Roles of Pre-Training and Fine-Tuning in Context-Dependent DBN-HMMs for Real-World Speech Recognition

Dong Yu, Li Deng
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
{dongyu,deng}@microsoft.com

George E. Dahl
Department of Computer Science
University of Toronto
Ontario, Canada
gdahl@cs.toronto.edu

Abstract

Recently, deep learning techniques have been successfully applied to automatic speech recognition tasks -- first to phonetic recognition with context-independent deep belief network (DBN) hidden Markov models (HMMs) and later to large vocabulary continuous speech recognition using context-dependent (CD) DBN-HMMs. In this paper, we report our most recent experiments designed to understand the roles of the two main phases of the DBN learning -- pre-training and fine tuning -- in the recognition performance of a CD-DBN-HMM based large-vocabulary speech recognizer. As expected, we show that pre-training can initialize weights to a point in the space where fine-tuning can be effective and thus is crucial in training deep structured models. However, a moderate increase of the amount of unlabeled pre-training data has an insignificant effect on the final recognition results as long as the original training size is sufficiently large to initialize the DBN weights. On the other hand, with additional labeled training data, the fine-tuning phase of DBN training can significantly improve the recognition accuracy.

1 Introduction

Automatic speech recognition (ASR) has been the subject of a significant amount of research and commercial development for many years. Systems in which ASR is one of the key components have been widely deployed in mobile phones, desktop/tablet computers, automobiles, call centers, and voice-mail systems. For example, using an ASR-enabled mobile phone, users can say a person's name or a phone number to make a call and can speak instead of typing a query to retrieve information of interest. Unfortunately, even after decades of research the performance of ASR systems in real-world usage scenarios remains far from satisfactory.

Almost all of the state-of-the-art large vocabulary continuous speech recognition (LVCSR) systems are hidden Markov model (HMM) based. An HMM is a generative model in which the observable acoustic features are assumed to be generated from a hidden Markov process that makes transitions between states. In conventional HMMs used for ASR, the observation probabilities are modeled using Gaussian mixture models (GMMs). These GMM-HMMs are typically first trained to maximize the likelihood of generating the observed features and then fine-tuned using discriminative training and/or large margin techniques [1][2]. The potential of GMM-HMMs, however, is restricted by limitations of the GMM observation distribution model.

Attempts have been made to develop models beyond the conventional GMM-HMM so that discriminative training becomes an inherent part of the model and so that the model is able to learn rich, distributed representations of its input. For example, artificial neural networks (ANNs) have been proposed as potential GMM replacements to estimate observation probabilities in the HMMs [3]. The resulting model is typically called an ANN-HMM. Alternatively, the phone posterior probabilities estimated by ANNs can be combined with the speech feature vectors to form an augmented feature vector, which can be used as the input to conventional GMM-HMMs. Such a model is normally referred to as the ANN-HMM tandem structure [4]. Combinations of ANNs and HMMs, although promising, are limited by the properties of ANNs trained with backpropagation. For instance, it is difficult to exploit more than two hidden layers well using the conventional ANN training algorithms [5][6].

Recently, a new architecture, the deep belief network (DBN)-HMM, has been proposed for ASR. Context-independent (CI)-DBN-HMMs have been shown to outperform context-dependent (CD)-GMM-HMMs on TIMIT phone recognition tasks [7][8]. The CD-DBN-HMMs have been shown to perform much better than the CD-GMM-HMMs on a real-world LVCSR task [22]. The performance gain mainly comes from using DBNs to estimate observation probabilities. DBNs have been demonstrated to be effective for many practical applications since it was proposed [10][11].

Since CD-DBN-HMMs are new LVCSR models, many questions remain unanswered. In this paper, we use the Bing mobile voice search (BMVS) task to evaluate whether unsupervised pre-training is important in learning CD-DBN-HMM model parameters and how important it is to increase the amount of unlabeled data for pre-training and labeled data for fine-tuning. We demonstrate that unsupervised pre-training is crucial in exploiting multiple hidden layers. However, increasing the fine-tuning (labeled) data is much more important than increasing the pre-training (unlabeled) data. This is endorsed by the fact that negligible recognition accuracy improvement was observed if only the pre-training data were doubled while significant gain was obtained if the fine-tuning data were doubled. Agreeing with common intuition, our explanation is that the pre-training phase only puts the weights into a relatively good range and fine-tuning is the key step to discriminate patterns.

