INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Improved Neural Network Initialization by Grouping Context-Dependent Targets for Acoustic Modeling

Gakuto Kurata, Brian Kingsbury

IBM Watson

gakuto@jp.ibm.com, bedk@us.ibm.com

Abstract

Neural Network (NN) Acoustic Models (AMs) are usually trained using context-dependent Hidden Markov Model (CD-HMM) states as independent targets. For example, the CD-HMM states of A-b-2 (second variant of beginning state of A) and A-m-1 (first variant of middle state of A) both correspond to the phone A, and A-b-1 and A-b-2 both correspond to the Context-independent HMM (CI-HMM) state A-b, but this relationship is not explicitly modeled. We propose a method that treats some neurons in the final hidden layer just below the output layer as dedicated neurons for phones or CI-HMM states by initializing connections between the dedicated neurons and the corresponding CD-HMM outputs with stronger weights than to other outputs. We obtained 6.5% and 3.6% relative error reductions with a DNN AM and a CNN AM, respectively, on a 50-hour English broadcast news task and 4.6% reduction with a CNN AM on a 500-hour Japanese task, in all cases after Hessian-free sequence training. Our proposed method only changes the NN parameter initialization and requires no additional computation in NN training or speech recognition runtime.

Index Terms: speech recognition, deep neural network, convolutional neural network, context-dependent HMM state target, parameter initialization

1. Introduction

Neural Network based Acoustic Models (NN-AMs) have been showing significant improvement over the conventional Gaussian Mixture Model-AMs (GMM-AMs) in Large Vocabulary Continuous Speech Recognition (LVCSR) [1]. Various types of NN including Feed-forward Deep Neural Network (DNN) [2], Convolution Neural Network (CNN) [3], Recurrent Neural Network (RNN) [4], Long Short Term Memory (LSTM) [5] which is a specific type of RNN, and the novel combination of them [6] have been used for acoustic modeling. For decoding the speech, bottleneck feature-based system [7, 8], NN/Hidden Markov Model (HMM) hybrid system [9, 10], and recently proposed end-to-end speech recognition system [11, 12, 13, 14, 15] have been proposed. While the end-to-end speech recognition system using such as the Connectionist Temporal Classification (CTC) and the encoder-decoder approaches have been showing promising results recently, the NN/HMM hybrid system is still the most-widely used system. In the NN/HMM hybrid system, NN-AMs are used to estimate the posterior probabilities for acoustic observations and these probabilities are used for decoding by being combined with an HMM. In the 1990s, due to the computational costs and limited training data, phones were used as the output targets of NN-AMs [16]. Recent studies have shown that using the context-dependent HMM (CD-HMM) states as targets for NN training resulted in significant improvement in accuracy [9, 17, 18].

When training NN-AMs using CD-HMM states as output targets, each output target is treated independently. For example, while the CD-HMM states of A-b-2 (second variant of beginning state of A) and A-m-1 (first variant of middle state of A) both correspond to the phone A, and A-b-1 and A-b-2 both correspond to the Context-independent HMM (CI-HMM) state A-b, these relationships are not explicitly considered.

In this paper, we propose a NN parameter initialization method to leverage the relationship between the target CD-HMM states. With our proposed method, we first define the group of CD-HMM states based on their corresponding CI-HMM states or phones. Then, for each group, we prepare *dedicated neurons* in the final hidden layer just below the output layer. Each dedicated neuron for a specific group is initialized to connect to the output units of the CD-HMM states that belong to this group more strongly than to other output units, as later explained in Figure 1 and Figure 2. Then we conduct cross-entropy training [19] and Hessian-free state-level Minimum Bayes Risk (sMBR) sequence training [20, 21] to train the NN-AM. Please note that our proposed method operates in NN parameter initialization and requires no computational overhead in NN training or speech recognition run-time.

The idea behind the proposed method is that the grouped CD-HMM states have some common acoustic characteristics and the dedicated neurons serve as the basis to represent these characteristics. Ideally, with enough training data and a novel optimization method, the NN-AM automatically learns these characteristics. In other words, our proposed method integrates prior knowledge of the relationship between the CD-HMM states into the NN-AMs for better optimization.

