37.9	27.5
4-gmm DMGN	37.9	27.6

4.4. Adaptation performance evaluation

The performance of all proposed adaptation methods for DMGN are shown in Table 2 including LHUC, adapted means and DLR. LHUC was applied to the first layer of the DMGN since in [14] the first layer gained almost performance improvement. The result reveals that LHUC doesn't work well for DMGN. Results of adapted means are illustrated in the third line of Table 2. This method yielded better performance on both *swbd* and *fsh* compared to the SI-DMGN systems. Significant performance improvement was obtained by using DLR compare to the SI system, which demonstrates that using transformation can indeed capture the characteristics of a specific speaker to get a more robust adapted mean. And since only the means of Gaussian are adapted, the adaptation process is fast. The numbers of parameters are 2048 for LHUC and 50×50 for DLR.

Table 2. WER (%) Comparison of different adaptation methods on the DMGN and the relative WER gains.

type	*swbd*	*fsh*
SI-DMGN	37.6	27.4
LHUC	37.6 (0%)	27.0 (1.46%)
Adapted means	37.4 (0.53%)	27.3 (0.36%)
DLR	**37.1 (1.33%)**	**27.0 (1.46%)**

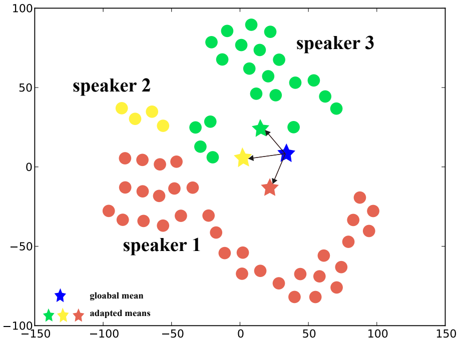

Fig. 2. Global and adapted means visualization of the GMM-layer in DMGN. This is the example of state 467 (phone n). The inputs of GMM in DMGN come from different 3 speakers and are drawn with green, yellow and red respectively. The blue star is the original global mean and the green, yellow and red star are the adapted means of each speaker.

Figure 2 illustrates how DLR works for speaker adaptation in DGMN. All hidden outputs of the BN layer that align to a random chosen senone (from phone n) among three speakers were projected to a 2D plane using t-SNE [29]. As shown in the Figure 2, dots with different color represent outputs from different speakers. It is observed that the hidden outputs from different speaker do have their own clustering center. The mean of the Gaussian in our SI DMGN (which is referred to as *global mean*) and the means after DLR (which are referred to as *adapted means*) are also drawn in the same graph. And the blue star is the original global mean from GMM in our baseline DMGN. The other three stars represent the adapted means. It is observed that the global mean is at the center of all samples, however for a specific speaker, the distance between the global mean and the real center is quite far. After DLR training, the mean is indeed transferred from the average point to the speaker specific center, which demonstrates the better explanation of DMGN.

5. CONCLUSION

This paper proposes adaptation methods on a structured and interpretable network, the deep mixture generative network. Adapted means are first proposed to perform adaptation for DMGN, which is more effective compared with the typical adaptation method such as LHUC for DNN. Moreover, discriminative linear regression (DLR) is proposed to estimate more robust speaker-dependent means when lack of adaptation data. Experiments reveal that all proposed methods are better than speaker independent baseline, and a significant performance improvement is obtained after using DLR for speaker adaptation. Visualizations for the global mean and adapted means show that the proposed methods actually help the mean of a Gaussian move from an average point to the speaker-specific center.

6. REFERENCES

[1] George E Dahl, Dong Yu, Li Deng, and Alex Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 20, pp. 30–42, 2012.

[2] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdelrahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, 2012.

[3] Yanmin Qian, Mengxiao Bi, Tian Tan, and Kai Yu, "Very deep convolutional neural networks for noise robust speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 24, no. 12, pp. 2263–2276, 2016.

[4] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *INTERSPEECH*, 2015, pp. 3214–3218.

[5] Dong Yu, Wayne Xiong, Jasha Droppo, Andreas Stolcke, Guoli Ye, Jinyu Li, and Geoffrey Zweig, "Deep convolutional neural networks with layer-wise context expansion and attention.," in *INTERSPEECH*, 2016, pp. 17–21.

