Sequence-discriminative training of deep neural networks

Karel Veselý[1], *Arnab Ghoshal*[2], *Lukáš Burget*[1], *Daniel Povey*[3]

[1]Brno University of Technology, Czech Republic
[2]Centre for Speech Technology Research, University of Edinburgh, UK
[3]Center for Language and Speech Processing, Johns Hopkins University, USA
iveselyk@fit.vutbr.cz, a.ghoshal@ed.ac.uk, burget@fit.vutbr.cz, dpovey1@jhu.edu

Abstract

Sequence-discriminative training of deep neural networks (DNNs) is investigated on a 300 hour American English conversational telephone speech task. Different sequence-discriminative criteria — maximum mutual information (MMI), minimum phone error (MPE), state-level minimum Bayes risk (sMBR), and boosted MMI — are compared. Two different heuristics are investigated to improve the performance of the DNNs trained using sequence-based criteria — lattices are re-generated after the first iteration of training; and, for MMI and BMMI, the frames where the numerator and denominator hypotheses are disjoint are removed from the gradient computation. Starting from a competitive DNN baseline trained using cross-entropy, different sequence-discriminative criteria are shown to lower word error rates by 8-9% relative, on average. Little difference is noticed between the different sequence-based criteria that are investigated. The experiments are done using the open-source Kaldi toolkit, which makes it possible for the wider community to reproduce these results.
Index Terms: speech recognition, deep learning, sequence-criterion training, neural networks, reproducible research

1. Introduction

This paper presents a reproducible set of experiments on speech recognition with a deep neural network (DNN) - hidden Markov model (HMM) hybrid. In such hybrid setups the DNN is used to provide pseudo-likelihooods ("scaled likelihoods") for the states of an HMM [1]. While computational constraints limited earlier uses of hybrid systems to estimating scaled likelihoods for monophones using a two layered network [2] and recurrent networks [3], recent years have seen a resurgence in their use [4, 5, 6, 7]. The principal modeling and algorithmic difference to previous systems is the use of RBM pretraining [8].

Neural networks (NNs) for speech recognition are typically trained to classify individual frames based on a cross-entropy criterion (section 2.1). Speech recognition, however, is inherently a sequence classification problem. As such, speech recognizers using Gaussian mixture model (GMM) as the emission density of an HMM achieve state-of-the-art performance when trained using sequence-discriminative criteria like maximum mutual information (MMI) [9, 10], boosted MMI (BMMI) [11], minimum phone error (MPE) [12] or minimum Bayes risk (MBR) [13, 14, 15]. It is possible to efficiently estimate the parameters based on any of these criteria using statistics collected from lattices [10, 12].

The theory for sequence-discriminative training of neural networks was also developed in early literature [16, 17]. In fact, the "clamped" and "free" posteriors described in [16] are same as the numerator and denominator occupancies [10] used in discriminative training of GMM-HMM systems. The logical extension of this fact is that sequence-discriminative training of NNs can take advantage of the lattice-based computations that are routinely used for GMM-HMM systems. This was pointed out in [18], where it was shown that the sequence-discriminative training can improve upon networks trained using cross-entropy. Subsequent results reported in [19, 20, 6] have also shown consistent gains from sequence-discriminative training of NNs. However, there is some disagreement about which of the criteria is suitable: [18, 20] suggest using a state-level minimum Bayes risk (sMBR) criterion, while [19] finds MMI to work better than MPE, and [6] only provide results using MMI.

Needless to say, such empirical observations depend on the choice of the dataset and specific details of the implementation. In this paper, we present a comparison of the different training criteria for DNNs on the 300-hour Switchboard conversational telephone speech task, which has also been used in [5, 20]. We do this using the Kaldi speech recognition toolkit [21], which is a free, open-source toolkit for speech recognition research. The tools and scripts used to produce the results reported in this paper are publicly available as part of the Kaldi toolkit[1], and anyone with access to the data should be able to reproduce our results.

2. Acoustic modeling with DNNs

In a DNN-HMM hybrid system, the DNN is trained to provide posterior probability estimates for the HMM states. Specifically, for an observation $\mathbf{o}_{ut}$ corresponding to time t in utterance u, the output $y_{ut}(s)$ of the DNN for the HMM state s is obtained using the softmax activation function:

$$y_{ut}(s) \triangleq P(s|\mathbf{o}_{ut}) = \frac{\exp\{a_{ut}(s)\}}{\sum_{s'} \exp\{a_{ut}(s')\}}, \tag{1}$$

Karel Veselý and Lukáš Burget were supported by the IARPA BABEL program, the IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070 and Czech Ministry of Education project No MSM0021630528. Arnab Ghoshal was supported by EPSRC Programme Grant no. EP/I031022/1 (Natural Speech Technology). Daniel Povey was supported by DARPA BOLT contract No HR0011-12-C-0015, and IARPA BABEL contract No W911NF-12-C-0015, and the Human Language Technologies Center of Excellence. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors alone and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DARPA/DoD, or the U.S. Government.

