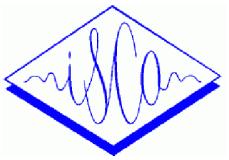

INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Scalable Minimum Bayes Risk Training of Deep Neural Network Acoustic Models Using Distributed Hessian-free Optimization

Brian Kingsbury, Tara N. Sainath, Hagen Soltau

IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA
{bedk,tsainath,hsoltau}@us.ibm.com

Abstract

Training neural network acoustic models with sequence-discriminative criteria, such as state-level minimum Bayes risk (sMBR), been shown to produce large improvements in performance over cross-entropy. However, because they entail the processing of lattices, sequence criteria are much more computationally intensive than cross-entropy. We describe a distributed neural network training algorithm, based on Hessian-free optimization, that scales to deep networks and large data sets. For the sMBR criterion, this training algorithm is faster than stochastic gradient descent by a factor of 5.5 and yields a 4.4% relative improvement in word error rate on a 50-hour broadcast news task. Distributed Hessian-free sMBR training yields relative reductions in word error rate of 7–13% over cross-entropy training with stochastic gradient descent on two larger tasks: Switchboard and DARPA RATS noisy Levantine Arabic. Our best Switchboard DBN achieves a word error rate of 16.4% on `rt03-FSH`.

Index Terms: deep learning, discriminative training, second-order optimization, distributed computing

10.21437/Interspeech.2012-3

1. Sequence-discriminative training

In the past few years there has been a resurgence in the use of neural networks for machine learning tasks, including acoustic modeling for speech recognition. The development of pretraining algorithms [1] and better forms of random initialization [2], as well as the availability of faster computers, has made it possible to train deeper networks than before, and in practice these deep networks have achieved excellent performance [3].

Neural networks trained as acoustic models have usually been trained to perform frame classification using the cross-entropy criterion. In contrast, sequence-discriminative criteria such as maximum mutual information (MMI) [4] and minimum Bayes risk [5, 6] have become the norm for Gaussian mixture model (GMM) acoustic models. In [7] it was shown that the lattice-based machinery developed for sequence-discriminative training of GMMs can be used for neural networks, and that the sMBR criterion improves word error rate by 18% relative over cross-entropy on a 50-hour English broadcast news task. One shortcoming of the experiments in [7] is that the networks were underparameterized for the amount of training data, using only 384 quinphone states and 153K weights. Recent work on a small-scale, 18-hour Wall Street Journal task, shows that MMI improves word error rate by 24% relative over cross-entropy using a more reasonable parameterization of 6,000 triphone states and 2.1M weights [8].

A shortcoming of both studies [7, 8] is that they use small training sets: GMM acoustic models are routinely trained on hundreds to thousands of hours of audio. Both generative pre-training and discriminative cross-entropy training of a deep neural network using 9,300 triphone states and 45.1M parameters (16.1M non-zero parameters following sparsification) has been scaled to a 300-hour Switchboard task by using GPGPU hardware and caching training data in memory [3]. However, even with high-performance hardware and careful algorithmic development, training still required about 30 days [9]. Sequence-discriminative training is potentially even more expensive because the lattices required for the gradient computation are too large to cache in memory. This motivates exploration of distributed algorithms that split computation and I/O across multiple nodes in a compute cluster.

2. Hessian-free optimization

The challenge in performing distributed optimization is to find an algorithm that uses large data batches that can be split across compute nodes without incurring excessive overhead, but that still achieves performance competitive with stochastic gradient descent. One class of algorithms for this problem uses second-order optimization, with large batches for the gradient and much smaller batches for stochastic estimation of the curvature [10, 11, 12]. A distributed implementation of one such algorithm has already been applied to learning an exponential model with a convex objective function for a speech recognition task [11].

