TIME-DELAYED BOTTLENECK HIGHWAY NETWORKS USING A DFT FEATURE FOR KEYWORD SPOTTING

Jinxi Guo[†], *Kenichi Kumatani*, *Ming Sun*,
Minhua Wu, *Anirudh Raju*, *Nikko Ström*, *Arindam Mandal*
lennyguo@g.ucla.edu[†], {kumatani, mingsun}@amazon.com
Department of Electrical and Computer Engineering, University of California, Los Angeles, USA[†]
Amazon Inc. USA

ABSTRACT

This paper presents a novel deep neural network (DNN) architecture with highway blocks (HWs) using a complex discrete Fourier transform (DFT) feature for keyword spotting. In our previous work, we showed that the feed-forward DNN with a time-delayed bottleneck layer (TDB-DNN) directly trained from the audio input outperformed the model with the log-mel filter bank energy feature (LFBE), given a large amount of training data [1]. However, the deeper structure of such an audio input DNN makes an optimization problem more difficult, which could easily fall in one of the local minimum solutions. In order to alleviate the problem, we propose a new HW network with a time-delayed bottleneck layer (TDB-HW). Our TDB-HW networks can learn a bottleneck feature representation through optimization based on the cross-entropy criterion without stage-wise training proposed in [1]. Moreover, we use the complex DFT feature as a method of pre-processing. Our experimental results on the *real* data show that the TDB-HW network with the complex DFT feature provides significantly lower miss rates for a range of false alarm rates over the LFBE DNN, yielding approximately 20 % relative improvement in the area under the curve (AUC) of the detection error tradeoff (DET) curves for keyword spotting. Furthermore, we investigate the effects of different pre-processing methods for the deep highway network.

Index Terms— Keyword Spotting, Highway Networks, Audio Input Acoustic Modeling

1. INTRODUCTION

Wake-word (WW) detection is the first important step before interactions through distant speech recognition [2–9]. WW detectors typically employ signal-processing techniques to obtain a compact feature representation such as LFBE [4–10] and tandem features [11].

Recently, there has been a great deal of attention paid to a fully trainable DNN front-end because of its scalability for large data sets. Several types of DNN architecture have been proposed to capture time-frequency characteristics of speech and model dynamics of speech features directly from raw features. Such DNN structures can fall into one of the following categories: Feed forward DNN using stacked bottleneck features from raw audio features [1, 12], Network-in-network DNN with supplemental signal statistics of hidden layer outputs [13], Convolutional layers with long short-term memory (LSTM) layers [14], Complex linear prediction (CLP) layer with LSTM layers instead of time-domain convolution [15].

While majority of the systems are focused on large-scale neural network models, little research has been done on designing a compact model for embedded platforms, which can directly model the raw feature input. In our previous work, we proposed a time-delay neural network (TDNN) [16] with a bottleneck layer between two sets of fully connected layers [1]: one for feature extraction, and the other for acoustic modeling. The resultant DNN becomes very deep. Such a deep structure suffers from the vanishing and exploding gradient problems. it can also easily converge to local minimum points. The three-stage training procedure described in [1] is very time consuming.

In order to alleviate the issues, we propose a unified highway (HW) network [17] with a time-delayed bottleneck (TDB) layer in the middle. Our TDB-HW network also has two parts: a feature extractor and a phone classifier. Each of them consists of stacked highway blocks, which can control the information flow between layers and make it feasible to train a very deep neural network. The HW networks have also been proven to be very efficient to reduce the size of the networks without sacrificing the recognition accuracy for small-footprint ASR [18]. Similarly, we adopt a thinner and deeper HW structure in this work. The bottleneck layer can force the network to learn the most salient features and can also greatly reduce a network size as well as computation cost. Moreover, the TDB-HW network can be trained from scratch, which can reduce an amount of training time by nearly 60 % of the stage-wise training. For the feature input, we use the discrete Fourier transform (DFT) as a method of signal normalization and generate the complex DFT features. Essentially, this is the same method presented in Variani's work [15]. Since the row vector of the input layer functions as a band-pass filter for raw audio DNN model as shown in [1, 12], DFT DNNs should decrease a chance of converging into a trivial local solution.

Notice that this work only focuses on feature extraction on the single channel audio after beamforming [3, 19] in contrast to the multi-channel audio DNN work [15, 20, 21] where the beamforming and feature extraction layers are trained directly from the multi-channel data. We will show that a significant improvement can be achieved with the single channel data only.

