INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Speaker normalization using Joint Variational Autoencoder

Shashi Kumar[1], *Shakti P. Rath*[2], *and Abhishek Pandey*[1]

[1]Samsung R&D Institute India - Bangalore, India
[2]Reverie Language Technologies, India

sk.kumar@samsung.com, shakti.rath@reverieinc.com,abhi3.pandey@samsung.com

Abstract

Speaker adaptation is known to provide significant improvement in speech recognition accuracy. However, in practical scenario, only a few seconds of audio is available due to which it may be infeasible to apply speaker adaptation methods such as i-vector and fMLLR robustly. Also, decoding with fMLLR transformation happens in two-passes which is impractical for real-time applications. In recent past, mapping speech features from speaker independent (SI) space to fMLLR normalized space using denosing autoencoder (DA) has been explored. To the best of our knowledge, such mapping generally does not yield consistent improvement. In this paper, we show that our proposed joint VAE based mapping achieves a large improvements over ASR models trained using filterbank SI features. We also show that joint VAE outperforms DA by a large margin. We observe a relative improvement of 17% in word error rate (WER) compared to ASR model trained using filterbank features with i-vectors and 23% without i-vectors.

Index Terms: Variational Autoencoders, fMLLR, i-vector, Joint Variational Autoencoders, Speaker Adaptation

1. Introduction

Deep Neural Networks (DNN) and Long Short Term Memory (LSTM) networks have achieved tremendous success in acoustic modeling of automatic speech recognition (ASR). Nevertheless, the performance of these models is adversely affected by speaker, background noise and other types of acoustic variabilities. The focus of this paper is speaker adaptation (normalization). Impact of the differences in speaker characteristics can be reduced either by feature-space transformations like vocal tract length normalization (VTLN) and feature-space maximum likelihood linear regression (fMLLR) [1, 2] before training acoustic model, or by providing it speaker specific codes like i-vectors [3, 4], bottleneck features [5, 6], speaker codes [7, 8]. Our focus in this paper is on fMLLR based feature space transformation. The issue with fMLLR is that it requires 10 to 30 seconds of enrollment speech (whereas i-vector requires at least 10 seconds) from every test speaker for robust estimation of the parameters. In a real-time scenario, available audio is much less which may lead to un-reliable estimation of these transforms resulting in sub-optimal ASR accuracy. Besides, fMLLR based decoding happens in two pass, which is infeasible for real-time applications. Recently these issues were addressed in [9], where the authors proposed to learn a DNN based mapping from filterbank (speaker independent space) to fMLLR-normalized features in a regression framework, and then use these normalized features predicted by the DNN to train acoustic model. We term this type of mapping as denoising autoencoder (DA). To the best of our knowledge, the DA based mapping does not show consistent improvements [9] for speaker adaptation.

Variational Autoencoder (VAE) [10] is a class of generative model that projects input space to a latent space using an encoder and then reconstructs the original input using a decoder. VAEs have been explored for many tasks like speech transformation from a source domain to target domain [11], showing orthogonality of speech attributes that can help in domain translation [12] and speaker verification [13]. However, mapping filterbank to fMLLR-normalized features can not be done by a conventional VAE because of constraint that input and output must be same mathematically.

Recently, we proposed a novel method for speech enhancement, termed joint VAE [14], which showed promising results in mapping distant to close-talk speech. The constraint with conventional VAE is addressed in joint VAE by learning a joint-distribution of filterbank and fMLLR features for a common latent space. In this paper, we propose to explore joint VAE as an alternative to DA for speaker adaption and show that it yields consistent and considerable improvement in ASR accuracy compared to the DA, an acoustic model trained using filterbank features and speaker aware training using i-vectors. In addition we show that performance of joint VAE is very close to acoustic model trained on fMLLR-normalized features. The proposed method does not require two-pass decoding. In addition separate enrollment data is not requited to estimate the fMLLR transforms in test time.

In Section 2, we review conventional VAE and joint VAE. In Section 3, sepaker adaptation with joint VAE is discussed. Experimental setup and results are presented in Section 4. Finally, conclusions are presented in Section 5.

