ERROR BACK PROPAGATION FOR SEQUENCE TRAINING OF CONTEXT-DEPENDENT DEEP NETWORKS FOR CONVERSATIONAL SPEECH TRANSCRIPTION

Hang Su[1,2], Gang Li[1], Dong Yu[3], and Frank Seide[1]

[1] Microsoft Research Asia, 5 Danling Street, Haidian District, Beijing 100080, P.R.C.
[2] Department of Electronic Engineering, Tsinghua University, 10084 Beijing, P.R.C.
[3] Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
{fseide,ganl,dongyu}@microsoft.com, suhang3240@gmail.com

ABSTRACT

We investigate back-propagation based sequence training of Context-Dependent Deep-Neural-Network HMMs, or CD-DNN-HMMs, for conversational speech transcription.

Theoretically, sequence training integrates with back-propagation in a straight-forward manner. However, we find that to get reasonable results, heuristics are needed that point to a problem with lattice sparseness: The model must be adjusted to the updated numerator lattices by additional iterations of frame-based cross-entropy (CE) training; and to avoid distortions from "runaway" models, we can either add artificial silence arcs to the denominator lattices, or smooth the sequence objective with the frame-based one (F-smoothing).

With the 309h Switchboard training set, the MMI objective achieves a relative word-error rate reduction of 11–15% over CE for matched test sets, and 10–17% for mismatched ones. This includes gains of 4–7% from realigned CE iterations. The BMMI and sMBR objectives gain less. With 2000h of data, gains are 2–9% after realigned CE iterations. Using GPGPUs, MMI is about 70% slower than CE training.

1. INTRODUCTION AND RELATED WORK

In this paper, we investigate the use of back-propagation for word-lattice based *sequence training* of context-dependent deep-neural-network HMMs, or CD-DNN-HMM. CD-DNN-HMMs [1, 2] achieve significant accuracy improvements over discriminatively trained GMM-HMMs, such as 16% on a business-search task [1, 2], and up to one-third on the Switchboard phone-call transcription benchmark [3], These CD-DNN-HMMs were trained with *error back-propagation* [4] using the frame-based *cross-entropy* (CE) objective.

Sequence training seeks to better match the MAP decision rule of LVCSR decoding by considering sequence constraints from HMMs, dictionary, and the language model. For GMM-HMMs, sequence training (there called "discriminative training") often yields relative error reductions around 10% over maximum-likelihood training (e.g., see [5]).

MMI sequence training with DNNs was first reported in [6], where a 5% relative gain over CE on the TIMIT corpus was achieved. For lattice-based DNN sequence training, [7] reports an excellent 17% gain on Switchboard from using the sMBR criterion with the second-order Hessian-free method [8], for a 5-hidden layer CD-DNN-HMM. Using back-propagation instead, [7] reports a 14% gain on a 50h set. [9] reports a 1% relative gain with MMI on voice search and Youtube videos for a 4-hidden layer CD-DNN-HMM trained on much larger sets of 6000 and 1400h, respectively. For sequence training of single-hidden-layer networks, reported gains include 24% using factored triphones [10], and up to 19% for a CD-ANN-HMM [11].

This paper investigates in more depth the use of back-propagation for lattice-based sequence training on a corpus of a few hundred hours, specifically the 309h Switchboard corpus. To use popular objective functions like MMI or sMBR with back-propagation, one "only" needs to cut in an additional processing step that replaces the top layer's error signal by expressions very similar to extended-Baum-Welch equations for GMMs, while making full use of the existing back-propagation machinery, as laid out by [11].

In practice, however, this proves a beast—several heuristics are needed to get reasonable results. After introducing the CD-DNN-HMM in section 2, section 3 discusses possible reasons and remedies to address the problem to some degree. Section 4 describes our efficient GPGPU-based implementation, and section 5 presents experimental results.

