IMPROVING WIDEBAND SPEECH RECOGNITION USING MIXED-BANDWIDTH TRAINING DATA IN CD-DNN-HMM

Jinyu Li, Dong Yu, Jui-Ting Huang, and Yifan Gong

Microsoft Corporation, One Microsoft Way, Redmond, WA 98052

ABSTRACT

Context-dependent deep neural network hidden Markov model (CD-DNN-HMM) is a recently proposed acoustic model that significantly outperformed Gaussian mixture model (GMM)-HMM systems in many large vocabulary speech recognition (LVSR) tasks. In this paper we present our strategy of using mixed-bandwidth training data to improve wideband speech recognition accuracy in the CD-DNN-HMM framework. We show that DNNs provide the flexibility of using arbitrary features. By using the Mel-scale log-filter bank features we not only achieve higher recognition accuracy than using MFCCs, but also can formulate the mixed-bandwidth training problem as a missing feature problem, in which several feature dimensions have no value when narrowband speech is presented. This treatment makes training CD-DNN-HMMs with mixed-bandwidth data an easy task since no bandwidth extension is needed. Our experiments on voice search data indicate that the proposed solution not only provides higher recognition accuracy for the wideband speech but also allows the same CD-DNN-HMM to recognize mixed-bandwidth speech. By exploiting mixed-bandwidth training data CD-DNN-HMM outperforms fMPE+BMMI trained GMM-HMM, which cannot benefit from using narrowband data, by 18.4%.

Index Terms— deep neural network, log filter bank, CD-DNN-HMM, wideband, narrowband, mixed-bandwidth

1. INTRODUCTION

Recently a new acoustic model named context-dependent deep neural network hidden Markov model (CD-DNN-HMM) [1][2] was proposed. The CD-DNN-HMM has been shown, by many groups [1][2][3][4][5][6][7], to outperform the conventional Gaussian mixture model (GMM)-HMMs in many large vocabulary speech recognition (LVSR) tasks. For example, it reduced errors by 16% on a voice search task [1][2][8] and one-third on the Switchboard phone-call transcription benchmark [3], over discriminatively trained GMM-HMMs.

In this paper, we investigate using mixed-bandwidth training data to improve the recognition accuracy for wideband speech in the CD-DNN-HMM framework. This study has practical importance since it is often the case that we have access to a large amount of narrowband training data but only small amount of wideband training data. This is typically because narrowband speech is easier to get than wideband speech in the past, since recording speech over the telephone is a relatively economical and efficient way to collect large amounts of data from a wide variety of geographic regions. For the voice search application, which is the focus of this study, it is mainly because the data collected from the old mobile devices are sampled at 8-kHz, although the new data are collected at sampling rate of 16-kHz. It is obvious that we should exploit these narrowband data to improve the wideband speech recognition instead of throwing them away.

In the GMM-HMM framework this is a difficult task. Several approaches have been proposed in the past for utilizing the narrowband training data. The simplest approach is to just down sample both the training and testing data so that the wideband speech is treated as the narrowband speech. This is obviously suboptimal since wideband speech contains additional information that is useful to distinguish phones [9][10]. An alternative approach is to extend the bandwidth of a narrowband speech waveform to obtain a wideband waveform [11][12][13][14][15]. The bandwidth extension procedure, however, is quite complicated, often introduces errors and typically requires stereo data to train the extension model. It provides benefit only if little wideband speech is available [11][12]. We have never seen gains in the real world LVSR system when a moderate amount (>50-hrs) of wideband speech is available.

Fortunately, in the CD-DNN-HMM framework exploiting mixed-bandwidth training data can be simple as we will show in this paper. This is because CD-DNN-HMMs have much higher flexibility than GMM-HMMs in using features other than MFCCs. More specifically we demonstrate that using the Mel-scale log-filter bank features we can achieve higher recognition accuracy than using MFCCs on LVSR tasks. This allows us to formulate the mixed-bandwidth training problem as a missing feature problem, in which several feature dimensions have no value when narrowband speech is presented. This treatment makes training CD-DNN-HMMs with mixed-bandwidth data significantly simpler since it does not require bandwidth extension at all. Our experiments on voice search data indicate that the proposed solution not only provides higher recognition accuracy for the wideband speech but also allows the same CD-DNN-HMM to recognize mixed-bandwidth speech, which is important in practice since some users may use Bluetooth microphones or old devices.