The rest of the paper is organized as follows. In Section 2, we discuss related work using deep learning techniques for ASR. In Section 3, we describe the architecture and associated training procedure of CD-DBN-HMMs. We show experimental results on BMVS task in Section 4 and conclude the paper in Section 5.

2 Related Work

Deep learning techniques have been successfully applied to applications such as natural language processing [12][13], language recognition [14], image processing [15], vision [16], audio encoding [17], audio classification [18], phone recognition [19], and semantic tagging [20] in the last several years. However, it was only very recently that DBN-HMMs were proposed and shown to be promising for ASR.

The first convincing evidence on the effectiveness of DBN-HMMs for speech recognition was presented in [8], where the CI-DBN-HMM was proposed and successfully applied to the TIMIT phone recognition task. In CI-DBN-HMMs the DBNs were trained to predict the state of context-independent phones and the HMMs were used to model the speech sequences. It was shown that even though the CI-DBN-HMM only models context-independent phones, it can significantly outperform the CD-GMM-HMMs that model context-dependent phones.

The DBN-CRF (conditional random field) model proposed in [7] improved the DBN-HMM model of [8] in three areas. First, it uses the CRF to replace the HMM in modeling the sequential information. Second, it optimizes the utterance conditional log-likelihood instead of the frame conditional log-likelihood when learning the DBN weights. Third, the sequential discriminative learning technique developed in [7] jointly optimizes the DBN weights, CRF's transition weights, and phone language model parameters. The DBN-CRF, with its associated joint optimization algorithm, achieved higher accuracy than the DBN-HMM phone recognizer trained using the frame-discriminative criterion implicit in the DBN's fine tuning procedure as implemented in [8] at the cost of much higher computational

complexity.

To further improve CI-DBN-HMM performance, the work of [21] incorporated a more powerful first layer model, the mcRBM, in order to better model the covariance structure of mel-scale filterbank DBN input features. mcRBM, however, is significantly more difficult to train than RBM.

The CI-DBN-HMM was extended to the CD-DBN-HMM in [22] and the task was changed from phone recognition to LVCSR. Experiments on the challenging BMVS dataset collected under real usage conditions demonstrate that the CD-DBN-HMM significantly outperforms the state-of-the-art CD-GMM-HMM systems. Three factors contribute to the success: 1) the use of tied tri-phone context-dependent units (or senones) as the DBN modeling units; 2) the use of the best available tri-phone GMM-HMM to generate the senone alignment; and 3) tuning of the transition probabilities. Experiments also indicate that the decoding of a five-hidden-layer CD-DBN-HMM is almost as fast as the already highly optimized, state-of-the-art tri-phone GMM-HMM.

In previous work, the same training set was used for both phases of DBN training to train the DBN-HMMs (or DBN-CRFs). It has not been studied in the context of LVCSR whether the pre-training and fine-tuning steps are equally important and whether the labeled and unlabeled data are equally valuable in training DBN-HMMs. This paper provides experimental verification of common wisdom about the role of pre-training and fine-tuning for CD-DBN-HMM systems for LVCSR and guides future experimental efforts.

3 CD-DBN-HMM

In this section, we review the architecture of the CD-DBN-HMMs proposed in [22], which serves as the platform in which experiments reported in this paper were conducted.