2. THE CONTEXT-DEPENDENT DEEP-NEURAL-NETWORK HMM

2.1. Context-Dependent Deep Neural Network

A deep neural network (DNN) is a conventional multi-layer perceptron (MLP [12]) with many layers, where training is commonly initialized by a pretraining algorithm [2]. A CD-DNN-HMM models the posterior probability $P(s|o)$ of a tied triphone state, or senone s [11, 1], given an observation vector o. As a brief recap, it is a stack of $(L+1)$ layers of log-linear models of the form $P(h^\ell|v^\ell) = 1/Z^\ell \exp((W^\ell)^T v^\ell + a^\ell)$ with layer-type specific partition functions Z^ℓ, weight matrices W^ℓ and bias vectors a^ℓ (the model parameters to train), and v^ℓ and h^ℓ denoting the input and output of each layer.

For hidden layers, the components of h^ℓ are assumed binary and conditionally independent, such that $P(h^\ell|v^\ell)$ has the form of a component-wise sigmoid. With the "mean-field approximation" [14], the expected value of h^ℓ is used as the input to the next layer: $v^{\ell+1} \stackrel{\text{def}}{=} E\{\mathbf{h}^\ell|v^\ell\}$. For the output layer, h^L is a unit vector with the position of the 1 denoting the senone s: $P(s|o) = P(h^L_s = 1|v^L)$. This constraint gives rise to the form of softmax. For details, see, e.g., [13].

Following [15], for decoding and lattice generation, the senone posteriors are converted into the HMM's emission likelihoods by dividing by the senone priors $P(s)$:

$$\log p(o|s) \quad = \quad \log P(s|o) \; - \; \log P(s) + \log p(o) \qquad (1)$$

where the observation vectors o are acoustic feature vectors augmented with neighbor frames. $p(o)$ is unknown but can be ignored as it cancels out in best-path decisions and word-posterior computation. Likewise, we can ignore Z^L in $P(s|o)$, which is a significant time-saver since it allows for on-demand computation in decoding.

CD-DNN-HMMs can be trained with the stochastic-gradient method *error back-propagation* (BP) [4] (typically after initialization by pre-training [16, 13]). Of the gradient terms, whose detailed formulas can be found e.g. in [13], the term of relevance here is the top layer's "error signal:"

$$e_s(r,t) \; = \; \frac{\partial \mathcal{F}}{\partial \log P(s|o^r(t))}$$

where $\mathcal{F}$ is short for $\mathcal{F}(W^0, a^0, ..., W^L, a^L)$, the objective function to maximize over all training utterances' frames $O^r = (o^r(1), o^r(2), ..., o^r(t), ...)$, and r is the utterance index.

2.2. Frame-Discriminative Training

To train CD-DNN-HMMs, one commonly maximizes the total log posterior probability over training frames $o^r(t)$ given ground-truth senone labels $\hat{s}^r(t)$ [15]. This is also known as the *cross-entropy* (CE) criterion (with Kronecker delta δ):

$$\mathcal{F}^{\mathrm{CE}} \quad = \quad \sum_r \sum_t \log P(\hat{s}^r(t)|o^r(t)), \qquad (2)$$

$$e_s^{\mathrm{CE}}(r,t) \quad = \quad \delta_{s,\hat{s}^r(t)} - P(s|o^r(t)) \qquad (3)$$

2.3. Sequence Training

Sequence training incorporates HMM, lexical, and some language-model constraints of the actual MAP decision rule. Popular sequence objectives, known from GMM systems, are maximum mutual information[1] (MMI), boosted MMI (BMMI, [17]), and minimum Bayes risk (MBR [18, 19]):

$$\mathcal{F}^{\mathrm{MMI}} \quad = \quad \sum_r \log P(\hat{S}^r|O^r) \qquad (4)$$

$$\mathcal{F}^{\mathrm{BMMI}} \quad = \quad \sum_r \log \frac{P(\hat{S}^r|O^r) \cdot e^{-b \cdot \mathcal{A}^r(\hat{S}^r)}}{\sum_S P(S|O^r) \cdot e^{-b \cdot \mathcal{A}^r(S)}} \qquad (5)$$