The current study uses Hessian-free optimization [10] because it is specifically designed for the training of deep neural networks, which is a non-convex problem. Let $\boldsymbol{\theta}$ denote the network parameters, $\mathcal{L}(\boldsymbol{\theta})$ denote a loss function, $\nabla\mathcal{L}(\boldsymbol{\theta})$ denote the gradient of the loss with respect to the parameters, $\mathbf{d}$ denote a search direction, and $\mathbf{B}(\boldsymbol{\theta})$ denote a matrix characterizing the curvature of the loss around $\boldsymbol{\theta}$. The central idea in Hessian-free optimization is to iteratively form a quadratic approximation to the loss,

$$\mathcal{L}(\boldsymbol{\theta}+\mathbf{d}) \approx \mathcal{L}(\boldsymbol{\theta}) + \nabla\mathcal{L}(\boldsymbol{\theta})^T\mathbf{d} + \frac{1}{2}\mathbf{d}^T\mathbf{B}(\boldsymbol{\theta})\mathbf{d}$$

and to minimize this approximation using conjugate gradient (CG), which accesses the curvature matrix only through matrix-vector products $\mathbf{B}(\boldsymbol{\theta})\mathbf{d}$ that can be computed efficiently for neural networks [13]. If $\mathbf{B}(\boldsymbol{\theta})$ were the Hessian and conjugate gradient were run to convergence, this would be a matrix-free Newton algorithm. In the Hessian-free algorithm, the conjugate gradient search is truncated, based on the relative improvement in approximate loss, and the curvature matrix is the Gauss-Newton matrix [14], which unlike the Hessian is guaranteed positive semidefinite, with additional damping: $\mathbf{G}(\boldsymbol{\theta}) + \lambda\mathbf{I}$.

Our implementation of Hessian-free optimization, which is illustrated as pseudocode in Algorithm 1, closely follows that of [10], except that it currently does not use a preconditioner. Gradients are computed over all the training data.

Algorithm 1 Hessian-free optimization (after [10]).

initialize $\boldsymbol{\theta}$; $\mathbf{d}_0 \leftarrow \mathbf{0}$; $\lambda \leftarrow \lambda_0$; $\mathcal{L}_{\text{prev}} \leftarrow \mathcal{L}(\boldsymbol{\theta})$
while not converged **do**
  $\mathbf{g} \leftarrow \nabla\mathcal{L}(\boldsymbol{\theta})$
  Let $q_{\boldsymbol{\theta}}(\mathbf{d}) = \nabla\mathcal{L}(\boldsymbol{\theta})^T\mathbf{d} + \frac{1}{2}\mathbf{d}^T(\mathbf{G}(\boldsymbol{\theta}) + \lambda\mathbf{I})\mathbf{d}$
  $\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\} \leftarrow \text{CG-MINIMIZE}(q_{\boldsymbol{\theta}}(\mathbf{d}), \mathbf{d}_0)$
  $\mathcal{L}_{\text{best}} \leftarrow \mathcal{L}(\boldsymbol{\theta} + \mathbf{d}_N)$
  for $i \leftarrow N-1, N-2, \ldots, 1$ **do** ▷ *backtracking*
    $\mathcal{L}_{\text{curr}} \leftarrow \mathcal{L}(\boldsymbol{\theta} + \mathbf{d}_i)$
    if $\mathcal{L}_{\text{prev}} \geq \mathcal{L}_{\text{best}} \wedge \mathcal{L}_{\text{curr}} \geq \mathcal{L}_{\text{best}}$ **then**
      $i \leftarrow i + 1$
      break
    $\mathcal{L}_{\text{best}} \leftarrow \mathcal{L}_{\text{curr}}$
  if $\mathcal{L}_{\text{prev}} < \mathcal{L}_{\text{best}}$ **then**
    $\lambda \leftarrow \frac{3}{2}\lambda$; $\mathbf{d}_0 \leftarrow \mathbf{0}$
    continue
  $\rho = (\mathcal{L}_{\text{prev}} - \mathcal{L}_{\text{best}})/q_{\boldsymbol{\theta}}(\mathbf{d}_N)$
  if $\rho < 0.25$ **then**
    $\lambda \leftarrow \frac{2}{3}\lambda$
  else if $\rho > 0.75$ **then**
    $\lambda \leftarrow \frac{3}{2}\lambda$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\mathbf{d}_i$; $\mathbf{d}_0 \leftarrow \beta\mathbf{d}_N$; $\mathcal{L}_{\text{prev}} \leftarrow \mathcal{L}_{\text{best}}$