The rest of the paper is organized as follows. We will first briefly introduce the CD-DNN-HMM and its three core components in Section 2. We will then compare Mel-scale log-filter bank features with log-FFT spectrum features and MFCCs on the voice search dataset in Section 3. In Section 4 we describe how to design the filter bank so that the narrowband speech can share a subset of filters in the wideband speech. We demonstrate the effectiveness of the proposed approach on the voice search dataset. We summarize our study in Section 5.

2. CD-DNN-HMM

In this section, we briefly describe the key components and the

training/decoding procedures of CD-DNN-HMMs.

2.1 Architecture of CD-DNN-HMMs

As illustrated in Figure 1, in the CD-DNN-HMM we replace the Gaussian mixture model in the conventional GMM-HMM systems with a DNN. We compute the HMM's state emission probability density function $p_{x|y}(x|y=s)$ by converting the state posterior probability $p_{y|x}(y=s|x)$ obtained from the DNN to

$$p_{x|y}(x|y=s) = \frac{p_{y|x}(y=s|x)}{p_y(y=s)} \cdot \mathrm{p}(x), \quad (1)$$

where s is the tied triphone states (also known as senones), x is the acoustic observation vector at the current frame augmented with neighbor frames, $p_y(y=s)$ is the prior probability of state s, and $\mathrm{p}(x)$ is independent of state.

There are three key components in the CD-DNN-HMM shown in Figure 1: modeling senones directly even though there might be thousands or even tens of thousands of senones; using DNNs instead of shallow multi-layer perceptrons; and using a long context window of frames as the input to the DNNs. These components are critical in achieving the huge accuracy improvements reported in [1][2].

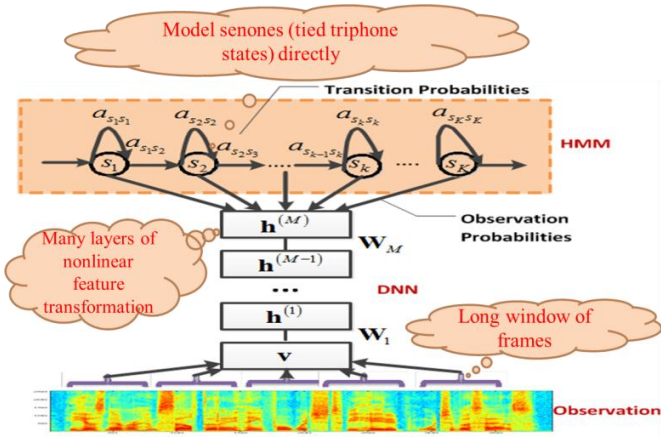

Figure 1: CD-DNN-HMM and its three core components.

2.2 Training and Decoding

In our current implementation, CD-DNN-HMMs are initialized from traditional CD-GMM-HMMs. More specifically, the CD-DNN-HMM inherits the model structure, including the phone set, the HMM topology, and tying of context-dependent states, directly from the CD-GMM-HMM system. In addition, the senone labels used for training the DNNs are extracted from the forced alignment generated using the CD-GMM-HMM. The detailed training procedure, including the bridging between CD-GMM-HMMs and CD-DNN-HMMs as well as the learning rate and momentum values used in the experiments, can be found in [2]. To improve the training speed, GPU is used [3].

Decoding is done by plugging the DNN into a conventional large vocabulary GMM-HMM decoder with tricks also described in [2]. Unlike training, decoding can be carried out in real time even on a single CPU core by exploiting quantization and SIMD architectures in modern CPUs [16].

3. FEATURES

One of the properties that make CD-DNN-HMMs promising for LVSR is the ability to use arbitrary features. To compare with CD-GMM-HMMs, the same MFCC/PLP features were used in the experiments reported in [1][2]. However, it does not prevent CD-DNN-HMMs from using other features.