The balance of the paper is organized as follows. Section 2 describes our baseline WW system with the LFBE feature, including the DNN-HMM based WW detection. Section 3 presents our deep highway network architecture with the DFT input. In Section 4, WW detection results are described. Section 5 concludes this work.

2. BASELINE WW SYSTEM

In this paper, our keyword spotting task refers to the application on wake-word (WW) detection. We will use the term WW. We employ the HMM-based approach with the WW and filler/background HMMs [10].

[†]The author contributed to this work as an intern at Amazon Alexa.

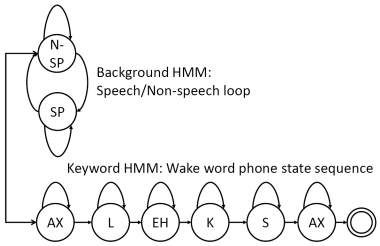


Fig. 1. HMM-based Keyword Spotting.

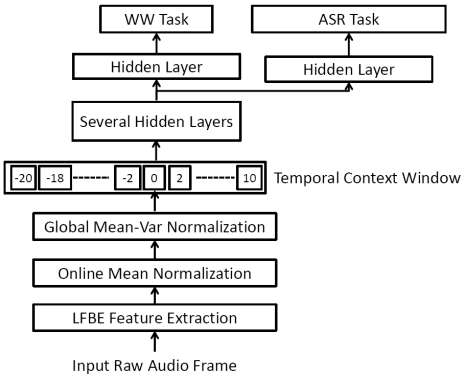


Fig. 2. Baseline WW DNN with the LFBE feature.

Figure 1 illustrates an example of the finite state transducer (FST) at a phone level for a WW 'Alexa', with six phones in its pronunciation. In our experiments 3-state HMMs are used to model 'Alexa' phones. For simplicity, only one-state HMMs are plotted in Figure 1. The HMM state is associated with a DNN. The output layer of the DNN models the HMM states of the keyword(s) of interest (i.e., WW-specific phone state distributions) and the two 1-state background phones (speech and non-speech); also see [9] for a WW system with more generic background phones.

Figure 2 shows a schematic view of the baseline DNN system. Our baseline system first computes the LFBE feature from the enhanced speech [10]. In our system, an audio signal is divided into overlapping frames of 25 ms with a frame shift of 10 ms. The LFBE features concatenated over multiple frames are then fed into the acoustic modeling DNN. The DNN consists of several layers of affine transform and sigmoid activation components. In addition to those layers, we put two separate branches for WW and ASR tasks so as to jointly classify the WW-specific phones and the context-dependent phones (LVCSR senones) based on our previously proposed multi-task training technique [10]. After the DNN is pre-trained layer-wise in a supervised fashion using a small subset of the training data, the entire DNN is further optimized with a distributed, asynchronous, stochastic gradient descent (SGD) training method [22] over the full dataset.

As illustrated in the FST of Figure 1, the WW hypothesis is generated when the final state of the WW FST is reached. We tune transition parameters and exit penalties in the WW and background HMMs for better accuracy, and a detection error trade-off (DET) curve can be obtained by plotting the lowest achievable false alarm rate (FAR) at a given miss rate (MR) or false reject rate (FRR).

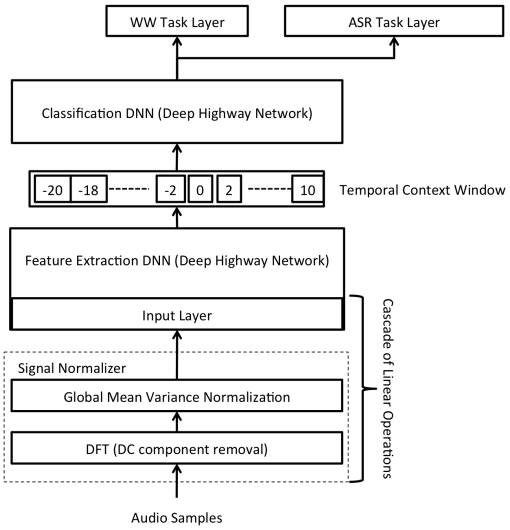


Fig. 3. Whole WW Highway DNN with the DFT input.

3. DFT-INPUT HIGHWAY NETWORKS

In this section, we will introduce the highway blocks, and show the structures of the proposed TDB-HW networks. We will also list different structures to be compared.