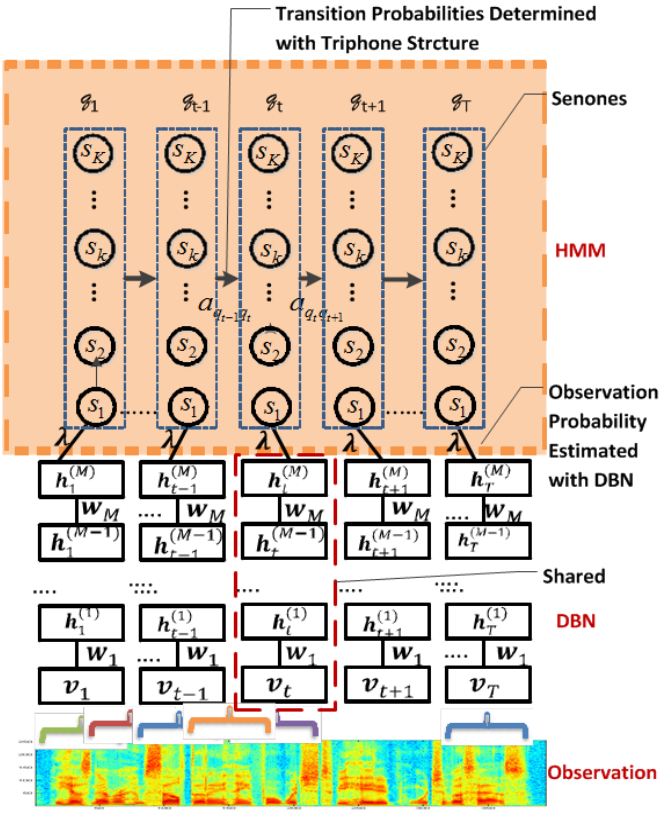

Figure 1: Illustration of the CD-DBN-HMM architecture.

Figure 1 illustrates the architecture of the CD-DBN-HMMs, in which a DBN is used to estimate the observation probabilities and an HMM is used to model the state transitions. The DBN, which takes a window of (typically 11) frames as its input, moves one frame at a time and generates a vector of posterior probabilities of the tied-triphone states (also called senones). The posterior probabilities are converted into likelihoods by dividing them by the priors of the states.

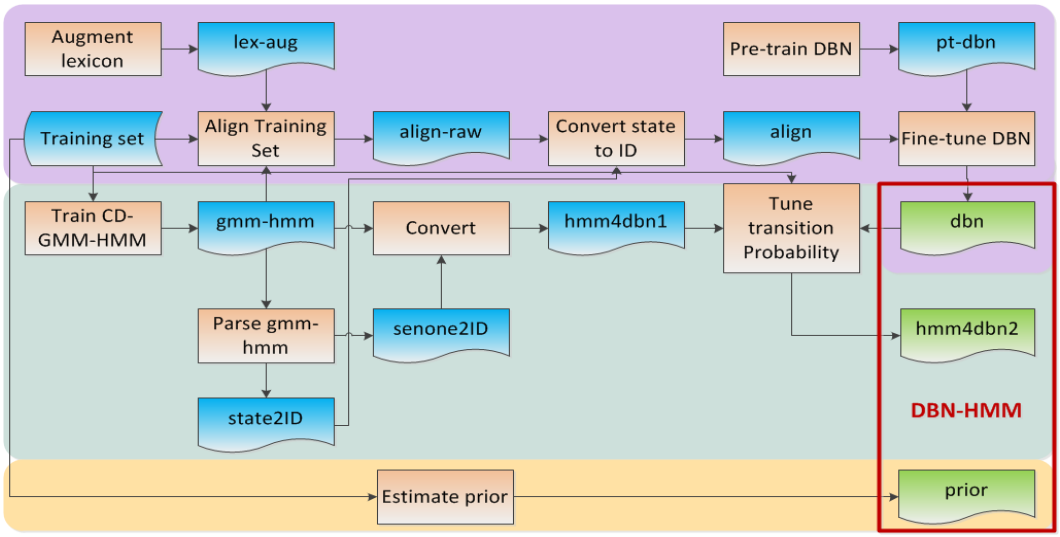

Figure 2: The procedure to train CD-DBN-HMMs.