[1]Available from http://kaldi.sf.net/

where $a_{ut}(s)$ is the activation at the output layer corresponding to state s. The recognizer uses a pseudo log-likelihood of state s given observation $\mathbf{o}_{ut}$,

$$\log p(\mathbf{o}_{ut}|s) = \log y_{ut}(s) - \log P(s), \quad (2)$$

where $P(s)$ is the prior probability of state s calculated from the training data [1].

The networks are trained to optimize a given training objective function using the standard *error backpropagation* procedure [22]. Typically, cross-entropy is used as the objective and the optimization is done through stochastic gradient descent (SGD). For any given objective, the important quantity to calculate is its gradient with respect to the activations at the output layer. The gradients for all the parameters of the network can be derived from this one quantity based on the back-propagation procedure.

2.1. Cross-Entropy

For multi-class classification, it is common to use the negative log posterior as the objective:

$$\mathcal{F}_{CE} = -\sum_{u=1}^{U}\sum_{t=1}^{T_u} \log y_{ut}(s_{ut}), \quad (3)$$

where s_{ut} is the reference state label at time t for utterance u. This is also the expected cross-entropy between the distribution represented by the reference labels and the predicted distribution $y(s)$. The necessary gradient is:

$$\frac{\partial \mathcal{F}_{CE}}{\partial a_{ut}(s)} = -\frac{\partial \log y_{ut}(s_{ut})}{\partial a_{ut}(s)} = y_{ut}(s) - \delta_{s;s_{ut}}, \quad (4)$$

where $\delta_{s;s_{ut}}$ is the Kronecker delta function. Minimizing the cross-entropy is the same as maximizing the mutual information between $y(s)$ and $\delta_{s;s_{ut}}$ computed at the frame-level.

2.2. MMI

The MMI criterion used in ASR [9] is the mutual information between the distributions of the observation and word *sequences*. With $\mathbf{O}_u = \{\mathbf{o}_{u1}, \ldots, \mathbf{o}_{uT_u}\}$ as the sequence of all observations, and W_u as the word-sequence in the reference for utterance u, the MMI criterion is:

$$\mathcal{F}_{MMI} = \sum_u \log \frac{p(\mathbf{O}_u|S_u)^\kappa P(W_u)}{\sum_W p(\mathbf{O}_u|S)^\kappa P(W)}, \quad (5)$$

where $S_u = \{s_{u1}, \ldots, s_{uT_u}\}$ is the sequence of states corresponding to W_u; and κ is the acoustic scaling factor. The sum in the denominator is taken over all word sequences in the decoded speech lattice for utterance u. Differentiating (5) w.r.t. the log-likelihood $\log p(\mathbf{o}_{ut}|r)$ for state r, we get:

$$\begin{aligned}\frac{\partial \mathcal{F}_{MMI}}{\partial \log p(\mathbf{o}_{ut}|r)} &= \kappa\delta_{r;s_{ut}} - \frac{\kappa \sum_{W:s_t=r} p(\mathbf{O}_u|S)^\kappa P(W)}{\sum_W p(\mathbf{O}_u|S)^\kappa P(W)}, \\ &= \kappa(\delta_{r;s_{ut}} - \gamma_{ut}^{DEN}(r)), \quad (6)\end{aligned}$$

where $\gamma_{ut}^{DEN}(r)$ is the posterior probability of being in state r at time t, computed over the denominator lattices for utterance u. The required gradient w.r.t. the activations is obtained as:

$$\begin{aligned}\frac{\partial \mathcal{F}_{MMI}}{\partial a_{ut}(s)} &= \sum_r \frac{\partial \mathcal{F}_{MMI}}{\partial \log p(\mathbf{o}_{ut}|r)} \frac{\partial \log p(\mathbf{o}_{ut}|r)}{\partial a_{ut}(s)}, \\ &= \kappa(\delta_{s;s_{ut}} - \gamma_{ut}^{DEN}(s)). \quad (7)\end{aligned}$$

Note that in this work we have assumed that the reference state labels are obtained through a forced alignment of the acoustics with the word transcript. More generally, one may use forward-backward over the word reference to obtain the numerator occupancies $\gamma_{ut}^{NUM}(s)$ instead of using $\delta_{s;s_{ut}}$ in equations (4) and (7).