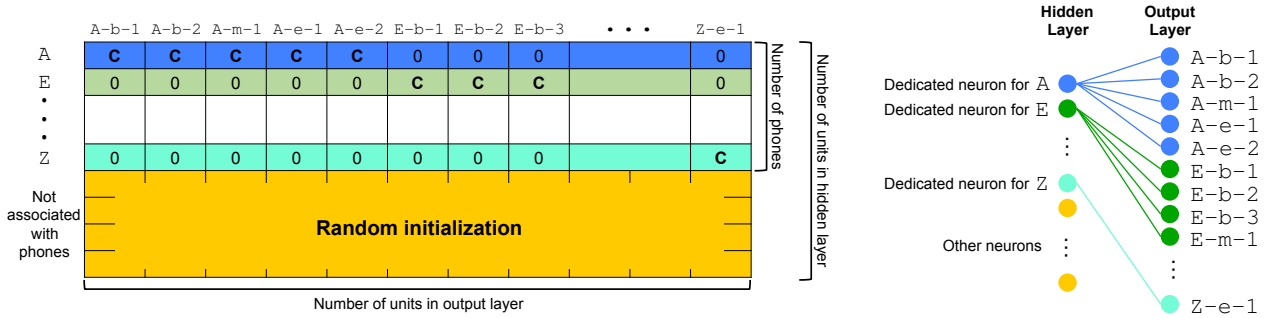
Due to the optimization difficulty in NN training, much attention has been paid to parameter initialization [22, 23, 24]. To the best of our knowledge, this is the first attempt to use the relation between the CD-HMM states for parameter initialization of NN-AMs.

We conducted experiments using a standard English broadcast news task and a Japanese LVCSR task and confirmed 3.6% to 6.5% relative error reduction after Hessian-free sequence training over a competitive baseline.

This paper has two main contributions:

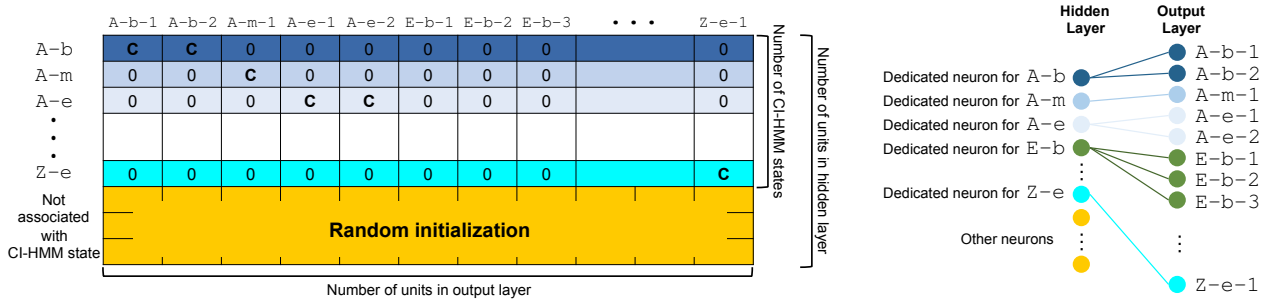
- it proposes a method to initialize NN-AMs by considering relations between the CD-HMM output targets, and
- it confirms improvement in speech recognition accuracy using the proposed initialization on an English broadcast news task and a Japanese LVCSR task.

http://dx.doi.org/10.21437/Interspeech.2016-725



Weight matrix between hidden and output layers **Strongly initialized connections between hidden and output layers**

Figure 1: *Overview of **Phone-grouping initialization**, where dedicated neurons for each phone are prepared in hidden layer, as shown on right. "Number of phones × number of output units" region of weight matrix is initialized so that dedicated neurons and corresponding output units are connected stronger than to others as shown on left.*



Weight matrix between hidden and output layers **Strongly initialized connections between hidden and output layers**