Figure 2 describes the key steps in training the CD-DBN-HMMs. The basic idea is to use the forced alignment to obtain frame-level senone labels for training the DBN and to borrow the triphone tying structure and transition probabilities from the CD-GMM-HMMs. The transition probabilities can be further tuned after the DBNs are trained. Here is a summary of the computational steps in the training:

1. Train a standard CD GMM-HMM system, called *gmm-hmm*, where state tying is determined based on the data-driven decision tree;
2. Parse *gmm-hmm*; give each senone an ordered *senoneid* starting from 0; generate a mapping , called *state2id*, from each physical tri-phone state (e.g., b-ah+t.s2) to the corresponding *senoneid;*
3. Convert *gmm-hmm* to DBN-HMM, called *hmm4dbn1*, using senone-to-*senoneid* mapping;
4. Pre-train each layer in the DBN bottom-up layer by layer; the pre-trained DBM is called *ptdbn*;
5. Augment the lexicon, called *lex-aug*, with short pause (sp), and silence (sil) appended. lex-aug;
6. Use *gmm-hmm* and *lex-aug* to generate a state-level alignment, called align-raw, on the training data;
7. Convert *align-raw* to *align* where each physical triphone state is converted to *senoneid*;
8. Use *senoneid* associated with each frame in *align*to fine-tune the DBN, called the result as *dbn*, using back-propagation initialized with *ptdbn*;
9. Estimate the prior state-occupation probability;
10. Re-estimate the transition probabilities using *dbn* and *hmm4dbn1* to maximize the likelihood of observing the features. This new DBN-HMM is called hmm4dbn2.
11. Exit if no recognition accuracy improvement is observed in the development set; Otherwise use *dbn, hmm4dbn2,* and *lex-aug* to generate a new state-level alignment align-raw on the training set and goto Step 7.

The key difference (excluding the use of DBN pre-training) between the CD-DBN-HMM architecture and the earlier CD-ANN-HMM architecture [3] is that we use senones as the DBN output units directly. This change provides two primary advantages. First, we can implement the CD-DBN-HMM system with only minimal modifications to an existing CD-GMM-HMM system. Second, any improvement in modeling units that are incorporated into the CD-GMM-HMM baseline system, such as cross-word triphone models, will be accessible to the CD-DBN-HMM through the use of the shared training labels.

Conventional wisdom holds that DBN has a better prediction capability than a GMM for static input-output patterns. We can easily show that the same prediction capability extends to dynamic or sequential input-output patterns if we use the DBN-HMMs. This proof was provided in [22].

4 Experimental Results

The experiments were conducted on the data collected from the Bing mobile voice search application (formerly known as Live Search for mobile [9]). This is a real-world large-vocabulary spontaneous and continuous speech recognition task. It allows the mobile users to do accurate and efficient business and web search from their mobile phones via voice. The dataset used in our experiments was collected under real usage scenarios in 2008, at which time the application was restricted to do location and business lookup. All audio files collected were sampled at 8 kHz and encoded with the GSM codec. This is a challenging task since the dataset contains all kinds of variations: noise, music, side-speech, accent, sloppy pronunciation, hesitation, repetition, interruption, and different audio channels. The dataset contains 130 hours of labeled and 2000 hours of lightly supervised (based on users' click information) training data, 6.5 hours (or 8,777 utterances) of development (validation) data, and 9 hours (or 12,758 utterances) of test data. All queries in the training set were collected before those in the development set, which in turn was collected before those in the test set. For the sake of easy comparisons, we have used the public lexicon from Carnegie Mellon University. The language model used in the evaluation contains a 65K word vocabulary, 3.2M word bi-grams, and 1.5M word tri-grams.

The sentence accuracy of a state-of-the-art CD-GMM-HMM system on this task discriminatively trained on 24 hours of labeled data using the minimum phone error criterion is 65.5% for the development set and 63.8% for the test-set . Using the same 24 hours of labeled training data, CD-DBN-HMMs achieved 71.7% and 69.6% sentence accuracy on the development and test sets, respectively, when the CD-DBN-HMMs contained 5 hidden layers each with 2048 units.