$$\mathcal{F}^{\mathrm{MBR}} \quad = \quad \sum_r \sum_S P(S|O^r) \cdot \mathcal{A}^r(S) \qquad (6)$$

with path $S = (s(1), s(2), ..., s(t), ...)$ denoting a senone sequence, $\hat{S}^r$ specifically being the ground-truth senone labels of utterance r, accuracy function $\mathcal{A}^r(S)$, BMMI weight b, and path posteriors $P(S|O^r)$ given the current model:

$$P(S|O^r) \quad = \quad \frac{p^\kappa(O^r|S)P(S)}{\sum_{S'} p^\kappa(O^r|S')P(S')} \qquad (7)$$

[1]Technically, the CE criterion is also MMI, applied to individual frames.

The acoustic likelihoods $p(O^r|S) = \prod_t p(o^r(t)|s(t))$ are computed using Eq. (1). The $P(S)$ are path priors that combine HMM transitions, lexicon, and LM. κ is the acoustic weight. Our specific accuracy function $\mathcal{A}^r(S)$ is on senone level (sMBR [23]), it counts correct senone labels in a path S against ground truth $\hat{S}^r$. The corresponding error signals are:

$$e_s^{\mathrm{MMI}}(r,t) \; = \; \delta_{s,\hat{s}(t)} - \gamma_s^r(t) \qquad (8)$$

$$e_s^{\mathrm{MBR}}(r,t) \; = \; \kappa \; \gamma_s^r(t) \; \Big[E\{\mathcal{A}^r(S)|s(t)=s\} - E\{\mathcal{A}^r(S)\} \Big]$$

$$\text{with} \qquad \gamma_s^r(t) \quad = \quad \sum\nolimits_S \delta_{s(t),s} \; P(S|O^r)$$

$$\mathcal{A}^r(S) \quad = \quad \sum_t \delta_{s(t),\hat{s}^r(t)}$$

$$E\{\mathcal{A}^r(S)|s(t)=s\} \quad = \quad \frac{\sum_S \delta_{s(t),s} \; P(S|O^r) \cdot \mathcal{A}^r(S)}{\sum_S \delta_{s(t),s} \; P(S|O^r)}$$

and $E\{\mathcal{A}^r(S)\}$ likewise. e^{BMMI} is the same as e^{MMI} except that $\gamma_s^r(t)$ is modified analogously to the change from Eq. (4) to (5). These error signals are computed efficiently using forward-backward procedures (e.g. [5]). Thus, sequence-training BP can reuse the existing CE BP machinery, just with a modified computation of the error signal [11].

3. BP SEQUENCE TRAINING IN PRACTICE

So far the theory. In practice, however, we find that back-propagation and lattice-based training are not made for each other: WER initially improves as expected, but after only tens of hours of data, it begins to decay, along with an increase of deletions. The objective, however, keeps improving. We investigated a number of potential causes—all, one way or another, related to the unavoidable sparseness of word lattices: Even the fattest lattices that we could practically generate reference only about 300 senones per frame, out of 9304.

3.1. "Run-away" Silence Model

The growing deletions turn out to be speech-silence substitutions, accompanied by a growth of the log-likelihoods of silence (e.g. about 10%) that we don't see for speech scores.

For sMBR, this issue disappears when using the known heuristic of counting silence frames as "incorrect" in $\mathcal{A}^r$ [5]. For (B)MMI, a similar effect can be achieved by setting $e_s(t)$ to zero for all silence states and all silence frames. However, we found that a slightly better approach is to augment the lattice with artificial silence arcs (one for each start/end node pair connected by a word arc, with an appropriate entering probability, and preventing redundant silence paths by forbidding silence-to-silence transitions). The goal is to make "run-away" silence visible to the objective function.