[6] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny, "Speaker adaptation of neural network acoustic models using i-vectors.," in *ASRU*, 2013, pp. 55–59.

[7] Romain Serizel and Diego Giuliani, "Vocal tract length normalisation approaches to dnn-based children's and adults' speech recognition," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 135–140.

[8] Tian Tan, Yanmin Qian, Dong Yu, Souvik Kundu, Liang Lu, Khe Chai Sim, Xiong Xiao, and Yu Zhang, "Speaker-aware training of lstm-rnns for acoustic modelling," in *ICASSP*, 2016, pp. 5280–5284.

[9] Ossama Abdel-Hamid and Hui Jiang, "Fast speaker adaptation of hybrid nn/hmm model for speech recognition based on discriminative learning of speaker code," in *ICASSP*, 2013, pp. 7942–7946.

[10] Bo Li and Khe Chai Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid nn/hmm systems," in *INTERSPEECH*, 2010, pp. 526–529.

[11] Joao Neto, Luís Almeida, Mike Hochberg, Ciro Martins, Luis Nunes, Steve Renals, and Tony Robinson, "Speaker-adaptation for hybrid hmm-ann continuous speech recognition system," 1995.

[12] Frank Seide, Gang Li, Xie Chen, and Dong Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *ASRU*. IEEE, 2011, pp. 24–29.

[13] 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)*. IEEE, 2012, pp. 366–369.

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

[15] Pawel Swietojanski, Jinyu Li, and Steve Renals, "Learning hidden unit contributions for unsupervised acoustic model adaptation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 8, pp. 1450–1463, 2016.

[16] C Zhang and Philip C Woodland, "Dnn speaker adaptation using parameterised sigmoid and relu hidden activation functions," in *ICASSP*, 2016, pp. 5300–5304.

[17] Tian Tan, Yanmin Qian, Maofan Yin, Yimeng Zhuang, and Kai Yu, "Cluster adaptive training for deep neural network," in *ICASSP*, 2015, pp. 4325–4329.

[18] Tian Tan, Yanmin Qian, and Kai Yu, "Cluster adaptive training for deep neural network based acoustic model," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 3, pp. 459–468, 2016.

[19] Chunyang Wu and Mark JF Gales, "Multi-basis adaptive neural network for rapid adaptation in speech recognition," in *ICASSP*, 2015, pp. 4315–4319.

[20] Marc Delcroix, Keisuke Kinoshita, Chengzhu Yu, Atsunori Ogawa, Takuya Yoshioka, and Tomohiro Nakatani, "Context adaptive deep neural networks for fast acoustic model adaptation in noisy conditions," in *ICASSP*, 2016, pp. 5270–5274.

[21] Chunyang Wu, Penny Karanasou, Mark JF Gales, and Khe Chai Sim, "Stimulated deep neural network for speech recognition.," in *INTERSPEECH*, 2016, pp. 400–404.

[22] Chunyang Wu and Mark JF Gales, "Deep activation mixture model for speech recognition," *Proc. Interspeech 2017*, pp. 1611–1615, 2017.

[23] Christopher M Bishop, "Mixture density networks," 1994.

[24] Korin Richmond, "A trajectory mixture density network for the acoustic-articulatory inversion mapping," in *INTERSPEECH*, 2006, pp. 577–580.

[25] Heiga Zen and Andrew Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *ICASSP*, 2014, pp. 3844–3848.

[26] Ehsan Variani, Erik McDermott, and Georg Heigold, "A gaussian mixture model layer jointly optimized with discriminative features within a deep neural network architecture," in *ICASSP*, 2015, pp. 4270–4274.

[27] Dong Yu, Adam Eversole, Mike Seltzer, Kaisheng Yao, Zhiheng Huang, Brian Guenter, Oleksii Kuchaiev, Yu Zhang, Frank Seide, Huaming Wang, et al., "An introduction to computational networks and the computational network toolkit," *Microsoft Technical Report MSR-TR-2014–112*, 2014.

[28] 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 *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[29] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.