Figure 2: *Overview of **CI-HMM-grouping initialization**, where dedicated neurons for each CI-HMM state are prepared in hidden layer, as shown on right. "Number of CI-HMM states × number of output units" region of weight matrix is initialized so that dedicated neurons and corresponding output units are connected stronger than to others as shown on left.*

2. Proposed Method

As mentioned in Section 1, we focus on the NN-AM that estimates the posterior probabilities for the CD-HMM states and is combined with an HMM to form the NN/HMM hybrid system. The output targets of NN-AMs correspond to CD-HMM states and thus, there exist relationships between the states. For example, one set of CD-HMM states will correspond to the same CI-HMM state, while a still larger set of CD-HMM states will correspond to the same phone. Since the output targets are treated independently in the training of NN-AMs, the relationships between the output targets have not been exploited. We propose a method to consider these relationships in NN-AM initialization.

The key idea is to treat some of the neurons in the final hidden layer just below the output layer as dedicated neurons for each group of CD-HMM states. More specifically, the dedicated neurons are initialized to have connections to the corresponding CD-HMM states with a constant weight of C and to others with a weight of 0, as shown on the right of Figure 1 and Figure 2. This is equivalent to associating each group of CD-HMM states with a row in the weight matrix between the final hidden layer and the output layer and initializing this row so that the columns corresponding to the CD-HMM states in the group have a weight C and the other columns have a weight of 0, as shown on the left of Figure 1 and Figure 2. The remaining rows that are not associated with groups of CD-HMM states are randomly initialized. Finally, we perform cross-entropy training followed by Hessian-free sequence training. Note that all of the connection weights in the NN, including the connection weights between the dedicated neurons and all output targets, are updated during training.

We consider two variants of grouping based on phones or CI-HMM states. The proposed methods using these groupings, namely *Phone-grouping initialization* and *CI-HMM-grouping initialization* are described as below in detail. CD-HMM states are represented as `X-Y-Z`, where `X` is the phone label; `Y` is `b`, `m`, or `e` corresponding to the beginning, middle, or end states of the HMM; and `Z` is the index of different variants. $*$ represents a wild card, matching any label or index. We represent the number of phones as N_p, the number of CI-HMM states as N_{ci}, the number of CD-HMM states as N_{cd} (which is equal to the number of units in the output layer), and the number of units in the final hidden layer just below the output layer as N_h. The size of the weight matrix our proposed method operates on is $N_h \times N_{cd}$.

Phone-grouping initialization We prepare N_p dedicated neurons, where each one corresponds to a phone. The dedicated neuron for a phone `X` is initialized to have strong connections to the output targets of the CD-HMM states that belong to this phone (`X-*-*`). In Figure 1, for example, a dedicated neuron is prepared for the phone `A`, and this neuron is initialized to strongly connect to the output targets of `A-b-1`, `A-b-2`, `A-m-1`, `A-e-1`, and

	Initialization Value C	XE	HF
Baseline	n/a	17.0	15.3
PHONE	1.0	17.1	n/a
	3.0	16.5	n/a
	5.0	16.7	n/a
	7.0	16.6	n/a
	9.0	16.7	n/a
CI-HMM	1.0	16.9	n/a
	3.0	16.6	n/a
	5.0	16.6	n/a
	7.0	**16.0 (-5.9)**	**14.3 (-6.5)**
	9.0	16.5	n/a

Table 1: *WER with DNN on BN50 [%]. Numbers in parentheses are Relative Error Reduction of best results compared with Baseline.*

	Initialization Value C	XE	HF
Baseline	n/a	15.6	14.0
PHONE	1.0	15.5	n/a
	3.0	15.2	n/a
	5.0	15.0	n/a
	7.0	15.1	n/a
	9.0	15.2	n/a
CI-HMM	1.0	15.6	n/a
	3.0	15.2	n/a
	5.0	15.0	n/a
	7.0	**14.9 (-4.5)**	**13.5 (-3.6)**
	9.0	15.0	n/a