2. Review of Variational autoencoders and joint VAE

In recent years, VAEs have gained traction for unsupervised learning of unknown and intractable distributions. These models are essentially an encoder-decoder based model where the encoder maps input feature space to latent space and the decoder tries to reconstruct the features given samples from latent space. Standard VAE tries to reconstruct the input space and hence it does not offer domain translation. Recently we proposed joint VAE [14] which enables domain transformation by learning a joint distribution of two domains for a common latent space. Details of standard VAE and joint VAE are explained in the following sections.

2.1. Variational Autoencoder (VAE)

The underlying principle in VAE is to assume that the observed data has been generated by a random process that involves latent variables. Let the sequence of latent variables be denoted by $\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_N$ and the observed data be denoted by $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N$. In order to model the observed process, it is necessary to estimate the the posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$ given samples from input space distribution, where p_θ denotes

http://dx.doi.org/10.21437/Interspeech.2021-467

family of distributions parameterized by θ. Using Baye's rule $p_\theta(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})/p_\theta(\mathbf{x})$. In practice $p_\theta(\mathbf{z}|\boldsymbol{x})$ becomes intractable even for a simpler distribution family. So it is approximated by another parametrized distribution by minimizing Kullback-Leibler (KL) divergence between the two. Let the other distribution be denoted by $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. Now, KL-divergence is minimized between $p_\theta(\mathbf{z}|\mathbf{x})$ and $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. It is straight forward to show that following relation holds

$$\log\, p_\theta(\boldsymbol{x}) = \mathcal{L}_1(\theta, \phi; \boldsymbol{x}) + KL(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \,||\, p_\theta(\boldsymbol{z}|\boldsymbol{x})) \tag{1}$$

$$\geq \mathcal{L}_1(\theta, \phi; \boldsymbol{x}) \tag{2}$$

where $p_\theta(\boldsymbol{x})$ denotes the marginal distribution of the observed data and $\mathcal{L}_1(\theta, \phi; \boldsymbol{x})$ is called the variational lower bound, which is defined as

$$\begin{aligned}\mathcal{L}_1(\theta, \phi; \boldsymbol{x}) &= \int_z q_\phi(\boldsymbol{z}|\boldsymbol{x})\, \log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \qquad (3)\\ &= E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) - KL\left(q_\phi(\boldsymbol{z}|\boldsymbol{x}) || p_\theta(\boldsymbol{z})\right)\end{aligned}$$

Commonly, the prior $p_\theta(\boldsymbol{z})$ is modeled by isotropic Gaussian distribution $p_\theta(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \mathbf{0}, \mathbf{I})$ and the distributions $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ by diagonal Gaussian distributions which are represented by neural networks. Parameters ϕ and θ of the these distributions are jointly estimated by minimizing negative of the variational lower bound (Eq 3). To compute expectation term in Eq 3 of variational lower bound, samples $\hat{\boldsymbol{z}}$ needs to be drawn from the posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. Since sampling is a non-differentiable operation, the standard error back-propagated cannot directly be applied for the training. To handle this limitation, the re-parameterization trick [10] is used to make the sampling operator differentiable.

It is important to note here that it may appear that conventional VAE may be extended for domain conversion. In past, it has been explored for speech enhancement task (denoising VAE (DVAE) [15]) where conventional VAE is applied to learn a mapping from noisy to clean speech domains. However from Eq. 3, it may be noted that in VAE, the input and output random processes must be the same, i.e., $\boldsymbol{x}$. If the input and output are forced to be different, as in the case of DVAE, the results may become unpredictable. Therefore from a theoretical point of view, such domain conversion cannot be justified within the premises of conventional VAE. For this reason, results are not shown for DVAE in this paper.

2.2. Joint Variational Autoencoder (Joint VAE)

In this section, we present a detailed description of joint VAE [14]. For consistency, we denote input features or input domain (speaker independent) by $\boldsymbol{x}$ and output domain (speaker normalized) by $\boldsymbol{y}$. Our motivation is to learn mapping from $\boldsymbol{x}$ to $\boldsymbol{y}$ in a time synchronous fashion. We assume to have access to parallel data for these domains aligned in time at the training phase. In joint VAE, the distribution of the data from input domain and output domains are modeled using a joint probability distribution, and the variational lower bound is re-defined as follows

$$\begin{aligned}\mathcal{L}_2(\theta, \phi; \boldsymbol{x}, \boldsymbol{y}) &= \int_{\boldsymbol{z}} q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}) \log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})}\\ &= E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \log\, p_\theta(\boldsymbol{x}|\boldsymbol{z}) + E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\, \log\, p_\theta(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z})\\ &\quad - KL(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \,||\, p_\theta(\boldsymbol{z})) \qquad (4)\end{aligned}$$