2.3. MPE/sMBR

While minimizing $\mathcal{F}_{CE}$ minimizes expected frame-error, maximizing $\mathcal{F}_{MMI}$ minimizes expected sentence error. The MBR family of objectives are explicitly designed to minimize the expected error corresponding to different granularity of labels [14]:

$$\mathcal{F}_{MBR} = \sum_u \frac{\sum_W p(\mathbf{O}_u|S)^\kappa P(W) A(W, W_u)}{\sum_{W'} p(\mathbf{O}_u|S)^\kappa P(W')}, \quad (8)$$

where $A(W, W_u)$ is the raw accuracy, that is, the number of correct phone labels (for MPE) or state labels (for sMBR) corresponding to the word sequence W with respect to that corresponding to the reference W_u. Differentiating (8) w.r.t. $\log p(\mathbf{o}_{ut}|r)$, we get:

$$\begin{aligned}\frac{\partial \mathcal{F}_{MBR}}{\partial \log p(\mathbf{o}_{ut}|r)} &= \kappa\, \gamma_{ut}^{DEN}(r) \left\{\bar{A}_u(s_t = r) - \bar{A}_u\right\}, \\ &= \kappa\, \gamma_{ut}^{MBR}(r),\end{aligned}$$

where $\bar{A}_u(s_t = r)$ is the average accuracy of all paths in the lattice for utterance u that pass through state r at time t; $\bar{A}_u$ is the average accuracy of all paths in the lattice; and $\gamma_{ut}^{MBR}(r)$ is the MBR "posterior" as defined in [12]. Like before, we get:

$$\frac{\partial \mathcal{F}_{MBR}}{\partial a_{ut}(s)} = \kappa \gamma_{ut}^{MBR}(s). \quad (9)$$

2.4. Boosted MMI

In boosted MMI [11], the MMI objective 5 is modified to boost the likelihood of paths that contain more errors:

$$\mathcal{F}_{BMMI} = \sum_u \log \frac{p(\mathbf{O}_u|S_u)^\kappa P(W_u)}{\sum_W p(\mathbf{O}_u|S)^\kappa P(W)\, e^{-b\, A(W,W_u)}}, \quad (10)$$

where b is the boosting factor. The BMMI criterion may also be interpreted as incorporating a margin term in the MMI objective [23]. The gradient computation is identical to that of MMI (eq. (7)), with the effect of the boosting showing up in the $\gamma_{ut}^{DEN}(s)$.

3. Experimental setup

In this paper, we report experiments on the 300 hour Switchboard conversational telephone speech task. Specifically, we use Switchboard-1 Release 2 (LDC97S62) as the training set, together with the Mississippi State transcripts[2] and the 30K-word lexicon released with those transcripts. The lexicon contains pronunciations for all words and word fragments in the training data. We use the Hub5 '00 (LDC2002S09) data as the development set and Hub5 '01 (LDC2002S13) data as a separate test set. A trigram language model (LM) is trained on 3M words of the training transcripts, which is then interpolated with another trigram LM trained on 11M words of the Fisher English Part 1 transcripts (LDC2004T19). The LMs are trained

[2]Available from: `http://www.isip.piconepress.com/`

Table 1: Results (% WER) for the baseline GMM-HMM systems on the subsets of the Hub5 '00 evaluation set.

System	Hours	SWB	CHE	Total
ML SAT GMM	300	21.2	36.4	28.8
BMMI SAT GMM	300	18.6	33.0	25.8
ML SAT GMM	110	23.8	38.6	31.2
BMMI SAT GMM	110	21.0	35.6	28.3

using interpolated Kneser-Ney smoothing and the interpolated LM has 950K trigrams and 1064K bigrams.

The acoustic models (both GMM-HMM and DNNs) are trained on features that are obtained by splicing together 7 frames (3 on each side of the current frame) of 13-dimensional MFCCs (C0-C12) and projecting down to 40 dimensions using linear discriminant analysis (LDA). The MFCCs are normalized to have zero mean per speaker. We also use a single semi-tied covariance (STC) transform [24] on the features obtained using LDA. The combined features are referred to as LDA+STC. Moreover, speaker adaptive training (SAT) is done using a single feature-space maximum likelihood linear regression (FMLLR) transform estimated per speaker. We select the first 100K utterances from the training data to create a second smaller training set with 110 hours of speech, in order to achieve faster turnaround times for the different tuning experiments.

