Improving Low-Resource CD-DNN-HMM using Dropout and Multilingual DNN Training

Yajie Miao, Florian Metze

Language Technologies Institute, School of Computer Science, Carnegie Mellon University
Pittsburgh, PA, USA
{ymiao,fmetze}@cs.cmu.edu

Abstract

We investigate two strategies to improve the context-dependent deep neural network hidden Markov model (CD-DNN-HMM) in low-resource speech recognition. Although outperforming the conventional Gaussian mixture model (GMM) HMM on various tasks, CD-DNN-HMM acoustic modeling becomes challenging with limited transcribed speech, e.g., less than 10 hours. To resolve this issue, we firstly exploit dropout which prevents overfitting in DNN finetuning and improves model robustness under data sparseness. Then, the effectiveness of multilingual DNN training is evaluated when additional auxiliary languages are available. The hidden layer parameters of the target language are shared and learned over multiple languages. Experiments show that both strategies boost the recognition performance significantly. Combining them results in further reduction in word error rate, achieving 11.6% and 6.2% relative improvement on two limited data conditions.

Index Terms: Dropout, deep neural networks, multilingual learning, speech recognition

1. Introduction

The recently proposed context-dependent deep neural network hidden Markov model (CD-DNN-HMM) has shown superior performance over the traditional state-of-the-art GMM-HMM on automatic speech recognition (ASR) tasks [1, 2, 3]. This acoustic modeling technique differs from the earlier ANN-HMM hybrid systems in that there are more hidden layers in the DNN topology. Moreover, CD-DNN-HMM models the tied context-dependent states directly, rather than takes context-independent phonemes as targets. Previous studies reveal that the number of parameters in CD-DNN-HMM is generally much larger than that of GMM-HMM [4]. For example, the 5-hidden-layer fully-connected CD-DNN-HMM in [4] has 12 times more parameters than its corresponding GMM-HMM system. Due to the large parameter space, CD-DNN-HMM has encountered special challenges when applied to limited training data, e.g., less than 10 hours of transcribed speech [5].

To alleviate the effects of data sparseness, attempts have been made to build sparse DNN, either through imposing regularizers on hidden-layer parameters [4, 6] or through rounding close-to-zero parameters back to zero [4, 7]. Although speeding up model training greatly, these methods fail to improve recognition performance significantly [4]. Meanwhile, in the deep learning community, Hinton et al. [8] proposed dropout to prevent overfitting and showed consistent improvement on various applications such as speech and video object recognition. On each presentation of a training example, dropout randomly omits each hidden unit with a probability which is referred to as *drop factor* in this work. The resulting DNN is an approximate averaging of multiple neural networks sharing parameters but trained separately. Effectiveness of random dropout has been reported on phone classification [8] and large vocabulary continuous speech recognition (LVCSR) [9]. In this paper, we investigate dropout to improve CD-DNN-HMM in the context of low-resource speech recognition. We implement dropout in the manner that activations of DNN hidden units are masked with a binomial distribution governed by the drop factor. Extensive empirical studies are conducted to find the optimal dropout configuration (drop factor, learning rate, etc) for LVCSR. On the GlobalPhone corpus [10], we experiment with two low-resource conditions with 5 hours and 2 hours of training data respectively. Dropout consistently improves the recognition performance of CD-DNN-HMM across different levels of data availability.

Another approach to boosting low-resource CD-DNN-HMM is to take advantage of additional data from other languages or domains. There has been a number of works dedicated to training multilingual bottleneck features [11, 12, 13, 14] or probabilistic posterior features [5, 12] used in tandem systems. In contrast, less attention has been paid to CD-DNN-HMM systems in multilingual settings. On this aspect, [5] employs out-of-language untranscribed speech to help unsupervised pretraining on the target language. This method can benefit both hybrid and tandem systems, especially when training data on the target language becomes highly limited. In this study, we explore the parameter sharing idea introduced in [15] for low-resource CD-DNN-HMM systems. Specifically, parameters of DNN hidden layers are shared and collaboratively learned over multiple languages. Experiments with GlobalPhone demonstrate the advantage of multilingual DNN training with superior recognition performance. Combining this strategy with dropout results in further WER reduction. Compared with the baseline CD-DNN-HMM, the best results we achieve give 6.2% and 11.6% relative improvement on the 5 hours and 2 hours low-resource conditions respectively.