The modified lower bound consists of two conditional distributions $p_\theta(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z})$ and $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ and the posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, each of which is represented using a neural network. It may be noted here that we have made an approximation $q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}) = q_\phi(\boldsymbol{z}|\boldsymbol{x})$ assuming the mapping between domains $\boldsymbol{x}$ and $\boldsymbol{y}$ is deterministic. All the above conditional distributions are modeled by a diagonal Gaussian distribution

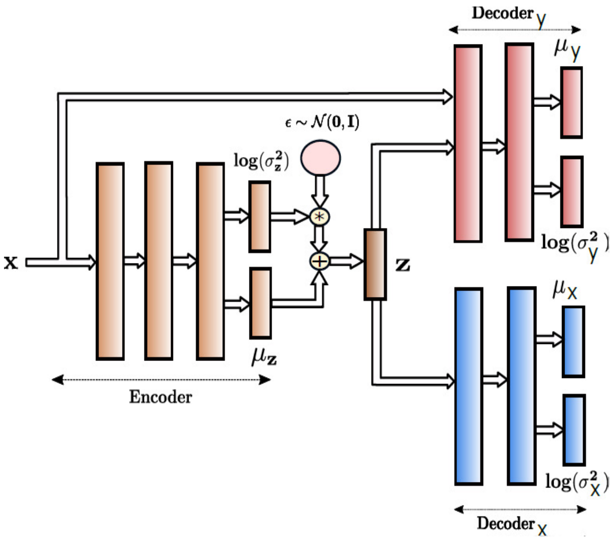

Figure 1: *Joint VAE architecture*

$$\begin{aligned}p_\theta(\boldsymbol{x}|\boldsymbol{z}) &= \mathcal{N}(\boldsymbol{x}; f^1_{\mu_{\boldsymbol{z}}}(\boldsymbol{z}; \theta), \exp(f^1_{\log \sigma^2_{\boldsymbol{z}}}(\boldsymbol{z}; \theta)))\\ p_\theta(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}) &= \mathcal{N}(\boldsymbol{y}; f^2_{\mu_{\boldsymbol{x}, \boldsymbol{z}}}(\boldsymbol{x}, \boldsymbol{z}; \theta), \exp(f^2_{\log \sigma^2_{\boldsymbol{x}, \boldsymbol{z}}}(\boldsymbol{x}, \boldsymbol{z}; \theta)))\\ q_\phi(\boldsymbol{z}|\boldsymbol{x}) &= \mathcal{N}(\boldsymbol{z}; g_{\mu_{\boldsymbol{x}}}(\boldsymbol{x}; \phi), \exp(g_{\log \sigma^2_{\boldsymbol{x}}}(\boldsymbol{x}; \phi))) \qquad (5)\end{aligned}$$

and the prior $p_\theta(\boldsymbol{z})$ is modeled by isotropic Gaussian and the network parameters are jointly optimized by minimizing negative of modified lower bound (Eq. 4). In practice, the actual loss that is used to train networks is defined by

$$\mathcal{L}_3 = \lambda_1\, \text{MSE}_\text{x} + \lambda_2\, \text{MSE}_\text{y} + \lambda_3\, \text{KLD}, \tag{6}$$

where the first term is MSE_x is heteroscedastic MSE [16] between input $\boldsymbol{x}$ and reconstructed $\boldsymbol{x}$ output, the second term MSE_y is heteroscedastic MSE between true $\boldsymbol{y}$ and reconstructed $\boldsymbol{y}$ output. The third term is KL-divergence between $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and prior distribution $p_\theta(\boldsymbol{z})$. In the joint VAE loss, the role of the the KLD term is to smoothen the decision boundaries among different classes. It forces the distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ to be as close to isotropic diagonal Gaussian and it induces inherent disentanglement [17], whereas the reconstruction terms encourage deviation from prior distribution in the latent space so as to encode data effectively in different dimensions of the latent variables.