3.1. Highway Blocks

Highway networks were first proposed by [17], and their basic element is the highway block. In the highway block, the output at the l-th layer is controlled by two gating functions: a carry gate $C(\mathbf{h}_{l-1})$ that controls the information flow directly from the previous hidden layer $\mathbf{h}_{l-1}$, and a transform gate $T(\mathbf{h}_{l-1})$ that controls information from the hidden activation $f(\mathbf{h}_{l-1})$. The final output is defined by:

$$\mathbf{h}_l = f(\mathbf{h}_{l-1}) \cdot T(\mathbf{h}_{l-1}) + \mathbf{h}_{l-1} \cdot C(\mathbf{h}_{l-1}) \quad (1)$$

Both carry and transform gate functions are defined by a nonlinear layer with Sigmoid function:

$$T(\mathbf{h}_{l-1}) = \sigma(\mathbf{W}_T \mathbf{h}_{l-1} + \mathbf{b}_T), \quad (2)$$

$$C(\mathbf{h}_{l-1}) = \sigma(\mathbf{W}_C \mathbf{h}_{l-1} + \mathbf{b}_C), \quad (3)$$

$$f(\mathbf{h}_{l-1}) = \sigma(\mathbf{W}_l \mathbf{h}_{l-1} + \mathbf{b}_l) \quad (4)$$

From our preliminary experiments, we observed that the HW network was easier to train without a bias vector. Therefore, we do not use the bias vector in the gate function. Moreover, in contrast to [17], we do not impose any constrain on two gates. The highway blocks can control information flow and gradient propagation between layers, which makes it feasible to train a very deep neural network and can also speed up the convergence rate.

3.2. Highway networks with the time-delayed bottleneck layer

In order to make the DNN learn the feature representation from complex DFT features generated from the linear signal normalizer, we

first use 4 stacked HW blocks and a bottleneck layer as the feature extractor. The bottleneck layer can reduce the large dimensionality of the input features (concatenated complex DFT coefficients), which may force the network to learn the most salient representation. The design of the bottleneck layer can also significantly reduce the network size, which makes it feasible for resource-constrained conditions. In this architecture, we use a linear bottleneck layer since our experimental results indicate that a linear layer performs slightly better than a non-linear layer. After the bottleneck layer, we use a time-delayed window to splice the bottleneck features from several past and future frames so as to capture the temporal information for phone classification. For the classification DNN, we use 6 HW blocks stacked together. Figure 3 illustrates our entire highway DNN architecture with the DFT normalizer for WW.

For the bottleneck layer, we use 28 output dimension. For the time-delayed window, we select 20 left and 10 right contexts from the bottleneck layer output. For back-propagation, the weights are updated when the gradients are accumulated from all the contexts in the window. In order to reduce the number of parameters, we tie the two gate weights of each layer inside the feature extraction DNN and also inside the classification DNN. The entire DNN is also optimized based on the multi-task cross-entropy criterion. The TDB-HW networks are directly trained from scratch with random initialization. Here random initialization refers to light supervised pre-training in a layer-wise manner on a small subset of training data.

3.3. Comparing architectures

In this paper, we will compare the TDB-HW networks with TDB-DNNs using complex DFT features. For the TDB-DNNs, we follow the same three-stage training procedure as described in [1]; first, we train a feature extraction DNN with a bottleneck layer on top. Then, we use a context window to splice the bottleneck features across several frames and use the stacked features to train the acoustic modeling DNNs. Finally, we joint optimize the feature extraction and acoustic modeling DNNs, by training the unified network as a DNN with a time-delayed bottleneck layer in the middle. We will also compare the complex DFT systems with LFBE systems. In order to have a fair comparison, besides the feed-forward DNN baseline system described in section 2, we also design a regular HW network (tie the two gate weights for each layer) using LFBE. All the networks compared in this paper have the same depth (11 layers) and similar number of parameters (around 3 M).

4. EXPERIMENTS AND RESULTS

Here, all the results are shown in the form of DET curves along with area under the curve (AUC) numbers. All the DET curves in this paper only show false alarm rates up to a multiplicative constant because of the sensitive nature of this information. The DET curves and AUC numbers presented here therefore indicate the relative improvement or degradation against the baseline system.

The training data used in this work consist of several thousand hours of the real far-field data captured in various rooms. This contains approximately several hundred thousand subjects. In order to improve the robustness against noise unseen in the training data, the training data are artificially corrupted and the SNR is adjusted from 0 to 40 dB uniformly. Our test set contains over several thousands of speech segments uttered by hundreds of subjects. The test data contain approximately 26,000 WW instances. The captured far-field array data are processed with beamforming and acoustic echo cancellation [3, 19].