2. CD-DNN-HMM and Dropout

A DNN is a multi-layer perceptron (MLP) which consists of many hidden layers. This section gives a brief review of the components in CD-DNN-HMM systems and describes the application of dropout. The multilingual DNN training strategy is examined in the experiments (see Section 3.5).

2.1. Deep neural network

In CD-DNN-HMM, we train a DNN with a softmax output layer to classify the input acoustic features into classes corresponding to context-dependent tied states. After training, the DNN output is an estimate of the posterior probability $P(s \mid \mathbf{o}_t)$ of each state s given the observation $\mathbf{o}_t$ at time t. On the hidden layers, DNN computes the activations of conditionally independent hidden units given the input vector. When using sigmoid activation, the emission of the *l*-th layer, i.e., the input to the *l+1*-th layer, can be computed as follows:

$$\mathbf{u}_l = \sigma(\mathbf{W}_l \mathbf{u}_{l-1} + \mathbf{b}_l), \quad 1 \leq l < L \qquad (1)$$

where $\mathbf{u}_0 = \mathbf{o}_t$, $\mathbf{W}_l$ is the matrix of connection weights between the *l*-1-th and *l*-th layers, $\mathbf{b}_l$ is the bias vector at the *l*-th layer, $\sigma(x) = (1+\exp(-x))^{-1}$ is the sigmoid function. The output layer produces an estimate of the posterior probability $P(s \mid \mathbf{o}_t)$ of each state *s* given the observation $\mathbf{o}_t$:

$$P(s \mid \mathbf{o}_t) = \frac{\exp(\mathbf{W}_L \mathbf{u}_{L-1} + \mathbf{b}_L)}{\sum_{\bar{S}} \exp(\mathbf{W}_L \mathbf{u}_{L-1} + \mathbf{b}_L)} \qquad (2)$$

DNN is trained through stochastic gradient descent (SGD), in which a cross-entropy cost function over the set of training examples is optimized. The true class label on each speech frame can be obtained by forced-alignment of the observations with the transcripts.

Training DNN directly with error back-propagation (BP) may be problematic in that BP easily gets stuck at poor local optima. Unsupervised pretraining based on restricted Boltzmann machine (RBM) has been shown to mitigate this effect. A RBM is an undirected graphical model with a set of nodes representing visible units and a set of nodes representing hidden units. RBM training involves maximizing the likelihood of the observations with the contrastive divergence algorithm [16]. A stack of RBMs can be trained in a greedy layer-wise manner and used to initialize the parameters of DNN. The first layer of a DNN corresponds to a Gaussian-Bernoulli RBM and each of the other hidden layers corresponds to a Bernoulli-Bernoulli RBM. Interested readers can refer to [17] for details on RBM.

2.2. Speech recognition with CD-DNN-HMM

In ASR, CD-DNN-HMM shares the model structure (phone set, HMM topology, tying of context-dependent states) coming from an initial GMM-HMM model that has been ML-trained on the same data. That model is also used to generate the class label of each frame through forced-alignment. However, CD-DNN-HMM differs from GMM-HMM in that the acoustic model's Gaussian mixtures are replaced with the DNNs. The emission probability of the HMM state *s* can be computed by converting state posteriors in Eq. (2) as follows [1]:

$$P(\mathbf{o}_t \mid s) = \frac{P(s|\mathbf{o}_t) \cdot P(\mathbf{o}_t)}{P(s)} \qquad (3)$$

where $P(s \mid \mathbf{o}_t)$ is the state posterior probability from Eq. (2), $P(s)$ is the state prior probability which can be approximately estimated from the training data by simple counting, the observation probability $P(\mathbf{o}_t)$ is independent of the word sequence and can be ignored.

2.3. DNN training with dropout

Dropout randomly omits each hidden unit with the probability equal to *hidden drop factor* (HDF) when each training frame is presented. This has the effect of performing model averaging over a large number of networks and thus enhancing model generality [8]. In our implementation, two modifications are made to realize dropout training of DNN.