The block diagram of joint VAE is shown in Figure 1 as discussed in original joint VAE paper [14]. In contrast to conventional VAE, the joint VAE consists of one encoder and two decoders shown as Decoder_x and Decoder_y. The last LSTM layer of the encoder is followed by two parallel fully connected layers with linear activation, predicting mean and log-variance. Similarly, the lower decoder network consists of two parallel fully connected layers with linear activation, predicting mean and log-variance. It takes $\boldsymbol{z}$ as input and predicts mean and log-variance of $\boldsymbol{x}$. The upper decoder takes $\boldsymbol{z}$ and $\boldsymbol{x}$ as input and predicts the mean and log-variance of $\boldsymbol{y}$.

Table 1: *WER (%) of acoustic models trained on different types of features*

Features	**Speaker-wise Normalization**				**Utterance-wise Normalization**			
	dev_clean	dev_other	test_clean	test_other	dev_clean	dev_other	test_clean	test_other
fMLLR	**6.35**	**21.26**	**6.97**	**22.97**	**8.49**	**27.54**	9.17	**29.32**
Filterbank	8.75	29.7	9.24	31.1	9.12	30.78	9.66	32.31
Filterbank + i-vector	8.09	27.65	8.58	30.04	8.61	29.63	**9.04**	31.86
Denoising Autoencoder	8.34	28.5	9.17	30.52	8.75	29.38	9.59	31.45
joint VAE	**6.70**	**23.55**	**7.32**	**25.26**	**7.03**	**24.14**	**7.65**	**25.88**

3. Speaker Adaptation using joint VAE

One of the popular methods for speaker adaptation (normalization) is fMLLR [2]. To facilitate fMLLR adaptation, an affine transform is estimated for every test speaker using a GMM-HMM model. It is a common practice to follow two pass decoding to estimate the fMLLR transforms. In the first pass, decoding is done using speaker independent (SI) features and SI model. The alignment generated using the first-pass hypothesis is used to collect statistics and estimate the fMLLR transform. Then decoding is applied again using the fMLLR transformed (speaker normalized) features using SI model. When fMLLR adaptation is also done in the training time, it is called speaker adaptive training (SAT) [2]. SAT has been shown to provide significant improvement even for DNN-HMM framework [18, 1, 19], where the DNN-HMM are trained and tested using fMLLR normalized features. One of the issues with SAT is that the fMLLR transforms are estimated in a two-pass decoding scenario, which is infeasible in most real-time applications. Moreover, usually it is necessary to collect about 10-30 seconds of enrollment data from every test speaker separately, which is cumbersome.

In this paper we propose to apply speaker adaptation using joint VAE, which is trained to facilitate domain conversion from speaker independent (SI) space to fMLLR (speaker normalized) space. Using the convention of joint VAE, $\boldsymbol{x}$ and $\boldsymbol{y}$ denote the SI and speaker normalized (SN) features respectively. The main advantage of the proposed scheme is that joint VAE learns the mapping from SI to SN domains in the training time and generalizes in the test time, i.e., estimation of fMLLR transforms is not required in the test time – the SI features are passed through the joint VAE network to generate SN features. As a consequence the proposed method is suitable for real-time speech recognition tasks, where two-pass decoding is infeasible. Moreover, it is not necessary to collect separate enrollment data from the test users.

4. Experiments and Results

4.1. Experimental Setup

The experiments are conducted on the Librispeech corpus [20]. Training is done on the 100 hours subset and the results are reported on all 4 standard Librispeech test cases, namely, dev_clean, dev_other, test_clean and test_other. The GMM-HMM is trained using standard recipe for Librispeech in the Kaldi toolkit [21] and the joint VAE networks and the LSTM-HMM based acoustic models are trained using pytorch. SAT training was applied to GMM-HMM model using global fMLLR estimated on top of LDA+MLLT features. We used this model to generate alignment for LSTM-HMM training. Decoding is done using "tgsmall" language model (LM), which is a 3-gram LM pruned with $3e^{-7}$ threshold, followed by rescoring using "fglarge" LM, which is a 4-gram LM. The LSTM-HMM consists of 3 LSTM layers with 512 cells each and cross entropy loss is used to train this model. Left context of 20 frames is used for better cell state initialization before processing a sequence of 30 frames. Adam optimizer is used with learning rate varying from 0.001 to 0.0001 following an exponential learning rate scheduler for initial 20 epochs and then further 5 epochs with fixed final learning rate.