In [17], it was shown that the Mel-scaled log filter-bank feature outperforms the MFCC feature on the TIMIT phone recognition task using context-independent DNN-HMMs. In this section, we demonstrate that Mel-scaled log filter-bank feature also helps to improve accuracy on a 72-hour voice search task when CD-DNN-HMM is used. We also compare the performance difference between different filter-bank designs in this section.

3.1. Experiment Setup

Our experiments were conducted on a commercial voice search (VS) task. The training data, called VS-1, consists of 72 hours of audio. The test set, called VS-T, has 26757 words in 9562 utterances. Both the training and test sets were collected at 16-kHz sampling rate.

The input feature to the CD-GMM-HMM system is a 36-dimention vector converted using HLDA from the 13-dimension mean-normalized MFCC with up to third-order derivatives. The speaker-independent 3-state cross-word triphones share 1803 senones. Each senone is modeled using a GMM with 20 Gaussian components on average. The CD-GMM-HMM was first trained with maximum likelihood estimation (MLE), and then refined discriminatively using the feature space minimum phone error (fMPE) transformation [18] and boosted maximum-mutual information (BMMI) [19] training.

Following [2], the DNN used in the experiments has 7 hidden layers, each with 2048 nodes. The input to the DNN is a feature vector augmented with previous and next 5 frames (5-1-5). The output layer has 1803 senones, determined by the MLE trained GMM-HMM system. The DNN is initialized using the DBN-pre-training procedure, and then refined with back-propagation using senone labels derived from the MLE model alignment [1].

3.2. Compare Different Features

Table 1 compares the discriminatively-trained CD-GMM-HMM baseline with the CD-DNN-HMMs using different input features. The 13-dimension MFCC feature is extracted from the 24-dimension Mel-scale log filter-bank feature with a truncated DCT transform. All the input features are mean normalized and with dynamic features. The MFCC feature is with up to third-order derivatives, while the log filter-bank feature and the FFT feature have up to the second-order derivatives. The HLDA transform is only applied to the MFCC feature for the CD-GMM-HMM system.

From this table we can make several observations.

First, the CD-DNN-HMM with MFCC feature obtains 8.7% relative word error rate (WER) reduction from the fMPE+BMMI trained CD-GMM-HMM. This agrees with the results reported in [3], which indicates a 16% relative WER reduction over the minimum phone error (MPE) trained CD-GMM-HMM, since fMPE typically provides around 10% relative WER reduction over

discriminatively trained GMM models. The smaller gain compared to that achieved on the SWB dataset seems to be related to the task. In the voice search dataset, all utterances are very short (less than three words per utterance on average) and have much larger percentage of silence. These two factors seem to have adverse effect to the training of CD-DNN-HMM. Our preliminary study indicates that reducing the silence frames during the training can improve the WER of CD-DNN-HMMs on our voice search task.

Table 1: Comparison of different input features for DNN. All the input features are mean-normalized and with dynamic features. Relative WER reduction in parentheses.

Setup	WER (%)
CD-GMM-HMM (MFCC, fMPE+BMMI)	34.66 (**baseline**)
CD-DNN-HMM (MFCC)	31.63 (-8.7%)
CD-DNN-HMM (24 log filter-banks)	30.11 (-13.1%)
CD-DNN-HMM (29 log filter-banks)	30.11 (-13.1%)
CD-DNN-HMM (40 log filter-banks)	29.86 (-13.8%)
CD-DNN-HMM (256 log FFT bins)	32.26 (-6.9%)

Second, we can observe that switching from the MFCC feature to the 24 Mel-scale log filter-bank feature leads to large WER reduction (4.7% relative). Increasing the number of filter banks from 24 to 40 only provides less than 1% relative WER reduction. Overall, CD-DNN-HMM outperforms CD-GMM-HMM trained using fMPE+BMMI by a relative WER reduction of 13.8%. Note that this is achieved with much simpler training procedure than that is used to build the CD-GMM-HMM baseline. Further improvement can be obtained by using sequence-level training [20][21] but this is not the focus of this paper.

Third, using 256 log FFT bins directly severely degrades the ASR performance. We believe this is because the values of log FFT spectrum, although providing extra information, are much less invariant than that of Mel-scale log filter-banks, especially in the high frequency bins.