Table 2: *WER with CNN on BN50 [%]. Numbers in parentheses are Relative Error Reduction of best results compared with Baseline.*

	Initialization Value C	XE	HF
Baseline	n/a	20.4	19.7
CI-HMM	3.0	20.1	n/a
	5.0	**19.3 (-5.7)**	**18.8 (-4.6)**
	7.0	19.4	n/a

Table 3: *CER with CNN on JPN500 [%]. Numbers in parentheses are Relative Error Reduction of best results compared with Baseline.*

`A-e-2`. Thus, a $N_p \times N_{cd}$ region of the weight matrix is initialized by the proposed method to represent the N_p dedicated neurons.

CI-HMM-grouping initialization We prepare N_{ci} dedicated neurons, where each one corresponds to a CI-HMM state. The dedicated neuron for a CI-HMM state `X-Y` is initialized to have strong connections to the output targets of the CD-HMM states that belong to this CI-HMM state (`X-Y-`$*$). In Figure 2, for example, a dedicated neuron is prepared for the CI-HMM state `A-b`, and this neuron is initialized to strongly connect to the output targets of `A-b-1` and `A-b-2`. Another dedicated neuron is prepared for the CI-HMM state `A-m`, and this neuron is initialized to strongly connect to the output target of `A-m-1`. Thus, a $N_{ci} \times N_{cd}$ region of the weight matrix is initialized by the proposed method to represent the N_{ci} dedicated neurons.

$N_h \geq N_p$ is required for phone-grouping initialization, and $N_h \geq N_{ci}$ is required for CI-HMM-grouping initialization, but these requirements are usually satisfied in practical NN-AMs for LVCSR.

Please note that our proposed method does not change the NN architecture, such as the number of hidden units, at all. Therefore, considering that the computation required for the proposed initialization itself is negligible, our method does not increase computation in training or speech recognition runtime.

3. Experiments

We conducted two sets of experiments to confirm the advantage of our proposed method. First, we used a 50-hour English Broadcast News task (BN50). Then, we applied our proposed method to a larger task with 500 hours of Japanese data (JPN500).

3.1. English Broadcast News (BN50)

The acoustic model training data was a 50-hour set of English broadcast news data from the 1996 and 1997 English Broadcast News Speech corpora [20]. Evaluation was done on the Defense Advanced Research Projects Agency (DARPA) Effective Affordable Reusable Speech-to-Text (EARS) `dev04f` set using Word Error Rate (WER) as the evaluation metric. A standard language model was used for evaluation, as described in [20].

We first trained a GMM-AM according to our standard recipe [25] that yielded sufficiently accurate alignments using FMLLR transformed features [26] derived from VTLN-warped [27, 28] PLP features. Then, we trained two types of NN-AMs, a DNN and a CNN on this task. The details of the DNN and CNN are as follows.

DNN We used 40 dimensional FMLLR features with a 9-frame context (± 4) around the current frame to train the DNNs. The DNNs had 5 hidden layers of 1,024 sigmoid units, 1 hidden layer of 512 sigmoid units, and the output layer of 5,000 units corresponding to the CD-HMM states estimated by the GMM-AM.

CNN We used 40 dimensional VTLN-warped log mel-filterbank+delta+double-delta coefficients to train the CNNs. The CNNs had 2 convolutional layers, 5 hidden layers of 1,024 sigmoid units, 1 hidden layer of 512 sigmoid units, and the output layer of 5,000 units corresponding to the CD-HMM states estimated by the GMM-AM. The convolutional filter size was 9×9 (frequency-time) and 4×3 for the first and second convolutional layers, respectively. Max-pooling over frequency with a size of 3 was used for the first layer and no pooling was conducted for the second layer.

For the proposed method, the number of dedicated neurons was 42 with the phone-grouping initialization and 126 with the CI-HMM initialization. We tried various initialization weight values C to investigate the effect of the magnitude of the initialization value.