Gauss-Newton matrix-vector products are computed over a sample (about 1% of the training data) that is taken each time `CG-Minimize` is called. The loss, $\mathcal{L}(\boldsymbol{\theta})$, is computed over a held-out set. `CG-Minimize`$(q_{\boldsymbol{\theta}}(\mathbf{d}), \mathbf{d}_0)$ uses conjugate gradient to minimize $q_{\boldsymbol{\theta}}(\mathbf{d})$, starting with search direction $\mathbf{d}_0$. This function returns a series of steps $\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}$ that are then used in a backtracking procedure. The parameter update, $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\mathbf{d}_i$, is based on an Armijo rule backtracking line search. $\beta < 1.0$ is a momentum term.

To perform distributed computation, we use a master/worker architecture in which worker processes distributed over a compute cluster perform data-parallel computation of gradients and curvature matrix-vector products and the master implements the Hessian-free optimization and coordinates the activity of the workers. All communication between the master and workers is via sockets.

3. Experiments

3.1. English Broadcast News

We first present results for our proposed algorithm on a 50-hour English broadcast news task [7]. We compare the performance of four models: a speaker-adaptive discriminatively trained (SA DT) GMM, a deep belief network (DBN) trained using cross-entropy, a DBN trained using stochastic gradient descent (SGD) and sMBR, and a DBN trained using the distributed Hessian-free algorithm (HF) and sMBR. Development is done on the DARPA EARS `dev04f` set. Testing is done on the DARPA EARS `rt04` evaluation set. System evaluation is in terms of word error rate (WER).

The GMM system is trained using our standard recipe [15], which is briefly described below. The raw acoustic features are 13-dimensional perceptual linear predictive (PLP) features with speaker-based mean, variance, and vocal tract length normalization. Temporal context is included by splicing 9 successive frames of PLP features into supervectors, then projecting to 40 dimensions using linear discriminant analysis (LDA). The feature space is further diagonalized using a global semi-tied covariance (STC) transform [16]. The GMMs are speaker-adaptively trained, with a feature-space maximum likelihood linear (FMLLR) transform estimated per speaker in training and testing. Following maximum-likelihood training of the GMMs, feature-space discriminative training (FMMI) and model-space discriminative training are done using the boosted maximum mutual information (BMMI) criterion. At test time, unsupervised adaptation using regression tree MLLR is performed. The GMMs use 2,220 quinphone states and 85K diagonal-covariance Gaussians, for a total of 6.9M trainable parameters. The recognition vocabulary contains 84K words with 1.08 pronunciation variants per word. The language model contains a total of 272M n-grams, and is an interpolated back-off 4-gram model smoothed using modified Kneser-Ney smoothing. The vocabulary and language model are those used in [7], which provides a more detailed description.

model	`dev04f`	`rt04`
SA DT GMM	14.9	14.5
SA DBN cross-entropy	15.5	15.2
SA DBN SGD sMBR	13.3	13.1
SA DBN HF sMBR	12.9	12.6

Table 1: Comparison of FMMI+BMMI discriminatively trained GMM with DBN models on English broadcast news tasks.

The DBN systems use the same FMLLR features and 2,220 quinphone states as the GMM system described above, with a 9-frame context (± 4) around the current frame, and use five hidden layers each containing 1,024 sigmoidal units. The DBNs therefore contain 6.8M trainable parameters. FMLLR features are used instead of FMMI features because FMMI features were found to offer no advantage for DBN acoustic models [17]. The DBN training begins with greedy, layerwise, pretraining based on restricted Boltzmann machines (RBMs) [1], and then continues with discriminative training, using stochastic gradient descent and the cross-entropy criterion. The procedure used is very similar to that described in [18], except that the samples are randomized frame-by-frame instead of utterance-by-utterance during both the generative pretraining and discriminative cross-entropy training [17], and is also quite similar to the procedures used in [3]. The stochastic gradient descent and Hessian-free sMBR training are initialized with the DBN weights trained with cross-entropy.