4.2. Baseline models

We compare the performance of the proposed joint VAE against four baseline systems. The first system is a SAT LSTM-HMM system trained on 60-dimensional fMLLR-normalized features with ±2 splicing. In the decoding time, the fMLLR matrices are obtained from the GMM-HMM SAT model following two pass decoding. This result form the upper bound model for all our experiments described in this paper. Second system is a speaker independent system trained on 41-dimensional log-mel filterbank features with ±2 splicing. The third model is trained on 41-dimensional log-mel filterbank features with ±2 splicing concatenated with 40-dimensional i-vectors [9] for speaker aware training. It is well known that i-vectors encode rich speaker information in a low dimensional space. When concatenated with acoustic features, it gives a significant improvement in ASR accuracy.

We also compare the performance against conventional denoising autoencoder (DA) framework. Following [9], the baseline DA is trained to map filterbank features to fMLLR-normalized features. This DA consists of 5 layer LSTM followed by a fully connected layer. It may be noted that our joint VAE architecture consists of 3 layer encoder and 2 layer decoder, thus has similar modeling power as the DA model. The mean square error (MSE) between true and predicted fMLLR-normalized features is minimized to train the DA. We use stochastic gradient descent (SGD) [22] optimizer with learning rate 0.001 and momentum of 0.9. Once this DA is trained, we train the acoustic model again using speaker normalized features obtained from this DA as input. At test time, filterbank features are first passed through the DA and then given to this LSTM-HMM acoustic model for decoding.

4.3. Proposed joint VAE

The joint VAE architecture is shown in Figure 1. The encoder network comprises of 3 LSTM [23] layers, each consisting of 512 hidden cells. The lower decoder subnetwork, $\text{Decoder}_{\text{x}}$, comprises of 2 LSTM layers consisting 512 hidden cells each. The upper decoder subnetwork, $\text{Decoder}_{\text{y}}$, has same layer structure as Decoder_x. The model is trained by minimizing the loss function defined in Eq 6 using SGD optimizer with momentum of 0.9. The learning rate is decreased from 0.001 to 0.0001 in a step-wise fahsion after every 10 epochs until 45 epochs. The model is trained for 5 more epochs with fixed final learning rate. The value of hyperparameters λ_1, λ_2 and λ_3 are taken as 1, 10 and 0.1 respectively, as mentioned in [14]. The acoustic model is trained jointly with the joint VAE model using the mean predicted by Decoder_y, implying that the ASR model is trained on predicted fMLLR-normalized features. At test time, filterbank features are passed through joint VAE and the resulting features are given to the acoustic model for decoding.

4.4. Results

Results are shown in Table 1. The columns under "Speaker-wise Normalization" imply that at test time fMLLR transforms and i-vectors are estimated on a per-speaker basis, whereas the columns under "Utterance-wise Normalization" mean per-utterance estimation. At the training time, the fMLLR and i-vectors are estimated on a per-speaker scenario, whereas test was conducted on both speaker-wise and utterance-wise estimation of fMLLR and i-vectors.

The results in the first row of Table 1 correspond to the ASR system trained and tested using fMLLR-normalized features, termed as fMLLR-ASR. It can be seen as the upper bound for the extent of this paper. It may be observed from second row of the Table that ASR trained with filterbank features (without speaker adaptation) performs significantly worse than fMLLR-ASR. This model is termed as fbank-ASR. From the third row, we note that concatenating i-vectors with filterbank features provides significant improvements over fbank-ASR, still it is considerably inferior to fMLLR-ASR.

Results with DA and joint VAE are also shown in Table 1. In these cases, the filterbank features are mapped to fMLLR-normalized features. It may be noted from the Table that DA provides a slight improvement over filterbanks only features (fbank-ASR). From the results, it may be concluded that the generalization power of DA to unseen speakers [9] is rather poor. The proposed joint VAE, on the other hand, yields a large improvement compared to filterbank features, indicating that the generalization power is far superior to DA. It is also noted that the performance of joint VAE is now close to the conventional SAT system. Specifically, our proposed method, on dev_clean test cases with speaker-wise normalization, shows a relative improvement of 23.42% in WER compared to fbank-ASR, 17.18% compared to filterbank+i-vectors ASR and 19.66% compared to conventional DA baseline. At the same time, the drop from the upper bound is only 5.22% in relative terms. It is clear from the results that joint VAE outperforms DA mapping and all the other ASR baselines by a large margin whereas the drop in WER from the upper bound is small. It may be noted here that decoding using our proposed method is done in a single pass, in contrast to two pass decoding in the case of fMLLR-ASR – still the performances are close.