3.2. Smoothing for "Run-away" Speech Models

This cannot be easily extended to non-silence. An alternative way of making "run-away" models visible is to use a form of H-criterion [20] that interpolates the sequence and frame objectives: $\mathcal{F}^{\mathrm{FSMMI}} = (1-H)\mathcal{F}^{\mathrm{CE}} + H\mathcal{F}^{\mathrm{MMI}}$. This is inspired by I-smoothing [5] and similar regularization approaches for

Table 1. *Results for MMI with BP for 309h and 2000h Switchboard models with 7 hidden layers (and GMM baselines) across seven test sets. WERs in percent, relative change to reference ("ref") in parentheses.*

acoustic model & training	Hub5'00 SWB	RT03S FSH	RT03S SW	voicemails MS	voicemails LDC	tele-conferences	executive presentation
GMM-ML (309h)	26.1 (ref)	30.2 (ref)	40.9 (ref)	45.0 (ref)	33.5 (ref)	35.2 (ref)	18.2 (ref)
+ BMMI	23.6 (-10%)	27.4 (-9%)	37.6 (-8%)	42.4 (-6%)	30.8 (-8%)	33.9 (-4%)	18.4 (+1%)
DNN-CE (309h)	16.2 (ref)	19.6 (ref)	28.4 (ref)	34.5 (ref)	24.6 (ref)	26.5 (ref)	9.6 (ref)
+ CE update with numerator lattices	15.6 (-4%)	18.6 (-5%)	27.4 (-4%)	32.8 (-5%)	23.0 (-7%)	24.6 (-7%)	9.8 (+2%)
+ MMI (F/S ratio 1:9)	13.7 (-15%)	17.1 (-13%)	25.4 (-11%)	29.8 (-14%)	20.5 (-17%)	23.3 (-12%)	8.6 (-10%)
DNN-CE (2000h) + CE update	14.6 (ref)	16.0 (ref)	22.1 (ref)	32.1 (ref)	25.6 (ref)	22.5 (ref)	7.7 (ref)
+ MMI (F/S ratio 1:4)	13.3 (-9%)	14.6 (-9%)	20.2 (-9%)	31.4 (-2%)	24.5 (-4%)	21.4 (-4%)	7.4 (-4%)

adaptation [21, 22]. We want to call it "frame smoothing," or *F-smoothing*. We find frame/sequence ratios of 1:4 to 1:10 highly effective. We also experimented with L2 regularization, but did not find it to help.

3.3. Effect of Realignment

Sequence training operates on numerator and denominator lattices generated with an initial CE model. In our system, "numerator lattice" is just another term for the ground-truth state alignment. In [1, 3], we showed that re-alignment followed by further CE iterations leads to notable WER gains (0.6 points for SWBD). I.e., the initial CE model is suboptimal for the lattices generated with it, so a sequence training started with that model (as in [11, 7]) will have to rectify this mismatch (in addition to the change of training objective) while being constrained by sparse lattices. To remedy this, we propose to first perform further *CE* iterations using the updated numerator lattices before entering sequence training.

4. EFFICIENT IMPLEMENTATION WITH GPGPUS

[7] did not "test stochastic gradient descent sMBR training [for Switchboard] because the experiment would have taken too long." Indeed, our initial CPU-side implementation increased runtime 12-fold. However, we achieve significant speed-ups through parallelized execution on a GPGPU (NVidia Tesla). Each arc is processed as a separate CUDA thread. This is straight-forward for acoustic-score computation. However, the lattice-level forward-backward processing must be decomposed into sequential, dependency-free CUDA launches. E.g., for a 7.5-second lattice with 211,846 arcs and 6974 nodes, we have 106 dependency-free node regions (=launches) at an average of 1999 arcs (=threads per launch).

Lattice forward/backward and error-signal accumulation require atomic summation of probabilities represented as logarithms. We emulate this through CUDA's atomic "compare-and-swap" instruction. It is critical to shuffle operations into a random-like order, in order to reduce target-operand clashes.

With this, reading the randomized lattice chunks from the network now takes up nearly 40% of the total runtime. Using a parallel read-ahead thread reduces that to zero.