The focus of this study is to evaluate how the pre-training and fine-tuning phases affect the recognition accuracy. For this purpose, we designed three experiments.

The first experiment verifies that pre-training is important. We compared systems with and without the pre-training step in training CD-DBN-HMMs with one and two hidden layers using 24 hours of labeled training data. Table 1 summarizes the experimental results on this setup. From Table 1 we can observe that when only one hidden layer was used no significant difference was observed with and without pre-training. However, when two hidden layers were used, pre-training becomes important as the development set sentence accuracy is only slightly better than that in the one hidden layer system when DBN weights were not pre-trained, and much higher improvement was observed when pre-training was performed. It is a clear indication that pre-training is indeed very important when training these deep networks.

Table 1: Comparison of sentence accuracy with and without pre-training

number of hidden layers	with pre-training	dev-set accuracy
1	No	68.0%
1	Yes	68.1%
2	No	68.2%
2	Yes	69.6%

The second experiment determines to what degree adding more data for pre-training and fine-tuning helps boosting the recognition accuracy. To evaluate this, we prepared a second 24-hour labeled training set and compared the performance with and without using this additional 24-hour set for pre-training and fine-tuning on a 5 hidden-layer 2048 hidden-unit DNBs. To get the results shown in Table 2, we have set the pre-training epochs N to 50 for Gaussian-Bernoulli RBMs and

20 for Bernoulli-Bernoulli RBMs, and set the fine-tuning epochs M to 12. This translates to 62 hours of pre-training and 17 hours of fine-tuning time using 24 hours of training data with GPU. Comparing row two with row one in Table 2 we can see that doubling the pre-training data size, which doubles the pre-training time, does not improve accuracy. However, if we double the fine-tuning data size, as shown in row three, we can obtain 2.2% and 1.8% sentence accuracy improvement on the development and test sets, respectively. These results are consistent with the usual intuition (explored in [5]) that pre-training only brings weights to a good region of weight space where fine-tuning can be effective. Adding additional pre-training data alone may only put the weights around the similar location. Adding fine-tuning data, however, has greater potential to discriminate between different classes and thus is more likely to lead to better recognition accuracy. Also observable from Table 2 is that if we reduce the training epochs as we increase the training size to make the total training time the same, most gains from using the additional fine-tuning data are washed away. This indicates that adding more data is only helpful if we can afford to run enough epochs to take advantage of the extra training data.

Table 2: Comparison of sentence accuracy with and without using additional 24 hours of data for pre-training and fine-tuning

units per layer	pre-training data	pre-training epochs	fine-tuning data	fine-tuning epochs	dev-set accuracy	test-set accuracy
2048	24 hrs	N	24 hrs	M	71.7%	69.6%
2048	48 hrs	N	24 hrs	M	71.7%	69.7%
2048	48 hrs	N	48 hrs	M	73.9%	71.5%
2048	48 hrs	N/2	24 hrs	M	72.3%	69.7%
2048	48 hrs	N/2	48 hrs	M/2	72.8%	71.0%

Adding additional pre-training and fine-tuning data may allow us to get better results using more hidden units. The third experiment, whose results are shown in Table 3, was designed to answer this question. Note that the only configuration difference between Table 3 and Table 2 is the usage of 2560 (instead of 2048) units per hidden layer in Table 3, which translates to 50% more weights in the DBN. By comparing Table 3 and Table 2, we can observe that increasing the hidden units only slightly improves the accuracy both with and without using additional training data. This seems to suggest that to see effect of using more hidden units we need much more training data. In addition, we can notice the same behavior as observable in Table 2 that doubling the pre-training data alone does not make difference to the performance while doubling the fine-tuning data boosted the performance significantly.