4.1. Comparison of different DNN architectures

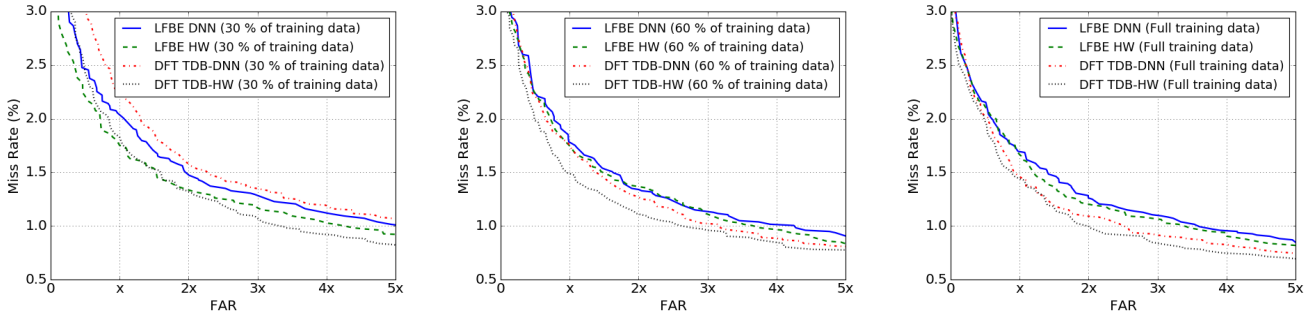
Figure 4 shows the DET curves obtained with the LFBE DNN, LFBE HW, DFT TDB-DNN and DFT TDB-HW on the test set with different amounts of training data. In order to generate the DET curves for Figure 4, we choose the best FST parameters with 4 HMM thresholds. Since we choose the FST parameters from the same pool of the FSTs, this result comparison is still fair. Those DET curves on the test data indicate the best possible WW performance without the grammatical language constraint.

It is clear from Figure 4 that the HW-based networks provide better accuracy than the DNN-based networks with both LFBE and DFT features. The effect of HW blocks is very prominent on training a deep network, especially under the 30%-training-data condition. As the amount of training data increases, the difference between the regular HW networks and DNNs becomes smaller for the LFBE system. This indicates that the hard optimization problem of deep network can be alleviated by a large amount of training data. For the DFT systems, the improvement of TDB-HW networks is still significant compared with TDB-DNN, even when trained on a large amount of data. The good performance of the TDB-HW networks may be a result of its ability of training a unified structure from scratch. By joint optimizing the feature extractor and phone classifier using HW blocks from scratch, the TDB-HW networks are able to learn the most useful features, which are also highly optimized for phone classification. In contrast, the TDB-DNN's three-stage training procedure may not be able to achieve such a global optimization. Overall, the proposed DFT system provides better performance than LFBE systems. From the AUC numbers in Figure 6 we can observe that, using full training data, the proposed DFT TDB-HW networks can outperform the baseline system (LFBE DNN) by 19.4%.

4.2. Effect of different feature inputs

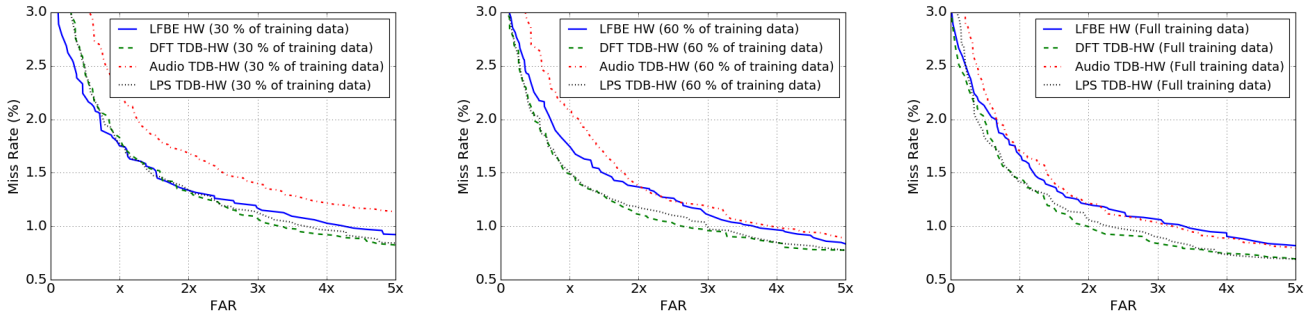
From the speech feature extraction point of view, it could be interesting to investigate the effect of different features as the HW network input. Figure 5 shows the DET curves of the HW-based networks obtained with the LFBE (LFBE HW), DFT coefficients (DFT TDB-HW), raw audio (AUDIO TDB-HW) and log-power spectrum (LPS TDB-HW) features. Figure 7 shows the AUC graphs that correspond to Figure 5. It is clear from Figure 5 and 7 that, when using 30% of training data, the LFBE, DFT and LPS features have similar performance. As the number of training data increase, the DFT coefficients start to provide the best performance, which is slightly better than LPS features and significantly better than raw audio and LFBE features. The results suggest that the phase information can provide slightly improvement for acoustic modeling and a fully trainable front-end can provide significant improvement compared with auditory-based hand-crafted features (LFBE). The AUC numbers in 7 shows that the DFT TDB-HW network gives more than 16% improvement compared with LFBE HW network, when using 60% or 100% of training data.