3.1. Baseline GMM-HMM systems

The baseline GMM-HMM systems are trained on the LDA+STC+FMLLR features described above. The models trained on the full 300 hour training set contain 8859 tied triphone states and 200K Gaussians. In Table 1, we compare the results of the maximum likelihood (ML) trained models with those trained using BMMI with a boosting factor $b = 0.1$ (cf. equation (10)). It is worth pointing out that the Hub5 '00 data contain 20 conversations from Switchboard (SWBD) and 20 conversations from CallHome English (CHE). The CallHome data tends to be harder to recognize, partly due to a greater prevalence of foreign-accented speech. Here, we present results on both of these subsets as well as the complete Hub5 '00 evaluation set. Only the results in the SWB column should be compared with the Hub5 '00 results presented in [5] and [20]. The models trained on the 110 hour training set contain 4234 tied triphone states and 90K Gaussians, the results for which are similarly presented in Table 1. For either of the training conditions, the leaves of the phonetic decision tree used for the GMM-HMM system correspond to the output units of the respective DNNs.

3.2. DNNs trained using cross-entropy

The DNNs are trained on the same LDA+STC+FMLLR features as the GMM-HMM baselines, except that the features are globally normalized to have zero mean and unit variance. The FMLLR transforms are the same as those estimated for the GMM-HMM system during training and testing. The network trained on the full 300 hour training set has 7 layers (that is, 6 hidden layers), where each hidden layer has 2048 neurons; the DNN has 8859 output units. The input to the network is an 11 frame (5 frames on each side of the current frame) context window of the 40 dimensional features. This DNN[3] is initialized with stacked restricted Boltzmann machines (RBMs) that are pretrained in a greedy layerwise fashion [8]. The Gaussian-Bernoulli RBM is trained with an initial learning rate of 0.01 and the Bernoulli-Bernoulli RBMs with a rate of 0.4. The initial RBM weights are randomly drawn from a Gaussian $\mathcal{N}(0, 0.01)$; the hidden biases of Bernoulli units as well as the visible biases of the Gaussian units are initialized to zero, while the visible biases of the Bernoulli units are initialized as $b_v = \log(p/1 - p)$, where p is the mean output of a Bernoulli unit from previous layer. During pretraining, the momentum m is linearly increased from 0.5 to 0.9 on the initial 50 hours of data, which is accompanied by a rescaling of the learning rate using $1 - m$. Also the L2 regularization is applied to the weights, with a penalty factor of 0.0002.

[3] We do not use different names (e.g. deep belief networks) depending on how the networks are initialized.

Table 2: Results (% WER) for the DNN systems on the subsets of the Hub5 '00 evaluation set. The DNNs are trained on LDA+STC+FMLLR features using the cross-entropy criterion.

System	Init	Hours	SWB	CHE	Total
DNN 7 layers	RBM	300	14.2	25.7	20.0
DNN 5 layers	Rand	110	17.1	29.6	23.4

The DNN trained on the smaller training set (110 hours) has 5 layers, where each hidden layer has 1200 neurons; the network has 4234 output units. This network is randomly initialized, with the weights drawn from $\mathcal{N}(0, 0.01)$ and the biases initialized uniformly at random from $\mathcal{U}(-4.1, -3.9)$. A 9 frame context window ($\pm$ 4 frames) is used at input, followed by a second LDA transform that keeps 350 dimensions out of the 360 and whose output is globally normalized to zero mean and unit variance. The second LDA was initially used since it gave a slight improvement over a 5 layer network, and it was faster to train than a 6 layer network. However, it is not important to consider this detail for the overall message of this paper.

The utterances and frames are presented in a randomized order while training both of these networks using SGD to minimize the cross-entropy between the labels and network output. The SGD uses minibatches of 256 frames, and an exponentially decaying schedule that starts with an initial learning rate of 0.008 and halves the rate when the improvement in frame accuracy on a cross-validation set between two successive epochs falls below 0.5%. The optimization terminates when the frame accuracy increases by less than 0.1%. Cross-validation is done on a set of 4000 utterances that are held out from the training data. The speed of training is accelerated by running on general-purpose graphics processing units (GPGPUs). The recognition results with these two DNNs are presented in Table 2.