For training the DNN AM and CNN AM, we conducted cross-entropy training followed by Hessian-free sequence training. The learning rate was controlled using a held-out set [29]. For evaluation, we combined the DNN and CNN with an HMM and formed DNN/HMM and CNN/HMM hybrid systems [1, 9, 10, 29].

The DNN and CNN evaluation results are reported in Table 1 and Table 2, respectively, where "XE" and "HF" stand for the

Configurations	Connections	XE	HF
BN50 / DNN / CI-HMM / C=7.0	**Dedicated Neurons ⇒ Corresponding Outputs**	**6.748031**	**6.778025**
	Dedicated Neurons ⇒ Other Outputs	0.002167	0.001928
	All Connections	0.013836	0.013838
BN50 / CNN / CI-HMM / C=7.0	**Dedicated Neurons ⇒ Corresponding Outputs**	**6.806395**	**6.832358**
	Dedicated Neurons ⇒ Other Outputs	0.001668	0.001461
	All Connections	0.013785	0.013787
JPN500 / CNN / CI-HMM / C=5.0	**Dedicated Neurons ⇒ Corresponding Outputs**	**4.908794**	**4.908931**
	Dedicated Neurons ⇒ Other Outputs	0.003083	0.003083
	All Connections	0.012207	0.012207

Table 4: *Average connection weights between dedicated neurons and corresponding outputs, between dedicated neurons and other outputs, and between all hidden units and output units after cross-entropy training (XE) and Hessian-free sequence training (HF). Before NN training, average weights of "Dedicated Neurons ⇒ Corresponding Outputs" and "Dedicated Neurons ⇒ Other Outputs" were C and 0, respectively.*

cross-entropy training and the Hessian-free sequence training and "PHONE" and "CI-HMM" stand for the proposed method with phone-grouping initialization and CI-HMM-grouping initialization. Hessian-free sequence training was conducted on both the baseline system and the system initialized with the proposed method that performed the best after cross-entropy training.

Both the phone-grouping and CI-HMM-grouping initializations improved WER after cross-entropy training for both DNN and CNN, except in the cases of the DNN with phone-grouping initialization with C=1.0 and the CNN with CI-HMM-grouping initialization with C=1.0. Setting C to fairly large values resulted in better WER in both DNN and CNN with both phone-grouping and CI-HMM-grouping initialization. Also both with DNN and CNN, CI-HMM grouping initialization worked better than phone-grouping initialization and achieved the best WER after cross-entropy training, resulting in 5.9% and 4.5% relative error reduction in DNN and CNN, respectively. After Hessian-free sequence training, we still found improvement; 6.5% and 3.6% relative error reduction in DNN and CNN, respectively.

3.2. Japanese LVCSR (JPN500)

We applied our proposed method to a Japanese LVCSR task, where about 500 hours of continuous speech were used for AM training. Evaluation was done with 2.7 hours of lecture speech from 8 speakers. Because Japanese has ambiguity in word segmentation and word spelling, we used the Character Error Rate (CER) for the Japanese LVCSR evaluation metric instead of WER. To calculate CER, the reference transcripts and recognized results were split into characters and then aligned.

For the Japanese experiment, we used the CNN because it exhibited better accuracy in the previous BN50 experiments. We used the 40 dimensional log mel-filterbank+delta+double-delta coefficients, and other configurations were the same as previously described in the English BN50 experiments. For the proposed method, we tried CI-HMM-grouping initialization with 171 dedicated neurons.

The evaluation results are reported in Table 3. With this larger training set, we obtained 5.7% and 4.6% relative error reduction after cross-entropy training and Hessian-free sequence training, respectively.

3.3. Analysis of the Trained Models

We investigated if the dedicated neurons prepared in the NN initialization step by our proposed method still have the strong connections with the corresponding output targets after cross-entropy training and Hessian-free sequence training. We examined the best DNN and CNN from the BN50 experiments and the best CNN from the JPN500 experiment trained with the proposed method. We compared the average connection weights between the dedicated neurons and the corresponding outputs, between the dedicated neurons and other outputs, and between all hidden units and output units. The results are listed in Table 4.