The results are presented in Table 1. The comparison between the GMM and DBN systems is done primarily to understand how DBNs compare to existing, state-of-the-art methods. The best DBN result, obtained with the minimum Bayes risk criterion and Hessian-free training (DBN HF sMBR) is 13% relative better than the GMM system on `rt04`. Comparing the DBN results, the minimum Bayes risk training yields relative improvements of 14–17% over cross-entropy training, with Hessian-free optimization outperforming stochastic gradient descent. The distributed Hessian-free training is also faster due to parallelization: training required 18 hours elapsed time using Hessian-free optimization, while stochastic gradient descent required 101 hours. It is more difficult to compare the training cost in CPU time. The Hessian-free training ran on 49 machines, consuming 882 CPU hours. An individual run of stochastic gradient descent uses only one machine, and thus consumes about 8.8 times less CPU time; however, we frequently do multiple stochastic gradient runs in parallel with different choices for the initial step size and weight decay, which is not necessary for the Hessian-free optimization. In practice the

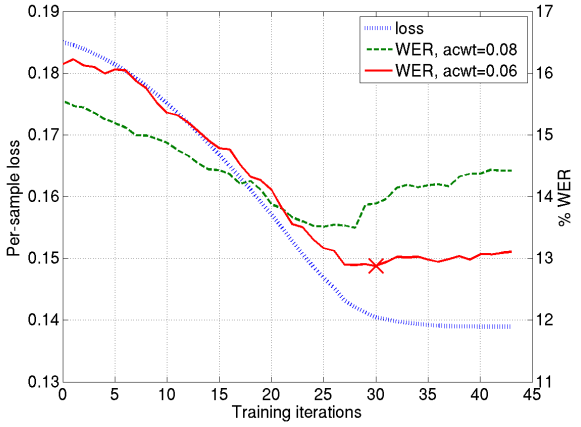

Figure 1: Held-out loss and `dev04f` word error rate as a function of training iterations on the 50-hour broadcast news task. The "X" marks the best `dev04f` word error rate.

two algorithms use roughly comparable amounts of CPU time.

To illustrate the behavior of Hessian-free sMBR optimization, the per-sample loss on a held-out set and the word error rate on `dev04f` are plotted as a function of the number of training iterations in Figure 1. A training iteration is defined as a single iteration of the `while` loop in Algorithm 1. Two word error rate curves are plotted, corresponding to decoding with acoustic weights of 0.08 and 0.06, which are the optimal weights at the beginning and end of the sMBR training, respectively. The loss improves smoothly over the course of the optimization, while the word error rates behave less regularly, which is expected because they are discrete losses. As reported in [7], the best acoustic scaling reduces with sMBR training.

3.2. Switchboard

Next, we demonstrate the scalability of distributed Hessian-free training on a larger task: 300 hours of conversational American English telephony data from the Switchboard corpus. We compare speaker-independent (SI) and speaker-adaptive (SA) GMM models with either maximum-likelihood (ML), model-space BMMI (BMMI), or feature- and model-space BMMI (FMMI+BMMI) training to DBN models trained using either cross-entropy optimized using stochastic gradient descent or sMBR optimized using the distributed Hessian-free algorithm. We do not test stochastic gradient descent sMBR training because the experiment would have taken too long. Development is done on the `Hub5'00` set, while testing is done on the `rt03` set, where we report performance separately on the Switchboard (`SWB`) and Fisher (`FSH`) portions of the set.