3.3. Sequence-discriminative training of DNNs

Just like with GMM-HMM systems, sequence-discriminative training of DNNs start from a set of alignments and lattices that are generated by decoding the training data with a unigram LM. For each training condition, the alignments and lattices are generated using the corresponding DNN trained using cross-entropy. The cross-entropy trained models are also used as the starting point for the sequence-discriminative training. As with CE training, the posterior probability computation using the DNN and the backpropagation are done on a GPU, while the lattice-based computations run on a CPU. It is possible to speed-up the training by using a distributed algorithm [20, 25]. However, this has not been done in this initial implementation.

Through some initial benchmarking experiments with MMI as the objective function, we found 1e-5 to be a suitable learning

Table 3: Results (% WER) of the DNNs trained on the full 300 hour training set using different criteria.

System	Hub5 '00 SWB	CHE	Total	Hub5 '01 SWB	SWB2P3	SWB-Cell	Total
GMM BMMI	18.6	33.0	25.8	18.9	24.5	30.1	24.6
DNN CE	14.2	25.7	20.0	14.5	19.0	25.3	19.8
DNN MMI	12.9	24.6	18.8	13.3	17.8	23.7	18.4
DNN sMBR	12.6	24.1	18.4	13.0	17.7	22.9	18.0
DNN MPE	12.9	24.1	18.5	13.2	17.7	23.4	18.2
DNN BMMI	12.9	24.5	18.7	13.2	17.8	23.5	18.3

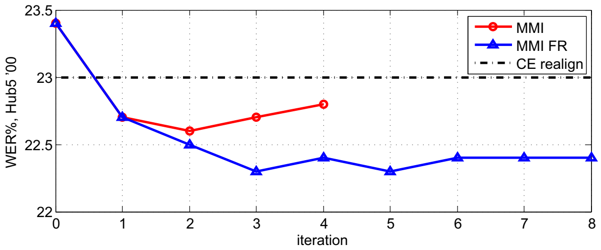

Figure 1: Hub5 '00 results: DNNs trained with MMI on 110h set, with and without frame rejection (FR).

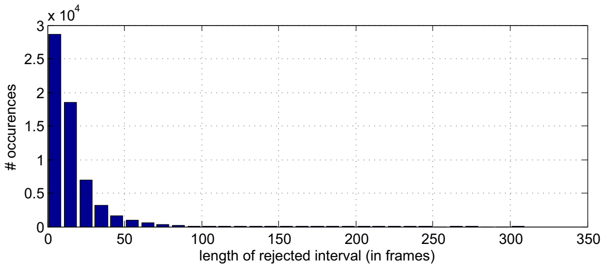

Figure 2: Histogram of lengths of rejected frame intervals.

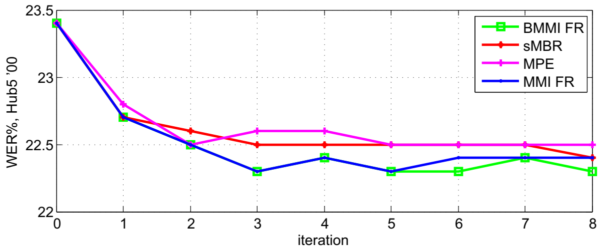

Figure 3: Hub5 '00: DNNs trained on 110h set, various criteria.

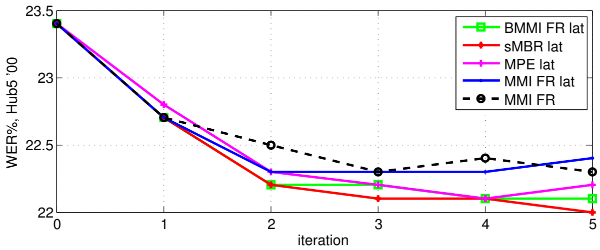

Figure 4: Hub5 '00 results: Lattice regeneration after 1st epoch (indicated by "`lat`" suffix), trained on the 110h set.

rate[4] and that an exponentially decaying learning rate provided no gains. Figure 1 shows the results with MMI trained DNNs on Hub5 '00. The horizontal line (`CE realign`) shows the results with CE training when starting with alignments from a CE trained DNN instead of the alignments from a GMM system. This accounts for about half of the improvements from MMI. We find the MMI objective to overfit after 2 iterations. A detailed analysis revealed somewhat anomalous objective and gradient values for utterances where the reference hypothesis is missing from the lattice. This may be caused by search errors or by a poor match of the acoustics to the model or even by errors in the reference transcription. However, only in the first of these cases, that is when there are search errors on the training data, it is reasonable to explicitly add the reference to the lattice.