3.3. Dynamic Features

The dynamic feature can be obtained through a linear transform of the static feature with a context window. Since DNNs are very powerful in transforming features through many layers of nonlinear transformations, one would think that the dynamic feature can be automatically learnt if we can use a longer context window and thus we can eliminate the calculation of dynamic features.

Table 2: Comparison of DNNs with and without dynamic features. All the input features are mean normalized.

CD-DNN-HMM (40 log filter-banks)	WER (%)
static+Δ+ΔΔ (11-frame)	29.86
static only (11-frame)	31.11
static only (19-frame)	30.48

The results in Table 2, however, seem to suggest that dynamic features are useful. In this table, the static feature is a vector of 40 log filter-bank outputs. By using up to second-order delta features and an 11-frame context window we can get 29.86% WER on the test set. Keeping only the static feature and the same 11-frame context window increases the WER from 29.86% to 31.11%. This is expected because it uses fewer frames of static features than the baseline setup.

However, even if we increase the number of context frames to 19, which accounts for 2 frames at each side introduced by the first-order delta and 2 more frames at each side introduced by the second-order delta, the 30.48% WER achieved is still worse than that obtained by the baseline setup. We believe this last 2% relative difference is attributed to the training algorithm which fails to find a better local optimum. For this reason, we should keep using the dynamic features.

3.4. Mean Normalization

Since the voice search data come from different users and environments, there is a large amplitude variation across utterances. Thus in the above experiments, we always apply mean normalization to the Mel-scale log filter-bank feature. From Table 3, however, we surprisingly observe that mean normalization is not necessary when Mel-scale log filter-bank feature is used. In fact, the system without mean normalization performs slightly better than the system with mean normalization. This could be attributed to DNNs' ability to learn more invariant and discriminative features at each higher layer and so the variations at the input are gradually reduced after many layers of processing. Another possible reason is that the data comes from the same resource so that the mean normalization is not very important.

Table 3: Comparison of features with and without mean normalization. Dynamic features are used.

CD-DNN-HMM (29 log filter banks)	WER (%)
With mean normalization	30.11
Without mean normalization	29.96

4. EXPLOITING MIXED-BANDWIDTH TRAINING DATA

Based on the investigation described in Section 3, we can clearly see that we should use the Mel-scale log filter-bank feature as the input to the DNNs. This observation suggests that we can exploit mixed-bandwidth training data in the CD-DNN-HMM framework quite easily. The only question left is how to design the Mel-scale filter banks so that we can align the filter banks of data sampled at 8-kHz sampling rate with the lower filter banks of data sampled at 16-kHz sampling rate. In other words, if we design the filter banks in this way the narrowband data can be considered as wideband data with some feature dimensions missing. The narrowband data can thus be used to optimize the connections between the hidden layers and the lower filter banks and the wideband data can be used to optimize the connections between hidden layers and the higher filter banks.

It turns out that designing such a filter bank is trivial and it has been done in [22]. In this paper, we use the same filter bank design that is described and used in [22]. More specifically we use 22 filter banks for 8-kHz data and 29 filter banks for 16-kHz data. The lower 22 filter banks for 16-kHz data spans 0-4 kHz, and are shared with the 22 filter banks for 8-kHz data. The higher 7 filter banks for 16-kHz data spans 4-8 kHz, with the central frequency of the first higher filter bank as 4kHz. For 8-kHz data, the 7 upper filter banks can be padded with either 0s (zero-padding or ZP) or the mean of those observed in the 16-kHz data (mean-padding or

MP) (Figure 2). The same 29 filter banks are used in Table 1 (row 4) for the wideband speech.

4.1. Empirical Evaluation

To evaluate the proposed approach, we used additional 197 hours of 16-kHz data, called VS-2, to simulate the scenario where DNNs are trained with the mixture of wideband and narrowband speech. The 8-kHz training data is obtained by down sampling the 16-kHz VS-2 training data, and the 8-kHz testing data for testing is obtained in the same way from VS-T 16-kHz testing data.