The GMM systems are trained using the same methods described above. The speaker-adaptive results include adaptation using regression tree MLLR. The speaker-independent GMMs use 9,300 quinphone states and 370K Gaussians, for a total of 30M trainable parameters. The speaker-adaptive GMMs use 8,260 quinphone states and 372K Gaussians, for a total of 30.1M trainable parameters. The recognition vocabulary contains 30.5K words with 1.08 pronunciation variants per word. The language model is small, containing a total of 4.1M n-grams, and is an interpolated back-off 4-gram model smoothed using modified Kneser-Ney smoothing. Both the lexicon and language model are described in more detail in [19].

	rt03		Hub5'00
model	FSH	SWB	SWB
SI ML GMM	27.5	38.4	23.3
SI BMMI GMM	25.4	36.0	21.4
SI FMMI+BMMI GMM	22.6	33.3	18.9
SA FMMI+BMMI GMM	17.6	26.3	15.1
SI DBN cross-entropy	18.9	28.8	16.1
SI DBN HF sMBR	16.4	25.5	13.3

Table 2: Comparison of GMM and DBN models on Switchboard tasks.

The DBN models and training procedure, including block size for randomization, are patterned after those in [3]. The input features are the same 40-dimensional PLP+LDA+STC features used in the speaker-independent GMM system, excluding the FMMI transform. An input of 11 frames of context (± 5 around the current frame) is provided as input to the DBNs, which use six hidden layers each containing 2,048 hidden units, to estimate the posterior probabilities of the same 9,300 quinphone units used by the SI GMM system. The DBNs contain a total of 41M trainable parameters. The same training steps are used as in the 50-hour broadcast news task: first, generative pretraining with RBMs, then discriminative training with the cross-entropy criterion and stochastic gradient descent, and a final optional sMBR optimization using distributed Hessian-free training. Some differences from [3] are that we use one less hidden layer and we reduce the step size in training based on performance on a held-out set [20].

The word error rate results are presented in Table 2. Prior to sMBR training, the performance of the DBN, which uses speaker-independent features, is between that of the speaker independent and speaker adapted GMMs. Following sMBR training, the DBN is the best model. It is 27% better than the SI GMM on `rt03-FSH` and 13% better than the DBN trained with cross-entropy.

3.3. Noisy Levantine Arabic

Finally, we demonstrate the versatility of distributed Hessian-free sMBR training by testing the method on a very different task: recognition of noisy conversational Levantine Arabic speech collected and distributed for the DARPA RATS program. The training set comprises conversational telephony audio, some from the Fisher corpus and some collected and transcribed by the Linguistic Data Consortium specifically for the RATS program, that has been transmitted over eight different high-frequency radio channels, where a channel corresponds to a specific transmitter-receiver pair. These experiments use 20–38 hours of audio per channel, for a total of 249 hours of training data. The audio is provided at a 16kHz sampling rate, but we downsample to 8kHz to match the usable bandwidth in the signals. The raw features are 13-dimensional PLP features with speaker-based mean- and variance normalization, but no vocal tract length normalization. Nine successive frames of PLP features are concatenated and projected to a 40-dimensional feature vector with LDA, diagonalized with an STC transform, and then are transformed to a canonical space with an FMLLR transform that is computed per-speaker using an auxiliary GMM. The language model is trained only from the acoustic transcripts, contains 600K n-grams, and is a 3-gram model smoothed with modified Kneser-Ney smoothing. The lexicon is a graphemic lexicon of 37.6K words, with no pronunciation variants.

condition	cross-entropy WER	HF sMBR WER	% rel. impr.
clean	58.1	50.7	13
noisy	72.0	66.8	7

Table 3: Comparison of cross-entropy and Hessian-free minimum Bayes risk training on clean and noisy Levantine Arabic from the DARPA RATS program.

We test two deep neural network (DNN)[1] acoustic models. The DNNs take 13 frames of 40-dimensional FMLLR features as input, have four hidden layers each containing 512 hyperbolic tangent units, and estimate the probabilities of 4,096 quingraph units. The DNNs have a total of 3.2M trainable parameters. One DNN is trained with the cross-entropy criterion and stochastic gradient descent (with utterance-by-utterance randomization). The other DNN uses sMBR and distributed Hessian-free optimization, and is initialized from the cross-entropy DNN. The RATS DNNs were to be used for a keyword search task in which deletions would be especially expensive, as they would increase the miss rate of the search; therefore, we used a modified sMBR criterion [21] in which silence is automatically counted as wrong to counteract the tendency of discriminatively trained models to make more deletions. We did not test stochastic gradient descent sMBR training because the experiment would have taken too long.[2]

We measure word error rate on the 16K word `dev04f` Levantine Arabic conversational telephony set from the DARPA EARS program, comparing performance for clean audio, prior to transmission, and results averaged over the eight different channels. The results are summarized in Table 3. The modified sMBR training improves word error rates in both cases, with about half as much improvement for the noisy case.