Table 2 summarizes the runtimes. Compared to CE training, the overall runtime increases by about 70%, with fat lattices of nearly 500 arcs per frame.

Table 2. *MMI vs. CE training runtime for 24h of data.*

	CE (GPGPU)	+MMI (CPU)	vs. +MMI (GPGPU)	+parallel read-ahead
training time	70 min	14.5h	2.75h	2h
increase	-	12.4×	2.4×	+71%

5. EXPERIMENTAL RESULTS

We evaluate mini-batch BP sequence training on speech-to-text transcription using the 309-hour Switchboard-I training set [24]. The GMM-HMM baseline has 40 mixtures per state and is trained with maximum likelihood (ML) and BMMI. The CD-DNN-HMM has 7 hidden layers of dimension 2048. The number of senones is 9304 for both. The data for system development is the 1831-segment SWB part of the NIST 2000 Hub5 evaluation set. All recognition is speaker independent.

5.1. Core Results

Table 1 shows the main results for several test sets. The row labeled "DNN-CE" shows the CE-trained Switchboard baseline. This model was then used to generate numerator and denominator lattices. As mentioned in Section 3.3, we need to adjust the model to the updated numerators by 7 additional CE data passes (computed parallel to lattice generation). Row "+CE update with numerator lattices" shows that this yields gains of 4% or more for all but one set.

The row labeled "+MMI" shows WERs obtained with the F-smoothed MMI objective (Sec. 3.2) at a frame/sequence ratio of 1:9. We used a fixed learning rate of 1/128,000 per frame. On the well-matched Hub5 set, WER is reduced from 16.2% to 13.7%, a total 15% relative reduction. A quarter of this is due to the CE update.[2] For comparison, [7] reports a 17% gain using Hessian-free optimization for a very similar setup (without explicit CE update, though). Unlike [7], we did not see gains from tuning LM weight or word penalty.

Lattice generation used the testing dictionary; the training set's out-of-vocabulary rate is 16%. If we use the training dictionary instead, the Hub5'00 WER increases by about 0.2%

[2]The CE WERs differ from our earlier reports [3, 13] due to a bug in error counting: Our spelling of the interjection word "um-hmm" was missing in the NIST error-counting tool's text-normalization table. The 16.2% for DNN-CE (309h) on Hub5'00 SWB had formerly been reported as 17.1%.

Table 3. *Effect of objective function. Shown are WERs in %.*

WER	CE	MMI + F/S	BMMI + F/S	sMBR
Hub5'00-SWB	16.2	13.7 (-15%)	14.0 (-14%)	14.7 (-9%)
RT03S-FSH	19.6	17.1 (-13%)	17.4 (-11%)	17.8 (-9%)

Table 4. *Effect of extra CE update using numerator lattices.*

model M	CE training w/ alignment from M	MMI w/ lattices from M and initial model... ...CE	...CE×2
GMM	16.2 (ref) → CE	15.8 (-2%)	–
CE	15.6 (-4%) → CE×2	14.1 (-13%)	13.7 (-15%)
CE×2	15.7 (-3%)	–	13.5 (-17%)

points. The numerator lattices were merged into the denominator lattices arc-by-arc in both setups.

The other six test sets in Table 1 cover a range of accuracies and difficulty levels: the Spring'03 NIST rich transcription set (RT03S), two voicemail corpora, internal teleconferences, and an 17-minute rehearsal of a company executive's presentation [25]. Relative gains from MMI range from 10 to 17%, with 4 to 7% due to the CE update alone. Those gains are somewhat consistent with the GMM model's gains from BMMI on these sets (second row).

In all cases, the model was trained for 6 data passes. We observe a slight tendency for over-training: After 8 additional data passes, the WER for Hub5'00 SWB reduces further, from 13.7 to 13.2%, while the other sets get worse by similar margins. In Table 1, for each of the seven test sets, we picked the optimal number of MMI data passes by averaging the relative MMI gains for the respective *other* six test sets in the table, which worked out to 6 data passes in each case.