For fair comparison, the same DNN architecture is used for all setups. The input layer uses 29 Mel-scale log filter-bank outputs (without mean normalization) together with dynamic features and an 11-frame context window. The input layer thus contains 29*3*11=957 nodes. The DNN has 7 hidden layers, each with 2048 nodes. The output layer has 1803 nodes, corresponding to the number of senones determined by the GMM system.

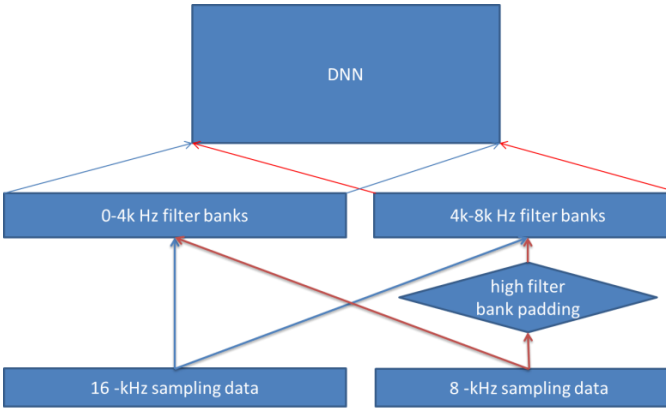

Figure 2: DNN training/testing with 16-kHz and 8-kHz sampling data

Table 4: DNN performance on wideband and narrowband test sets using mixed-bandwidth training data.

Training Data	WER (16-kHz VS-T)	WER (8-kHz VS-T)
16-kHz VS-1 (**B1**)	29.96	71.23
8-kHz VS-1 + 8-kHz VS-2 (**B2**)	-	28.98
16-kHz VS-1 + 8-kHz VS-2 (**ZP**)	28.27	29.33
16-kHz VS-1 + 8-kHz VS-2 (**MP**)	28.36	29.37
16-kHz VS-1 + 16-kHz VS-2 (**UB**)	27.47	53.51

The experimental results are summarized in Table 4. In this table, there are two baselines. One is that using only the 72 hours of 16-kHz VS-1 training data (marked as B1 in the table). In this baseline we just throw away the narrow band training data. As shown in the table, we can achieve 29.96% WER on the wideband test data using this setup. However, since the system has not been exposed to any narrowband training data it performs poorly on the narrowband test data. The second baseline, which is marked as B2 in the table, down samples both the training and test data to 8-kHz. This can be beneficial since it allows for using all the training data available. As indicated in the table, it achieved 28.98% WER on the down sampled test set. This is better than the B1 baseline which only uses wideband training data. This baseline setup, however, is suboptimal for 16-kHz testing data since the information in the frequency range of 4-8 kHz is not exploited.

The results of our proposed approach are summarized as ZP (zero-padding) and MP (mean padding) in Table 4. It is quite obvious that these two padding strategies perform similarly and both outperform the baseline systems on the wideband testing data. For example, using zero-padding, we can achieve 28.27% WER on the wideband test set, which translates to 5.6% and 2.4% relative WER reduction over B1 and B2 setups, respectively. Note that, the systems trained using mixed-bandwidth data also perform very well for narrowband test data, which is a plus since many users may use Bluetooth microphones or old devices. We also point out that by using bandwidth extension techniques we seldom see improvements over B1 and never see improvements over B2 when GMM is used and a reasonable amount of wideband speech is available.

To get the idea how good our proposed approach is getting, we compare it with the upper bound setup (UB in Table 4) which assumes we have access to the same amount of wideband training data as the narrowband data. We can see that the gap between UB and B2 is 1.51% absolute, consistent with what we obtained in the CD-GMM-HMM system internally in Microsoft. Our approach, which used the mixed-bandwidth training data, recovered half of the gap. When compared to B1, which is a poorer baseline, our approach recovered two-thirds of the gap.