The DFT TDB-HW network uses the linear-normalized raw audio input, without any non-linear pre-processing based on auditory knowledge. While the LPS feature's computation involves the non-linear process which takes more computation. For the raw-audio-input condition, we believe that the raw audio DNN may easily converge to trivial local minima due to absence of adequate normalization also as indicated in Bhargava's work [12].



(a) 30 % of training data (b) 60 % of training data (c) Full amount of training data

Fig. 4. DET curves of LFBE DNN, LFBE HW, DFT TDB-DNN, DFT TDB-HW on different amounts of data



(a) 30 % of training data (b) 60 % of training data (c) Full amount of training data

Fig. 5. DET curves of LFBE HW, DFT TDB-HW, Audio TDB-HW, LPS TDB-HW on different amounts of data

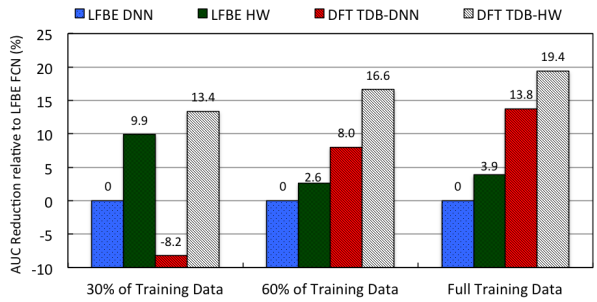


Fig. 6. AUCs calculated from figure 4

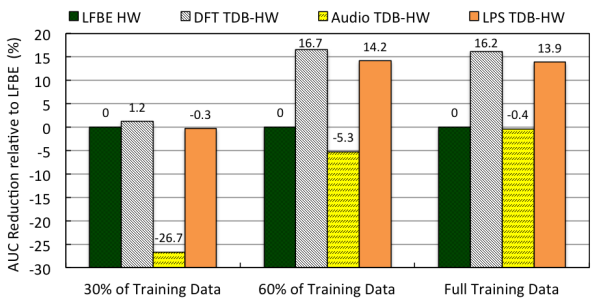


Fig. 7. AUCs calculated from figure 5

5. CONCLUSION

In this paper, we have proposed the novel highway network with a time-delayed window component on the bottleneck layer. The proposed network can directly model the audio signal by cascading a DFT-based linear normalizer. Through the WW experiments on the *real* far-field data, we have demonstrated that our TDB-HW networks with the complex DFT feature reduced the AUC of approximately 20 % relative to the feed-forward DNN with the LFBE feature in the case that a large amount of training data was available. We also investigated different pre-processing methods in feature extraction for the deep highway network. It turned out that DFT-based normalization on the audio signal could provide better performance than direct audio-input given the same amount of training data. It should be noted that this DFT normalization process does not result in an increase in computational complexity during decoding because the cascade of the linear operations from the DFT to the input layer can be represented as a single affine transform; This is indeed an advantage against feature extraction including non-linear processing such as log-power spectrum.

Acknowledgement

We would like to you thank Sankaran Panchapagesan, Gautam Tiwari, Aaron Challenner and Andrew Grasberger for supporting the experiment.

6. REFERENCES

[1] Kenichi Kumatani, Sankaran Panchapagesan, Minhua Wu, Minjae Kim, Nikko Ström, Gautam Tiwari, and Arindam Mandal, "Direct modeling of raw audio with dnns for wake word detection," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017.