In all cases, the weights between the dedicated neurons and the corresponding neurons are stronger than the weights between others: training did not substantially alter the prior structure imposed by our initialization.

4. Conclusions

We proposed a NN initialization method to use the relationship between output targets consisting of CD-HMM states. More specifically, we prepared dedicated neurons in the final hidden layer for groups of CD-HMM states corresponding to phones or CI-HMM states. The dedicated neuron for a specific group is initialized to strongly connect to the output units belonging to this group.

Through experiments on an English broadcast news task, we confirmed that CI-HMM-grouping initialization was better than phone-grouping initialization. With using CI-HMM-grouping initialization, we obtained improvement even after Hessian-free sequence training on both an English broadcast news task and a Japanese LVCSR task. We also confirmed that after cross-entropy and sequence training, the dedicated neurons still have strong connections to the corresponding output units, which suggests that the dedicated neurons serve as the basis for the corresponding groups of the CD-HMM states.

Our future work includes appropriately setting the initialization value C. We tried various initialization values and obtained improvement in most cases, which supports the potential of our proposed method. It is more preferable to have a sophisticated method to set an appropriate initialization value that results in improvement.

Preparing multiple dedicated neurons for each group is another extension. Characteristics of distinct states in the same group can be represented by the separate dedicated neurons if necessary.

Another direction we would like to pursue is to define finer-grained groups of CD-HMM states based on the structure of the phonetic decision trees used for CD-H MM state clustering.

5. Acknowledgements

We would like to thank Bowen Zhou and Bing Xiang of IBM Watson for the fruitful discussion. We are grateful to Ryuki Tachibana of IBM Watson for valuable comments.

6. References

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[3] T. N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A. Mohamed, G. Dahl, and B. Ramabhadran, "Deep convolutional neural networks for large-scale speech tasks," *Neural Networks*, vol. 64, pp. 39–48, 2015.

[4] G. Saon, H. Soltau, A. Emami, and M. Picheny, "Unfolded recurrent neural networks for speech recognition." in *Proc. INTERSPEECH*, 2014, pp. 343–347.

[5] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." in *Proc. INTERSPEECH*, 2014, pp. 338–342.

[6] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. ICASSP*, 2015, pp. 4580–4584.

[7] D. Yu and M. L. Seltzer, "Improved bottleneck features using pretrained deep neural networks." in *Proc. INTERSPEECH*, 2011, pp. 237–240.

[8] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Proc. ICASSP*, 2013, pp. 3377–3381.

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

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

[11] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, 2013, pp. 6645–6649.

[12] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[13] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NIPS*, 2015, pp. 577–585.

[14] L. Lu, X. Zhang, K. Cho, and S. Renals, "A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition," in *Proc. INTERSPEECH*, 2015, pp. 3249–3253.

[15] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," *arXiv preprint arXiv:1507.06947*, 2015.

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

[17] D. Yu, L. Deng, and G. Dahl, "Roles of pre-training and fine-tuning in context-dependent dbn-hmms for real-world speech recognition," in *Proc. NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2010.

[18] 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.

[19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.

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

[21] 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*, 2012.

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

[23] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. ICML*, 2013, pp. 1139–1147.

[24] G. Kurata, B. Xiang, and B. Zhou, "Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence," in *Proc. NAACL-HLT*, 2016 (to appear).

[25] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," in *Proc. SLT*, 2010, pp. 97–102.

[26] M. J. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech & Language*, vol. 12, no. 2, pp. 75–98, 1998.

[27] S. Wegmann, D. McAllaster, J. Orloff, and B. Peskin, "Speaker normalization on conversational telephone speech," in *Proc. ICASSP*, vol. 1, 1996, pp. 339–341.

[28] H. Soltau, B. Kingsbury, L. Mangu, D. Povey, G. Saon, and G. Zweig, "The IBM 2004 conversational telephony system for rich transcription." in *Proc. ICASSP*, vol. 1, 2005, pp. 205–208.

[29] 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.