A closer look at the number of frames in intervals where the reference is missing from the lattice (Figure 2) reveals that most of them are short segments. In fact, 78% of such frames lie in intervals shorter than 50 frames (i.e. 0.5 seconds). While the errors are mostly local, these frames may disproportionately impact the training since $\gamma_{ut}^{DEN}(s) = 0$ for them and hence the computed gradients are larger. As a result, we decided to remove such frames from the gradient computation, which reduces the amount of training data by 2.5%. The results in Figure 1 show that this frame rejection (FR) heuristic leads to more stable learning. Nearly all of the reduction in errors is on the CallHome part, which is more mismatched to the training data.

Next, comparing the different sequence-discriminative criteria in Figure 3, we do not find a big difference between them. A learning rate of 1e-5 was also found to work well for these other criteria. In Figure 4, we compare the results when the lattices are regenrated after the first epoch. We see that regenerating lattices provide a small gain. However, this is computationally expensive and regenerating lattices after the second epoch did not produce any further gains. Finally, Table 3 summarizes the results of the different systems trained on the entire 300 hour training set. The results are presented on both the development set (Hub5 '00) and the test set (Hub5 '01) and their respective subsets.

4. Conclusions

We have presented experiments with DNN-HMM hybrid systems trained using frame-based cross-entropy and different sequence-discriminative criteria on the 300 hour Switchboard conversational telephone speech task. We achieved state-of-the-art results on this task. While the gain from sequence-discriminative training is found to be 8-9% on average, it varies depending on how matched the test set is to the train set. The system building scripts and the neural network training code are released as part of the free and open-source Kaldi toolkit, making it possible for the wider speech recognition research community to use these state-of-the-art techniques in their work.

[4] In our implementation, the gradients are not scaled by the acoustic scale κ, but its effect is subsumed in the learning rate. So, with $\kappa = 0.1$, the effective learning rate is 1e-4.

5. References

[1] H. Bourlard and N. Morgan, *Connectionist Speech Recognition—A Hybrid Approach*. Kluwer Academic, 1994.

[2] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco, "Connectionist probability estimators in HMM speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 1, pp. 161–174, 1994.

[3] A. J. Robinson, "An application of recurrent nets to phone probability estimation," *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 298–305, 1994.

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

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

[6] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," in *Proc. INTERSPEECH*, September 2012.

[7] 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. IEEE ASRU*, December 2011, pp. 30–35.

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

[9] 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. IEEE ICASSP*, vol. 1, April 1986, pp. 49–52.

[10] V. Valtchev, J. Odell, P. Woodland, and S. Young, "MMIE training of large vocabulary recognition systems," *Speech Communication*, vol. 22, no. 4, pp. 303–314, September 1997.

[11] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *Proc. IEEE ICASSP*, 2008, pp. 4057–4060.

[12] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, University of Cambridge, Cambridge, UK, 2003.

[13] 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*, vol. 2, October 2000, pp. 887–890.

[14] M. Gibson and T. Hain, "Hypothesis spaces for minimum Bayes risk training in large vocabulary speech recognition," in *Proc. INTERSPEECH*, September 2006, pp. 2406–2409.

[15] D. Povey and B. Kingsbury, "Evaluation of proposed modifications to MPE for large scale discriminative training," in *Proc. IEEE ICASSP*, vol. 4, April 2007, pp. IV–321–IV–324.

[16] J. S. Bridle and L. Dodd, "An Alphanet approach to optimising input transformations for continuous speech recognition," in *Proc. IEEE ICASSP*, vol. 1, April 1991, pp. 277–280.

[17] A. Krogh and S. K. Riis, "Hidden neural networks," *Neural Computation*, vol. 11, no. 2, pp. 541–563, February 1999.

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

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

[20] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum Bayes risk training of deep neural network acoustic models using distributed Hessian-free optimization," in *Proc. INTERSPEECH*, September 2012.

[21] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *IEEE ASRU*, December 2011.

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, October 1986.

[23] G. Heigold, T. Deselaers, R. Schlüter, and H. Ney, "Modified MMI/MPE: A direct evaluation of the margin in speech recognition," in *Proc. ICML*, 2008, pp. 384–391.

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

[25] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng, "Large scale distributed deep networks," in *Advances in Neural Information Processing Systems*. MIT Press, 2012, pp. 1232–1240.