Finally, the last section of Table 1 shows results for 2000h of training data. The F/S ratio here had to be increased to 1:4 to avert decay. The 2000h model uses more senones (18k), which aggravates the sparseness effect. The 2000h CE baseline had already undergone several re-alignments in its training recipe. Considering that, the relative gains are nearly the same as for 309h for the matched Hub5 and RT03S sets, while only about half of that for the mismatched ones.

Table 3 compares the three sequence criteria for the dev set and RT03S (FSH portion). Unlike others (e.g. [11]), we do not seem to benefit from BMMI or sMBR compared to MMI.

5.2. Training: Realignment and Randomization

Table 4 shows the impact of the additional CE update after lattice generation (Sec. 3.3). The row labelled "CE" shows that when entering MMI training with the same CE model that was used to generate the lattices, we see a relative WER gain of 13%, compared to 15% if we start MMI with model "CE×2" that has undergone an additional CE update on the realigned numerator lattices. I.e., the CE model is suboptimal w.r.t. the lattices, and MMI BP fails to rectify that. Generating lattices with CE×2 (last row) increases the gain to 17%. The first table row shows that lattices generated with the seed GMM model are not suitable for DNN MMI training.

In all results in this paper, individual frames are presented to BP in random order for CE training; while in sequence training, randomization units are entire utterances. Table 5 shows (on a slightly different setup) that for CE training, utterance randomization degrades WER by 7%, which indicates opportunities for further gains by better randomization. The batch-based Hessian-free method [7, 8] has no such issue.

Table 5. *Impact of randomization on CE training.*

randomization unit:	frame	utterance
training-set frame accuracy	55.1%	52.0%
WER [%] (Hub5'00-SWB)	16.6	17.7 (+7%)

Table 6. *Experiments on WER decay. Shown are WERs in %.*

modeling technique	Number of data passes 1	2	3	4	5
DNN-CE	16.2				
+ CE update w/ numer. lattices	15.6				
+ MMI, plain	17.2	24.2			
+ no silence frame/state update	14.3	14.5	14.6	14.9	15.3
vs. + augment silence arcs	14.4	14.3	14.4	14.7	15.1
+ regenerated lattices	↪	14.3	14.3	14.2	14.3
vs. + F-smoothing (F/S ratio 1:9)	13.9	13.7	13.8	13.6	13.8

5.3. Experiments on WER Decay

Table 6 shows the WER decay behavior over the first five data passes for the heuristics described in Sec. 3. The row labeled "MMI, plain" shows that the plain MMI criterion of Eq. (4) without heuristics quickly drives to a drastic accuracy drop. The MMI objective kept improving, though. Reducing the learning rate can slow it but not solve it.

Either technique of Sec. 3.1, "no silence frame/ state update" or "augment silence arcs," substantially slows down the problem, confirming the "run-away" silence hypothesis. Silence-arc augmentation works slightly better. Regenerating the lattices after one data pass (next row) yet again slows down the effect, but not as effectively as F-smoothing, which eliminates the decay problem and finally leads to good results.

6. CONCLUSION

Back-propagation sequence training of CD-DNN-HMMs is attractive due to its mathematical simplicity. It can be efficiently implemented with GPGPUs (about 70% slower than cross-entropy), although it is overall slower and less scalable than second-order batch methods [7].

However, on the Switchboard corpus, we observe severe over-training like issues: After an initial significant gain, WERs begin to decay—while the objective keeps increasing. We have shown that this is largely due to the unavoidable sparseness of lattices: The vast majority of senones is unseen most of the time in the lattices due to pruning. Inappropriate increases of their scores remain mostly invisible to the objective function and do not get appropriately corrected—with increasing negative impact as the model moves away from the original model used to generate the lattices. We suspect that this is particularly harmful for stochastic gradient methods that rely on quick error feedback.

We developed three heuristics to work around the lattice-sparseness issue: separating the problem of initial lattice/model mismatch from the objective-function change by inserting a CE update on the numerator lattices before entering the sequence-training iterations; augmenting the lattices with artificial silence edges; and F-smoothing. The latter two aim at making "run-away" models visible to the objective function.