4.2. Analysis

To understand the power that many layers of nonlinear feature transformation in DNNs brings, we take the output vectors at each layer h for the 8-kHz and 16-kHz input feature pair and measure their Euclidean distance,

$$d_h\left(x^{(16k)}, x^{(8k)}\right) = \sqrt{\sum_{i=1}^{N_h}\left(h_i^{(16k)} - h_i^{(8k)}\right)^2}, \quad (2)$$

where N_h is the number of nodes at the hidden layer h, and h_i is the value of the i-th node at that layer. For the top layer, whose output is the senone posterior probability, we calculate their KL-divergence in nats,

$$d_y\left(x^{(16k)}, x^{(8k)}\right) = \sum_{j=1}^{N} p_{y|x}\left(s_j|x^{(16k)}\right)\log\frac{p_{y|x}\left(s_j|x^{(16k)}\right)}{p_{y|x}\left(s_j|x^{(8k)}\right)}, \quad (3)$$

where N is the number of senones, and s_j is the senone id. Table 5 shows the statistics of d_h and d_y over randomly sampled 40K frames in the test set for both the DNN trained using wideband speech only (UB setup in Table 4) and that trained using mixed-bandwidth data (ZP setup in Table 4).

From Table 5 we can observe that in both DNNs, the distance between hidden layer vectors generated from the 8-kHz and 16-kHz input feature pair is significantly reduced at the layers close to the output layer compared to that in the first hidden layer. However, what's more interesting is that the average distances and variances in the data-mix DNN are consistently smaller than that in the 16-kHz DNN. This indicates that by using mixed-bandwidth training data, the DNN learns to consider the difference in the wideband and narrowband input features as irrelevant variations. These variations are suppressed after many layers of nonlinear transformation. The final representation is thus more invariant to

this variation and yet still has the ability to distinguish between different senones. This behavior is even more obvious at the output layer since the KL-divergence between the paired outputs is only 0.22 in the mixed-data DNN and is much smaller than 2.03 that is observed in the 16-kHz DNN. This explains why the mixed-data DNN significantly outperforms the 16-kHz DNN when the narrowband testing set is used.

Table 5: The Euclidean distance (ED) for the output vectors at each hidden layer (L1-L7) and the KL-divergence (in nats) for the posterior vectors at the top layer between 8-kHz and 16-kHz input features

	16-kHz DNN (UB)		Data-mix DNN (ZP)	
Layer	**Mean (ED)**	**Variance (ED)**	**Mean (ED)**	**Variance (ED)**
L1	13.28	3.90	7.32	3.62
L2	10.38	2.47	5.39	1.28
L3	8.04	1.77	4.49	1.27
L4	8.53	2.33	4.74	1.85
L5	9.01	2.96	5.39	2.30
L6	8.46	2.60	4.75	1.57
L7	5.27	1.85	3.12	0.93
Layer	**Mean (KL)**		**Mean (KL)**	
Top layer	2.03		0.22	

5. SUMMARY

In this paper, we proposed a simple and effective technique to improve wideband speech recognition in CD-DNN-HMMs by exploiting mixed-bandwidth training data. Our approach is based on the observation that DNN has the flexibility of using arbitrary features and that Mel-scale log filter-bank feature outperforms the MFCC feature in CD-DNN-HMMs. We thus can formulate and reduce the mixed-bandwidth training problem into a missing feature problem by designing the filter-bank wisely.

Our experiments on the voice search task clearly indicate the effectiveness of our proposed approach, which achieved 5.6% and 2.4% relative WER reduction, respectively, over the system trained using only the wideband data (B1) and that trained using narrowband data by down sampling wideband speech (B2). By comparing with the oracle upper bound, which can only be achieved if the same amount of wideband speech is available, our proposed approach recovered two thirds and one half of the gaps between the upper bound and that of B1 and B2, respectively. Overall, by exploiting the mixed-bandwidth training data CD-DNN-HMM outperforms fMPE+BMMI trained GMM-HMM, which cannot benefit from using narrowband data, by 18.4%.

We point out that exploiting mixed-bandwidth training data in the GMM framework is much more difficult and much less effective. Actually using bandwidth extension techniques we seldom see improvements over B1 and never see improvements over B2 when GMM is used and a reasonable amount of wideband speech is available.