Table 3: Comparison of sentence accuracy using more hidden units with and without using additional 24 hours of data for pre-training and fine-tuning

units per layer	pre-training data	pre-training epochs	fine-tuning data	fine-tuning epochs	dev-set accuracy	test-set accuracy
2560	24 hrs	N	24 hrs	M	71.7%	69.8%
2560	48 hrs	N	24 hrs	M	71.9%	69.8%
2560	48 hrs	N	48 hrs	M	74.3%	71.7%
2560	48 hrs	N/2	24 hrs	M	72.0%	70.1%
2560	48 hrs	N/2	48 hrs	M/2	73.0%	70.8%

Overall, by doubling the fine-tuning data and using 2560 units in hidden layers we can achieve 71.7% sentence accuracy on the test set. This represents a 7% relative reduction of recognition errors compared with the best results reported in [22] on the same task.

5 Summary and Future Research Directions

Clear evidence was presented in [8] that the deep learning technique is capable of outperforming state-of-the-art GMM-HMM systems in the phonetic recognition task of TIMIT. Significant research has since been carried out to extend the basic deep learning architecture to enable real-world, large vocabulary speech recognition applications. One key extension is to develop elaborately constructed context-dependent phone states as the output units of the otherwise standard DBN [22]. In this paper, we reported our most recent experiments designed to understand the roles of the pre-training and fine-tuning phases of the DBN learning in the system performance of a large-vocabulary continuous speech recognizer. We also reported the results on the system performance as a function of the training data size. The results presented in this paper suggest that as far as recognition accuracy is concerned, a moderate increase in the amount of unlabeled pre-training data has an insignificant effect on the final recognition results as long as the original training size is sufficiently large to initialize the DBN weights. As expected however, with additional labeled training data, the fine-tuning phase of DBN training is more effective at separating different speech classes. Note, however, due to the current resource limitations, we have not been able to conduct extensive experiments to examine whether using orders of magnitude more unlabeled data for pre-training can improve the DBN-HMM system performance, although our experiments do indicate that the value of unlabeled data is significantly less than that of the labeled data in DBN-HMMs. This suggests that if algorithms other than the gradient based approaches [23] are to be developed to scale up the training we should focus more on the fine-tuning phase.

Deep learning is an emerging technology for ASR as well as for other information processing fields. Despite the empirical promising results reported in this and other recent papers, much needs to be developed. Human information processing mechanisms (e.g., vision and speech) clearly suggest the need of deep architectures for extracting complex structure and building internal representation from rich sensory inputs. For example, human speech production and perception systems are both equipped with clearly layered hierarchical structures in transforming the information from the waveform level to the linguistic level in the perception mode and in the opposite direction in the generation mode [24][25]. The DBN architecture studied so far has not been compatible with many of the key properties in the human speech production and perception mechanisms, albeit a significant advancement over the GMM architecture currently in use in major speech recognizers. We need to develop deeper understanding of the power of deep learning in terms of theory, architecture, computational algorithm, and implementation. We need to develop better feature extraction models at each layer of the DBN and other deep learning architectures. To enable real-world success of deep learning based ASR, we also need to develop effective and scalable parallel algorithms to train deep models, and to develop effective adaptation techniques for deep models as has been successfully done for HMM-based systems. The latter is of special importance as the speech data distributions under the deployment conditions are typically different from the training data distribution in common real-world ASR applications. Finally, we need to develop better deep architectures than the current DBN and its variants for modeling sequential data that respect essential temporal properties in human speech including its elastic timing. The DBN-HMM and DBN-CRF that we have explored represent highly simplistic and loose integration to exploit the power of DBNs in static pattern recognition. More advanced models that embed and exploit hierarchically dynamic structure in natural speech using DBN-inspired architectures and learning as the constituents of the overall model are important to further improve the performance of speech recognition as a most important sequential classification task with wide-spread practical applications.

References

[1] Yu, D., Deng, L., He, X. and, Acero, A, "Large-margin minimum classification error training: A theoretical risk minimization perspective," *Computer Speech and Language*, vol. 22, no. 4, 2008, pp. 415-429.

[2] He, X., Deng, L., Chou, W. "Discriminative learning in sequential pattern recognition --- A unifying review for optimization-oriented speech recognition," *IEEE Sig. Proc. Mag.*, vol. 25,

2008, pp. 14-36.

[3] Renals, S., Morgan, N., Boulard, H., Cohen, M., and Franco, H., "Connectionist probability estimators in hmm speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 161–174, 1994.