First, in the feed-forward network, the emission of each hidden unit is masked to zero with the probability of HDF. The activation at the *l*-th hidden layer can be rewritten as

$$\mathbf{u}_l = \sigma(\mathbf{W}_l \tilde{\mathbf{u}}_{l-1} + \mathbf{b}_l), \quad 1 \leq l < L \qquad (4)$$

The masked emission at the *l*-1-th hidden layer is obtained from the following operation:

$$\tilde{\mathbf{u}}_{l-1} = \mathbf{u}_{l-1} \otimes \mathbf{v} \qquad (5)$$

where the vector $\mathbf{v}$ has the same dimension as $\mathbf{u}_{l-1}$, $\otimes$ represents element-wise product. Elements in $\mathbf{v}$ are binary variables sampled from a binomial distribution governed by HDF. Since unit activations are conditionally independent given the input, each variable in $\mathbf{v}$ is further simplified as an independent sample from a Bernoulli distribution where $P(v_i = 0)$ = HDF with $1 \leq i \leq |\mathbf{v}|$. Dropout can also be applied to the network input. The same operation in Eq. (5) is performed on the observations with the masking probability which is called *input drop factor* (IDF). The suitable HDF and IDF values will be examined in our experiments.

The dropout network can still be trained with SGD. Most of the configurations (batch size, decaying schedule, momentum, etc) are inherited from the standard BP without dropout. It has been shown that dropout is not sensitive to the choice of these configurations [8]. However, we must set a larger learning rate for dropout than for the standard BP. This is because dropout is training an model ensemble and thus each update must have a large impact [18].

The second change involves compensating DNN model parameters in testing (i.e., decoding for ASR). Specifically, when the training of dropout DNN terminates, the connection matrices are scaled according to the dropout factors, i.e.,

$$\bar{\mathbf{W}}_1 = (1 - IDF) \cdot \mathbf{W}_1$$

$$\bar{\mathbf{W}}_l = (1 - HDF) \cdot \mathbf{W}_l \quad 2 \leq l <= L \qquad (6)$$

Note that dropout is imposed only in the training stage. We can consider this compensation by assuming HDF=0.5. During training, DNN parameters are optimized with half of hidden units randomly deactivated. In testing when dropout is removed, all the hidden units (twice as many as in training) become active, and thus the connection parameters need to be halved.

3. Experiments

3.1. Experimental setup

In our experiments, we use the GlobalPhone corpus [10] which contains recordings of native speakers reading newspapers in up to 19 languages. Among these languages, German (GE) is taken as the target language on which we try to improve CD-DNN-HMM. To evaluate multilingual DNN training, we simulate a multilingual setting by taking Spanish (SP) and Portuguese (PO) from the corpus as the auxiliary languages. When preparing the datasets, we notice that the previously published results are not directly comparable with each other, mainly due to different data partitions, language models and corpus releases. The full GE, SP and PO datasets

Table 1. *Statistics of the datasets used in our experiments.*

	GE	SP	PO
training (Hr)	14.9	17.6	22.7
dev (Hr)	2.0	2.0	1.6
eval (Hr)	1.5	1.7	1.8

in our experiments have the statistics in Table 1.

We are interested in how CD-DNN-HMM performs when training data becomes highly limited. On the target language GE, we experiment with two low-resource conditions with 2 hours and 5 hours of training data respectively. Both subsets are created by randomly selecting 47 utterances (for 5 hours) and 17 utterances (for 2 hours) from each speaker [19, 20]. To ensure coverage of variability, we keep the number of training speakers the same as in the full set.

3.2. Baseline GMM-HMM

We build the standard ML GMM-HMM systems on the full GE 15-hour training set, as well as the 5-hour and 2-hour subsets. In each system, 9 frames of 13-dimensional MFCCs, normalized with per-speaker cepstral mean subtraction, are spliced together and projected down to 40 dimensions with linear discriminant analysis (LDA). The number of context-dependent triphone states (i.e., DNN targets) for the three systems are 2568, 1228 and 894 respectively, with an average of 12, 9 and 6 Gaussian components per state. On top of the ML systems, discriminative training is performed using the boosted maximum mutual information (BMMI) criterion [21]. Figure 2 presents the performance of the BMMI models on the GE dev set and Table 2 presents their results on the eval set.