In this paper we have also explored three properties of the CD-DNN-HMMs. First, CD-DNN-HMMs provide flexibility of using arbitrary features. We believe that features better than Mel-scale filter-bank may be discovered in the near future to further boost CD-DNN-HMMs' performance. Second, CD-DNN-HMM has the ability to generate more invariant and selective features at higher hidden layers as demonstrated in our analysis of the 16-kHz DNN and mixed-data DNN. This ability allows us to just feed in heterogeneous data collected under different environments and expect DNNs to reduce the mismatch and be robust to the variation. Third, building a state-of-the-art LVSR system using CD-DNN-HMM is much easier than using GMM-HMM. We believe these properties would make CD-DNN-HMM a very promising model for LVSR.

REFERENCES

[1] D. Yu, L. Deng, and G. Dahl, "Roles of pretraining and fine-tuning in context-dependent DNN-HMMs for real-world speech recognition," in *Proc. NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, Dec. 2010.

[2] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Trans. Speech and Audio Proc.*, vol. 20, no. 1, pp. 30 – 42, 2012.

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

[4] D. Yu, F. Seide, G. Li, J. Li, and M. Seltzer, "Why deep neural networks are promising for large vocabulary speech recognition," submitted to *IEEE Trans. on Audio, Speech, and Language Processing*, 2012.

[5] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "An application of pretrained deep neural networks to large vocabulary conversational speech recognition," *Tech. Rep. 001*, Department of Computer Science, University of Toronto, 2012.

[6] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Improvements in using deep belief networks for large vocabulary continuous speech recognition," *Tech. Rep. UTML TR 2010-003*, Speech and Language Algorithm Group, IBM, February 2011

[7] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, A.-r. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition", in *Proc. ASRU 2011*, pp. 30-35.

[8] D. Yu, Y. C. Ju, Y. Y. Wang, G. Zweig, and A. Acero, "Automated directory assistance system --- from theory to practice," in *Proc. Interspeech*, 2007, pp. 2709–2711.

[9] P. Moreno and R. M. Stern, "Sources of degradation of speech recognition in the telephone network," in *Proc. ICASSP*, Adelaide, Australia, vol. I, pp.109-112, Apr. 1994.

[10] X. Huang, A. Acero, and H. -W. Hon, *Spoken Language Processing*, Prentice-Hall, May 2001.

[11] M. L. Seltzer and A. Acero, "Training wideband acoustic models using mixed-bandwidth training data for speech recognition", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 235 – 245, 2007.

[12] M. L. Seltzer, A. Acero, and J. Droppo, "Robust bandwidth extension of noise-corrupted narrowband speech," in *Proc. Interspeech*, pp. 1509-1512, 2005.

[13] Y. M. Cheng, D. O'Shaughnessy, and P. Mermelstein, "Statistical recovery of wideband speech from narrowband speech," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 4, pp. 544–548, Oct. 1994.

[14] K.-Y. Park and H. S. Kim, "Narrowband to wideband conversion of speech using GMM based transformation," *in*

Proc. ICASSP, Istanbul, Turkey, Jun. 2000, vol. 3, pp. 1843–1846.

[15] P. Jax and P. Vary, "Wideband extension of telephone speech using a hidden Markov model," in *IEEE Workshop on Speech Coding*, Delavan, WI, Sep. 2000, pp. 133–135.

[16] A. Senior V. Vanhoucke and M. Z. Mao (2011), "Improving the speed of neural networks on CPUs," in *Proc. Deep Learning and Unsupervised Feature Learning Workshop, NIPS*, 2011.

[17] A. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modelling", in *Proc. ICASSP*, pp. 4273-4276, 2012.

[18] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau and G. Zweig, "fMPE: discriminatively trained features for speech recognition," in *Pro. ICASSP*, 2005.

[19] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon and K. Visweswariah, "Boosted MMI for model and feature space discriminative training", in *Proc. ICASSP*, 2008

[20] A. Mohamed, D. Yu, and L. Deng, "Investigation of full-sequence training of deep belief networks for speech recognition", in *Proc. Interspeech 2010*, pp. 1692-1695.

[21] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Proc. ICASSP 2009*, pp. 3761–3764.

[22] X. Fan, M. Seltzer, J. Droppo, H. Malvar, and A. Acero, "Joint encoding of the waveform and speech recognition features using a transform codec," in *Proc. ICASSP*, pp.5148-5151, May 2011.