With these, we observe up to 17% relative WER gains, which is competitive with the best known results for 2nd-order methods. F-smoothed MMI is also effective when increasing the training data from 300 to 2000h hours of speech.

7. 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," 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., Special Issue on Deep Learning for Speech and Language Processing, 2011.

[3] F. Seide, G. Li, and D. Yu, "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks," Interspeech, 2011.

[4] D. Rumelhart, G. Hinton, and R. Williams, "Learning Representations By Back-Propagating Errors," Nature, vol. 323, Oct. 1986.

[5] D. Povey, "Discriminative Training for Large Vocabulary Speech Recognition," PhD thesis, Cambridge University, 2003.

[6] A.-R. Mohamed, D. Yu, and L. Deng, "Investigation of Full-Sequence Training of Deep Belief Networks for Speech Recognition," Interspeech, 2010.

[7] B. Kingsbury, T. Sainath, and H. Soltau, "Scalable Minimum Bayes Risk Training of Deep Neural Network Acoustic Models Using Distributed Hessian-free Optimization," Interspeech, 2012.

[8] J. Martens, "Deep learning via Hessian-free optimization," ICML, 2010.

[9] N. Jaitly P. Nguyen, A. Senior, and V. Vanhoucke, "Application of Pretrained Deep Neural Networks to Large Vocabulary Speech Recognition," Interspeech 2012.

[10] G. Wang, K. C. Sim, "Sequential Classification Criteria for NNs in Automatic Speech Recognition," Interspeech 2011

[11] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," ICASSP, 2009.

[12] F. Rosenblatt, "Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms", Spartan Books, Wash. DC, 1961.

[13] F. Seide, G. Li, X. Chen, and D. Yu, "Feature Engineering in Context-Dependent Deep Neural Networks for Conversational Speech Transcription," Proc. ASRU, Waikoloa Village, 2011.

[14] L. Saul *et al.*, "Mean Field Theory for Sigmoid Belief Networks", Journal: Computing Research Repository–CORR, pp. 61-76, 1996.

[15] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco, "Connectionist Probability Estimators in HMM Speech Recognition," IEEE Trans. Speech and Audio Proc., January 1994.

[16] G. Hinton, S. Osindero, and Y. Teh, "A Fast Learning Algorithm for Deep Belief Nets", Neural Computation, vol. 18, pp. 1527–1554, 2006.

[17] D. Povey *et al.*, "Boosted MMI for Model and Feature-Space Discriminative Training," ICASSP, 2008.

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

[19] M. Gibson and T. Hain, Hypothesis spaces for minimum Bayes risk training in large vocabulary speech recognition, in Proc. Interspeech, 2006.

[20] P. S. Gopalakrishnan, D. Kanevsky, A. Nádas, and D. Nahamoo, "An Inequality for Rational Functions with Applications to Some Statistical Estimation Problems," IEEE Trans. Information Theory, Vol. 37, pp. 107–113, 1991.

[21] D. Albesano, R. Gemello, P. Laface, F. Mana, and S. Scanzio, "Adaptation of Artificial Neural Networks Avoiding Catastrophic Forgetting," International Joint Conference on Neural Networks, 2006.

[22] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-Divergence Regularized Deep Neural Network Adaptation For Improved Large Vocabulary Speech Recognition," ICASSP 2013.

[23] J. Zheng and A. Stolcke, "Improved Discriminative Training Using Phone Lattices," Eurospeech, 2005.

[24] J. Godfrey and E. Holliman, "Switchboard-1 Release 2," Linguistic Data Consortium, Philadelphia, 1997.

[25] R. Rashid, "Microsoft Research shows a promising new breakthrough in speech translation technology," http://blogs.technet.com, Nov. 2012.

[26] J. Fiscus *et al.*, "2000 NIST Evaluation of Conversational Speech Recognition over the Telephone: English and Mandarin Performance Results," from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.27.5417.