3.3. Baseline CD-DNN-HMM

On each GE training set, CD-DNN-HMM is directly based on the corresponding ML system built in the previous section. CD-DNN-HMM inherits the HMM structure from the ML system. We use 11 frames (5 on each side) of MFCCs, which are globally normalized to zero mean and unit variance, as DNN input. Layer-wise pretraining is carried out to initialize the network parameters, with the learning rate of 0.005 for Gaussian-Bernoulli RBM and 0.01 for Bernoulli-Bernoulli RBM. Pretraining runs for 20 epochs for each layer. During finetuning, we use an exponentially decaying learning rate schedule for SGD. Specifically, the learning rate starts from 0.08 and remains unchanged until we observe increase of cross-validation (CV) error. Then the learning rate is halved at each epoch until the CV error stops to drop any more. A momentum of 0.5 is used in both pretraining and finetuning for gradient smoothing. The batch size is 128 for pretraining and 256 for finetuning. Under each low-resource condition, pretraining is performed only on the available data. This differs from [5] which takes additional untranscribed speech from GE or even other languages for pretraining. Each DNN hidden layer consists of 1024 units, which is observed to perform better than 512 units and similarly as 2048 units. The performance of baseline CD-DNN-HMM on the GE dev set is shown in Figure 2.

3.4. CD-DNN-HMM with dropout

For the dropout DNN, finetuning uses the same decay schedule, but a much larger starting learning rate which is set to 1.2 in our experiments. There are two variations in dropout configurations: input dropout factor (IDF) and hidden dropout factor (HDF). We first explore the value of HDF by setting IDF=0, i.e., no dropout on the network input. From Figure 1(a), we observe that on both 5-hour and 2-hour sets, CD-DNN-HMM performs best when HDF equals 0.2. Then we fix HDF=0.2 and tune the value of IDF. It turns out that IDF greater than 0 definitely degrades the recognition results (see Figure 1(b)). This contradicts with [8] in which dropout on the observations brings further improvement. We think it is partly because the speech data in our datasets are relatively clean, and thus the denoising effects [22] caused by IDF > 0 are blurred. The impact of IDF on more noisy datasets will be examined in our future work.

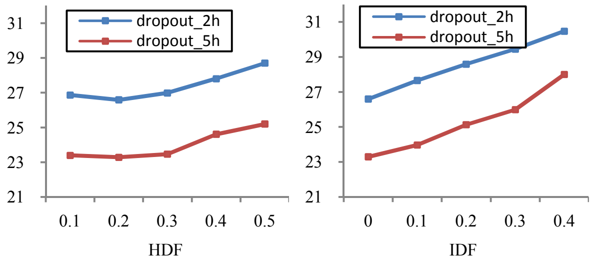
A comparison between dropout (IDF=0, HDF=0.2) and the standard BP is made on the GE dev set and the results are shown in Figure 2. We can see that CD-DNN-HMM provides lower WER than the corresponding BMMI GMM-HMM model. With the same pretraining, dropout outperforms standard BP consistently and performs better when DNN is finetuned with limited training data. Additionally, as the number of hidden layers increases, standard BP encounters overfitting indicated by the degradation of WER. This effect is mitigated by dropout, especially on the 2-hour and 5-hour sets. Figure 3 further demonstrates the ability of dropout to deal with overfitting by showing the frame CV error in finetuning. After some (e.g., 40) epochs, the CV error begins to rise when using standard BP. In contrast, dropout controls overfitting effectively and achieves consistently lower CV error.

For the different DNN configurations in Figure 2, we pick the ones with the lowest WER and use them to decode the GE eval set. The results are shown in Table 2. Again, dropout brings gains to CD-DNN-HMM across various training sets. Its advantage is more pronounced when the amount of data becomes less, e.g., on the 5-hour and 2-hour sets.