[4] Morgan, N., Qifeng, Z., A. Stolcke, K. Sonmez, S. Sivadas, T. Shinozaki, M. Ostendorf, P. Jain, H. Hermansky, D. Ellis, G. Doddington, B. Chen, O. Cretin, H. Bourlard, and M. Athineos, "Pushing the envelope—Aside," *IEEE Signal Process. Mag.*, vol. 22, no. 5, pp. 81–88, 2005.

[5] Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P. and Bengio, S., "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, 11(625--660), 2010.

[6] Bengio, Y. and Glorot, X., "Understanding the difficulty of training deep feedforward neural networks", Proc. *AISTATS* 2010, Chia Laguna Resort, Sardinia, Italy, pages 249-256, 2010.

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

[8] Mohamed, A., Dahl, G.E., and Hinton, G. "Deep belief networks for phone recognition," *NIPS Workshop on deep learning for speech recognition and related applications*, 2009.

[9] Acero, A., Bernstein, N., Chambers, N., Ju, Y., Li, X., Odell, J., Nguyen, P., Scholtz, O., and Zweig, G. "Live search for mobile: Web services by voice on the cellphone," in Proc. ICASSP, 2008, pp. 5256–5259.

[10] Hinton, G., Osindero, S., and Teh, Y. "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527-1554, 2006.

[11] Hinton, G. E. and Salakhutdinov, R., "Reducing the dimensionality of data with neural networks", *Science*, Vol. 313. no. 5786, pp. 504 - 507, 28 July 2006.

[12] Collobert, R. and Weston, J. "A unified architecture for natural language processing: Deep neural networks with multitask learning," Proc. *ICML*, 2008.

[13] Deselaers, T., Hasan, S., Bender, O. and Ney, H. "A deep learning approach to machine transliteration," Proc. *4th Workshop on Statistical Machine Translation*, pp. 233–241, 2009.

[14] Yu, D., Wang, S., Karam, Z., Deng, L. "Language recognition using deep-structured conditional random fields," Proc. *ICASSP*, pp. 5030-5033, 2010.

[15] Ranzato, M. and Hinton, G. E., "Modeling pixel means and covariances using factored third-order Boltzmann machines", Proc. *CVPR*. 2010.

[16] Lee, H., Ekanadham, C. and Ng, A.Y., "Sparse deep belief net model for visual area V2", *NIPS* 2008.

[17] Deng, L., Seltzer, M., Yu, D., Acero, A., Mohamed, A., and Hinton, G. "Binary coding of speech spectrograms using a deep auto-encoder," in Proc. *Interspeech, 2010*.

[18] Lee, H., Largman, Y., Pham, P., Ng, A. "Unsupervised feature learning for audio classification using convolutional deep belief networks," Proc. *NIPS*, Dec. 2009.

[19] Yu, D. and Deng, L. "Deep-structured hidden conditional random fields for phonetic recognition," Proc. *Interspeech*, 2010.

[20] Yu, D., Wang, S., Deng, L. "Sequential labeling using deep-structured conditional random fields", J. *of Selected Topics in Signal Processing - special issue on statistical learning methods for speech and language processing*, 2010.

[21] Dahl, G., Ranzato, M., Mohamed, A., and Hinton, G. "Phone recognition with the mean-covariance restricted Boltzmann machine" *Advances in Neural Information Processing Systems, vol. 24*, 2010.

[22] Dahl, G., Yu, D., and Deng, L. "Context-dependent DBN-HMMs for large vocabulary continuous speech recognition," *IEEE Trans. Audio, Speech, & Language Proc*. (under review).

[23] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, 86(11):2278-2324, November 1998.

[24] Baker, J., et. al. "Research developments and directions in speech recognition and understanding," *IEEE Sig. Proc. Mag.*, vol. 26, May 2009, pp. 75-80.

[25] Deng, L. "Computational models for speech production," in *Computational Models of Speech Pattern Processing*, pp. 199-213, Springer, 1999.