4. Conclusions

We have presented a scalable, distributed algorithm based on Hessian-free optimization for state-level minimum Bayes risk training of deep neural network acoustic models. On a 50-hour broadcast news task we observe an improvement in word error rate of 4% relative and a 5.5x reduction in training time using distributed Hessian-free optimization instead of stochastic gradient descent to do sMBR training. On a 300-hour Switchboard task, we observe an improvement of 13% relative on the `rt03-FSH` set, from 18.9% to 16.4% word error rate, using HF optimization and sMBR instead of stochastic gradient optimization and cross-entropy. On a 249-hour noisy Levatine Arabic task from the DARPA RATS program, we observe relative improvements of 7–13% using HF optimization and sMBR instead of stochastic gradient optimization and cross-entropy. The results on the two larger tasks show that our algorithm is scalable and applicable to a variety of speech recognition tasks.

5. Acknowledgments

We thank James Martens for his prompt and clear answers to our questions about Hessian-free optimization. This work was supported in part by Contract No. D11PC20192 DOI/NBC under the RATS program. The views, opinions, findings and recommendations contained in this article are those of the authors and should not be interpreted as representing the views or policies, either expressed or implied, of the DOI/NBC.

[1] We make a distinction between deep belief networks, which are generatively pretrained, and deep neural networks, which are initialized from random weights.

[2] Stochastic gradient training was started, but then abandoned after it had run for a week without completing a single epoch.

6. References

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

[2] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010, pp. 249–256.

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

[4] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. ICASSP*, 1986.

[5] J. Kaiser, B. Horvat, and Z. Kačič, "A novel loss function for the overall risk criterion based discriminative training of HMM models," in *Proc. ICSLP*, 2000.

[6] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. ICASSP*, 2002.

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

[8] G. Wang and K. C. Sim, "Sequential classification criteria for NNs in automatic speech recognition," in *Proc. Interspeech*, 2011, pp. 441–444.

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

[10] J. Martens, "Deep learning via Hessian-free optimization," in *Proc. Intl. Conf. on Machine Learning (ICML)*, 2010.

[11] R. Byrd, G. M. Chin, W. Neveitt, and J. Nocedal, "On the use of stochastic Hessian information in unconstrained optimization," *SIAM Journal on Optimization*, vol. 21, no. 3, pp. 977–995, 2011.

[12] O. Vinyals and D. Povey, "Krylov subspace descent for deep learning," in *Proc. NIPS Workshop on Optimization and Hierarchical Learning*, 2011.

[13] B. A. Pearlmutter, "Fast exact multiplication by the Hessian," *Neural Computation*, vol. 6, no. 1, pp. 147–160, 1994.

[14] N. N. Schraudolph, "Fast curvature matrix-vector products for second-order gradient descent," *Neural Computation*, vol. 14, pp. 1723–1738, 2004.

[15] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," in *Proc. IEEE Workshop on Spoken Language Technology*, 2010, pp. 97–102.

[16] M. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[17] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Improvements in using deep belief networks for large vocabulary continuous speech recognition," Tech. Rep., IBM, 2012.

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

[19] S. F. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, and G. Zweig, "Advances in speech transcription at IBM under the DARPA EARS program," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1596–1608, 2006.

[20] N. Morgan and H. Bourlard, "Neural networks for statistical recognition of speech," *Proc. IEEE*, vol. 83, no. 5, pp. 742–772, May 1995.

[21] D. Povey and B. Kingsbury, "Evaluation of proposed modifications to MPE for large scale discriminative training," in *Proc. ICASSP*, 2007, vol. IV, pp. 321–324.