[2] Kenichi Kumatani, John W. McDonough, and Bhiksha Raj, "Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 127–140, 2012.

[3] Rohit Prasad, "Spoken Language Understanding for Amazon Echo," 2015, Keynote in Speech and Audio in the Northeast (SANE).

[4] M. Sun, V. Nagaraja, B. Hoffmeister, and S. Vitaladevuni, "Model shrinking for embedded keyword spotting," in *International Conference on Machine Learning and Applications (ICMLA)*, 2015, pp. 369–374.

[5] George Tucker, Minhua Wu, Ming Sun, Sankaran Panchapagesan, Gengshen Fu, and Shiv Vitaladevuni, "Model compression applied to small-footprint keyword spotting," in *Proc. Interspeech*, 2016, pp. 1878–1882.

[6] M. Sun, A. Raju, G. Tucker, S. Panchapagesan, G. Fu, A. Mandal, S. Matsoukas, N. Ström, and S. Vitaladevuni, "Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting," in *IEEE Spoken Language Technology Workshop (SLT) Workshop*, 2016.

[7] M. Sun, D. Snyder, Y. Gao, V. Nagaraja, M. Rodehorst, S. Panchapagesan, N. Ström, S. Matsoukas, and S. Vitaladevuni, "Compressed time delay neural network for small-footprint keyword spotting," in *Proc. Interspeech*, 2017, pp. 3607–3611.

[8] M. Sun, A. Schwarz, M. Wu, N. Ström, S. Matsoukas, and S. Vitaladevuni, "An empirical study of cross-lingual transfer learning techniques for small-footprint keyword spotting," in *International Conference on Machine Learning and Applications (ICMLA)*, 2017.

[9] Minhua Wu, Sankaran Panchapagesan, Ming Sun, Jiacheng Gu, Ryan Thomas, Shiv Naga Prasad Vitaladevuni, Björn Hoffmeister, and Arindam Mandal, "monophone-based background modeling for two-stage on-device wake word detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[10] Sankaran Panchapagesan, Ming Sun, Aparna Khare, Spyros Matsoukas, Arindam Mandal, Björn Hoffmeister, and Shiv Vitaladevuni, "Multi-task learning and weighted cross-entropy for dnn-based keyword spotting," in *Proc. Interspeech*, 2016, pp. 760–764.

[11] Nelson Morgan, "Deep and wide: Multiple layers in automatic speech recognition," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 11, 2012.

[12] M. Bhargava and R. Rose, "Architectures for deep neural network based acoustic models defined over windowed speech waveforms," in *Proc. Interspeech*, 2015.

[13] P. Ghahremani, V. Manohar, D. Povey, and S. Khudanpur, "Acoustic modelling from the signal domain using CNNs," in *Proc. Interspeech*, 2016.

[14] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," in *Proc. Interspeech*, 2015.

[15] Ehsan Variani, Tara N. Sainath, Izhak Shafran, and Michiel Bacchiani, "Complex linear projection (CLP): A discriminative approach to joint feature extraction and acoustic modeling," in *Proc. Interspeech*, 2016.

[16] Kevin J. Lang, Alex Waibel, and Geoffrey E. Hinton, "A time-delay neural network architecture for isolated word recognition," *Neural Networks*, vol. 3, no. 1, pp. 23–43, 1990.

[17] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," in *arXiv preprint: 1505.00387*, 2015.

[18] Liang Lu and Steve Renals, "Small-footprint deep neural networks with highway connections for speech recognition," in *Proc. Interspeech*, 2016.

[19] M. Wölfel and J.W. McDonough, *Distant Speech Recognition*, Wiley, London, 2009.

[20] Tara N. Sainath, Arun Narayanan, Ron J. Weiss, Ehsan Variani, Kevin W. Wilson, Michiel Bacchiani, and Izhak Shafran, "Reducing the computational complexity of multimicrophone acoustic models with integrated feature extraction," in *Proc. Interspeech*, 2016.

[21] Bo Li, Tara N. Sainath, Ron J. Weiss, Kevin W. Wilson, and Michiel Bacchiani, "Neural network adaptive beamforming for robust multichannel speech recognition," in *Proc. Interspeech*, 2016.

[22] Nikko Ström, "Scalable distributed DNN training using commodity GPU cloud computing," in *Proc. Interspeech*, 2015, pp. 1488–1492.