3.5. Multilingual DNN training

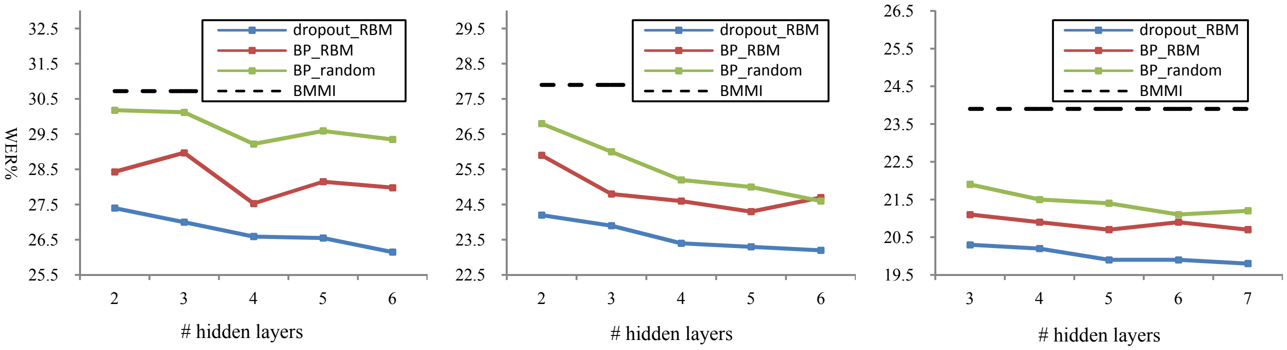
Finally, low-resource CD-DNN-HMM is further improved with multilingual DNN training. To prepare the auxiliary data, we select 5 hours of speech from each source language SP or PO. On the SP_5Hr (or PO_5Hr) set, we build an ML GMM-HMM system which has 1244 (or 1157) tied states and produces the frame-level class labels.

Under each GE limited condition (2 or 5 hours), the DNN parameters, except the softmax layer, are tied across all the languages. In this work, we only focus on how multilingual training helps in DNN finetuning. That is, the additional SP and PO data are not used for pretraining. RBMs trained under this GE condition are used for DNN initialization. DNNs over



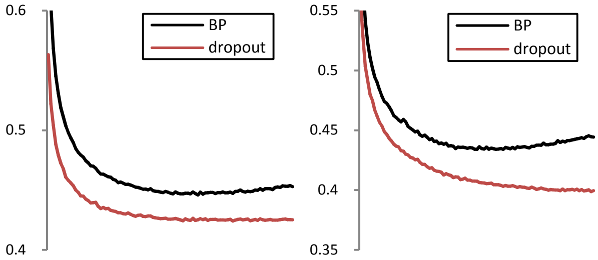
(a) (b)

Figure 1: *WER of CD-DNN-HMM on GE dev set. (a) HDF is varied with IDF=0. (b) IDF is varied with HDF=0.2. DNN has 4 hidden layers on 2 hours and 5 hidden layers on 5 hours.*



(a) *2-hour training set* (b) *5-hour training set* (c) *full training set*

Figure 2: *Comparison between dropout and standard BP in terms of WER% on GE dev set. BP_RBM and dropout_RBM represent BP and dropout with RBM pretraining, while BP_random means that network parameters are randomly initialized.*



(a) *2-hour training set* (b) *5-hour training set*

Figure 3: *CV error rate during DNN finetuning over 100 epochs. The starting learning rate, 0.08 for BP and 1.2 for dropout, is not halved through the 100 epochs. DNN has 4 hidden layers on 2 hours and 5 hidden layers on 5 hours.*

the three languages have the configuration (number of hidden layers) corresponding to the best case in Figure 2. Then each epoch of finetuning traverses data from all the languages, rather than only from GE. To prevent overfitting, the tied DNN parameters are decoupled when learning rate on the target language GE begins to halve.

Table 3 shows how CD-DNN-HMM performs on the GE eval set when different auxiliary data are available. We can see that using SP_5Hr brings significant gains to the baseline CD-DNN-HMM on both 2-hour and 5-hour GE conditions. In comparison, gains obtained from adding more data PO_5Hr become smaller. Dropout can be naturally applied to this multilingual setting, where every language adopts dropout in finetuning. Incorporating dropout reduces WER further down to 24.6% on the 2- hour set and 22.5% on the 5-hour set. This corresponds to 11.6% and 6.2% relative improvement on the two low-resource conditions.