As evident from Table 1, fMLLR-ASR performs significantly worse in utterance-wise normalization scenario as compared to speaker-wise normalization scenario. At the same time, fMLLR-ASR performs better than the ASR trained on filterbank features in utterance-wise normalization scenario. We would like to emphasize here that utterance-wise normalization resembles closely with real time applications of ASR, in which case our proposed method outperforms every baseline method considered in this paper by a large margin.

5. Conclusions

In this paper, we explore joint VAE for speaker adaptation. We show that joint VAE based mapping from filterbank features to fMLLR-normalized features gives consistent improvements in all the test sets. Experimental results on Librispeech corpus show that the proposed method yields a relative improvement of 17.18% in WER compared to ASR trained using filterbank features concatenated with i-vectors. It is also shown that joint VAE outperforms conventional denoising autoencoder based mapping by a significant margin, obtaining a relative improvement of 19.66% in WER. In all related works explored by us, we believe that for the first time it is being shown that such mapping works with significant improvement in ASR accuracy.

6. References

[1] S. H. K. Parthasarathi, B. Hoffmeister, S. Matsoukas, A. Mandal, N. Strom, and S. Garimella, "fmllr based feature-space speaker adaptation of dnn acoustic models," in *Sixteenth annual conference of the international speech communication association*, 2015.

[2] M. J. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[3] A. Senior and I. Lopez-Moreno, "Improving dnn speaker independence with i-vector inputs," in *2014 ICASSP*. IEEE, 2014, pp. 225–229.

[4] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 55–59.

[5] T. Tan, Y. Qian, D. Yu, S. Kundu, L. Lu, K. C. Sim, X. Xiao, and Y. Zhang, "Speaker-aware training of lstm-rnns for acoustic modelling," in *2016 ICASSP*. IEEE, 2016, pp. 5280–5284.

[6] Y. Liu, P. Zhang, and T. Hain, "Using neural network front-ends on far field multiple microphones based speech recognition," in *2014 ICASSP*. IEEE, 2014, pp. 5542–5546.

[7] 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 *2013 ICASSP*. IEEE, 2013, pp. 7942–7946.

[8] S. Xue, O. Abdel-Hamid, H. Jiang, and L. Dai, "Direct adaptation of hybrid dnn/hmm model for fast speaker adaptation in lvcsr based on speaker code," in *2014 ICASSP*. IEEE, 2014, pp. 6339–6343.

[9] N. M. Joy, S. R. Kothinti, and S. Umesh, "Fmllr speaker normalization with i-vector: In pseudo-fmllr and distillation framework," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 4, pp. 797–805, 2018.

[10] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[11] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation," in *2017 ASRU*. IEEE, 2017, pp. 16–23.

[12] ——, "Learning latent representations for speech generation and transformation," *arXiv preprint arXiv:1704.04222*, 2017.

[13] W.-N. Hsu and J. Glass, "Extracting domain invariant features by unsupervised learning for robust automatic speech recognition," in *2018 ICASSP*. IEEE, 2018, pp. 5614–5618.

[14] M. K. Chelimilla, S. Kumar, and S. P. Rath, "Joint distribution learning in the framework of variational autoencoders for far-field speech enhancement," in *Accepted in ASRU*. IEEE, 2019.

[15] Y. Jung, Y. Kim, Y. Choi, and H. Kim, "Joint learning using denoising variational autoencoders for voice activity detection." in *Interspeech*, 2018, pp. 1210–1214.

[16] S. Kumar and S. P. Rath, “Far-field speech enhancement using heteroscedastic autoencoder for improved speech recognition,” *Proc. Interspeech 2019*, pp. 446–450, 2019.

[17] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, “beta-vae: Learning basic visual concepts with a constrained variational framework.” *ICLR*, vol. 2, no. 5, p. 6, 2017.

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

[19] F. Seide, G. Li, X. Chen, and D. Yu, “Feature engineering in context-dependent deep neural networks for conversational speech transcription,” in *2011 ASRU*. IEEE, 2011, pp. 24–29.

[20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *2015 ICASSP*. IEEE, 2015, pp. 5206–5210.

[21] 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. CONF. IEEE Signal Processing Society, 2011.

[22] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proceedings of COMPSTAT’2010*. Springer, 2010, pp. 177–186.

[23] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.