Table 2. *Performance of CD-DNN-HMM on the German evaluation set with various training sets.*

Systems	2 hours	5 hours	full
BMMI	30.4	27.9	25.2
BP_random	29.0	24.6	21.4
BP_RBM	27.8	24.1	21.2
dropout_RBM	26.3	23.1	20.4

Table 3. *Performance of multilingual DNN training on the German evaluation set.*

Method	auxiliary data	GE 2Hr	GE 5Hr
BP_RBM	N/A	27.8	24.1
BP_RBM	SP_5Hr	26.2	23.6
BP_RBM	SP_5Hr+PO_5Hr	25.8	23.4
dropout_RBM	SP_5Hr+PO_5Hr	24.6	22.5

4. Conclusions and Future Work

In this paper, we investigate two strategies to improve CD-DNN-HMM in the context of low-resource speech recognition. Firstly, the dropout method is applied in DNN finetuning to prevent overfitting and improve model robustness. Secondly, we evaluate the effectiveness of multilingual DNN training when additional auxiliary data are available from other languages. Experiments show that both strategies can improve the recognition performance of CD-DNN-HMM especially with sparse training data. Combining them results in further reduction in WER, achieving 11.6% and 6.2% relative improvement on the 2-hour and 5-hour limited conditions respectively.

As discussed in Section 3.4, in the future work, we will examine the effectiveness of input dropout under more noisy datasets. Also, dropout will be extended to the tandem systems where DNN is intended for bottleneck or class posterior front-end extraction.

5. Acknowledgements

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD / ARL) contract number W911NF-12-C-0015. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575.

6. References

[1] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Transactions on Speech and Audio Processing, Special Issue on Deep Learning for Speech and Lang Processing*, 2012.

[2] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech*, pp. 437–440, 2011.

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

[4] D. Yu, F. Seide, G. Li, and L. Deng, "Exploiting sparseness in deep neural networks for large vocabulary speech recognition," in *Proc. ICASSP*, pp. 4409-4412, 2012.

[5] P. Swietojanski, A. Ghoshal, and S. Renals, "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR," in *Proc. SLT*, pp. 246-251, 2012.

[6] G. S. V. S. Sivaram, and H. Hermansky, "Sparse multilayer perceptron for phoneme recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 20, no. 1, pp. 23-29, 2012.

[7] J. Langford, L. Li, and T. Zhang, "Sparse online learning via truncated gradient," *Journal of Machine Learning Research*, vol. 10, pp. 777-801, 2009.

[8] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[9] G. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proc. ICASSP*, 2013.

[10] T. Schultz, "GlobalPhone: a multilingual speech and text database developed at Karlsruhe University," in *Proc. ICLSP*, pp. 345–348, 2002.

[11] N. T. Vu, F. Metze, and T. Schultz, "Multilingual bottle-neck features and its application for under-resourced languages," in *Proc. SLTU*, 2012.

[12] F. Grezl, M. Karafiat, and M. Janda, "Study of probabilistic and bottle-neck features in multilingual environment," in *Proc. ASRU*, pp. 359-364, 2011.

[13] K. Vesely, M. Karafiat, F. Grezl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *Proc. SLT*, pp. 336-341, 2012.

[14] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Proc. ICASSP*, 2013.

[15] S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana, "On the use of a multilingual neural network front-end," in *Proc. Interspeech*, pp. 2711-2714, 2008.

[16] M. A. Carreira-Perpinan, and G. E. Hinton, "On contrastive divergence learning," *Artificial Intelligence and Statistics* , 2005.

[17] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," UTML TR., Deparment of Computer Science, University of Toronto, 2010.

[18] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," *arXiv preprint arXiv:1302.4389*, 2013.

[19] L. Lu, A. Ghoshal, and S. Renals, "Regularized subspace Gaussian mixture models for cross-lingual speech recognition," in *Proc. ASRU*, pp. 365-370, 2011.

[20] Y. Miao, F. Metze, and A. Waibel, "Subspace mixture model for low-resource speech recognition in cross-lingual settings," in *Proc. ICASSP*, 2013.

[21] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Cambridge Univ., Cambridge, U.K., 2003.

[22] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371-3408, 2010.