Neural Network Adaptive Beamforming for Robust Multichannel Speech Recognition

Bo Li, Tara N. Sainath, Ron J. Weiss, Kevin W. Wilson, Michiel Bacchiani

Google Inc.

{boboli,tsainath,ronw,kwwilson,michiel}@google.com

Abstract

Joint multichannel enhancement and acoustic modeling using neural networks has shown promise over the past few years. However, one shortcoming of previous work [1, 2, 3] is that the filters learned during training are fixed for decoding, potentially limiting the ability of these models to adapt to previously unseen or changing conditions. In this paper we explore a neural network adaptive beamforming (NAB) technique to address this issue. Specifically, we use LSTM layers to predict time domain beamforming filter coefficients at each input frame. These filters are convolved with the framed time domain input signal and summed across channels, essentially performing FIR filter-and-sum beamforming using the dynamically adapted filter. The beamformer output is passed into a waveform CLDNN acoustic model [4] which is trained jointly with the filter prediction LSTM layers. We find that the proposed NAB model achieves a 12.7% relative improvement in WER over a single channel model [4] and reaches similar performance to a "factored" model architecture which utilizes several fixed spatial filters [3] on a 2,000-hour Voice Search task, with a 17.9% decrease in computational cost.

Index Terms: speech recognition, multichannel, beamforming, adaptive filtering

1. Introduction

While automatic speech recognition (ASR) performance has improved dramatically in recent years, particularly with the advent of deep learning [5], performance in realistic noisy and far-field scenarios is still far-behind clean speech conditions [6, 7, 8] To improve robustness, microphone arrays are commonly utilized to enhance the speech signal and eliminate unwanted noise and reverberation [9, 10].

A widely adopted multichannel signal processing technique is delay-and-sum (DS) beamforming [10], in which signals from different microphones are aligned in time to adjust for the propagation delay from the target speaker to each microphone, and then mixed to a single channel. This has the effect of enhancing the signal from the target direction and attenuating noise coming from other directions. However, it is difficult to accurately estimate the time delay of arrival in reverberant environments [11] and DS beamforming does not take into account the effect of spatially correlated noise. It is possible to improve performance using the more general filter-and-sum (FS) technique [12, 13], where a linear filter is applied to each channel before summing. Such filters are commonly chosen to optimize signal level objectives such as SNR [10, 14, 15], which differ from the acoustic model (AM) training objective.

Joint training of enhancement and AM stages has led to performance improvements, both for Gaussian mixture model [16] and neural network [2, 3, 17] acoustic models. For example,

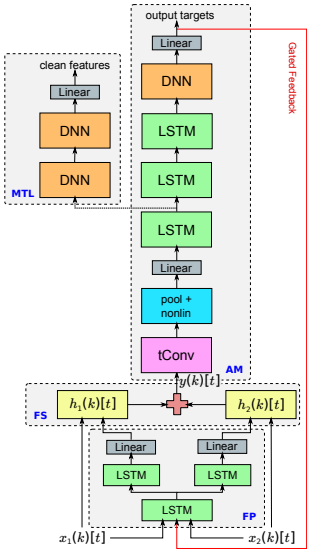

Figure 1: *Neural network adaptive beamforming (NAB) model architecture. It consists of filter prediction (FP), filter-and-sum (FS) beamforming, acoustic modeling (AM) and multitask learning (MTL) blocks. Only two channels are shown for simplicity.*

[2] trains neural nets to operate directly on multichannel waveforms using a single layer of multichannel "time convolution" FIR filters, each of which independently filters each channel of the input and then sums the outputs in a process analogous to FS beamforming. After training, the filters in this multichannel filterbank learn to jointly perform spatial and spectral filtering, with typical filters having a bandpass response in frequency, but steered to enhance or attenuate signals arriving from different directions. A factored multichannel waveform model is proposed in [3] which separates the spatial and spectral filtering behavior into separate layers, and improves performance, but comes at a large increase in computational complexity. While both of these architectures have shown improvements over traditional DS and FS signal processing techniques, one drawback is that the estimated spatial and spectral filters are fixed during decoding.

To address the limited adaptability and reduce the computational complexity of the models from [2, 3], we propose a neural network adaptive beamforming (NAB) model which re-estimates a set of spatial filter coefficients at each input frame using a neural network. Specifically, raw multichannel waveform signals are passed into a filter prediction (FP) LSTM whose outputs are used as spatial filter coefficients. These spatial filters for each channel are then convolved with the corresponding waveform input, and the outputs are summed together to form a single channel output waveform containing the enhanced speech signal. The resulting

single channel signal is passed to a raw waveform acoustic model similar to [18], which is trained jointly with the FP LSTM layers. A similar model was proposed in [17], although filtering was performed in the frequency domain, as opposed to our model which processes time domain signals. We will show in the results section that performing NAB in the time domain requires estimation of many fewer filter coefficients, and results in better WER compared to frequency domain filter prediction.

In addition, we propose two improvements to the NAB model. First, we explore explicitly feeding activations of the upper layers of the acoustic model from the previous time step, which capture high-level information about the acoustic states, as an additional input to the FP layers. A gating mechanism is further adopted [19] to attentuate the potential errors in these predictions. It analyzes the predictions together with inputs and model states to output a confidence score that scales down the feedback vectors when necessary. Second, we incorporate a multitask learning (MTL) strategy to regularize training and aid in filter prediction. This works by training the NAB model to jointly predict acoustic model states and clean features, which has previously been shown to improve acoustic models trained on noisy data [3, 20].

2. Neural Network Adaptive Beamforming

The proposed neural network adaptive beamforming (NAB) model is depicted in Figure 1. At each time frame k, it takes in a small window of M waveform samples for each channel c from the C channel inputs, denoted as $x_1(k)[t], x_2(k)[t], \cdots, x_C(k)[t]$ for $t \in \{1, \cdots, M\}$.

2.1. Adaptive Spatial Filtering

A finite impulse response (FIR) filter-and-sum beamformer, which can be written as:

$$y[t] = \sum_{c=0}^{C-1} \sum_{n=0}^{N-1} h_c[n] x_c[t - n - \tau_c] \quad (1)$$

where $h_c[n]$ is the n-th tap of the filter associated with microphone c, $x_c[t]$ is the signal received by microphone c at time t, τ_c is the steering delay induced in the signal received by a microphone to align it to the other array channels, and $y[t]$ is the output signal. N is the length of the filter.

Enhancement algorithms that optimize Equation 1 require an estimate of the steering delay τ_c, which is typically obtained from a separate localization model [21]. The filter coefficients are often obtained by optimizing signal-level objectives [12, 13]. In the NAB model, we estimate the filter coefficients jointly with the AM parameters by directly minimizing a cross-entropy or sequence loss function. Instead of explicitly estimating the steering delay for each microphone, τ_c can be implicitly absorbed into the estimated filter coefficients. The resulting adaptive filtering at each time frame k is given by Equation 2, where $h_c(k)[t]$ is the estimated filter for channel c at time frame k.

$$y(k)[t] = \sum_{c=0}^{C-1} \sum_{n=0}^{N-1} h_c(k)[n] x_c(k)[t - n] \quad (2)$$

In order to estimate $h_c(k)[t]$, we train an FP LSTM to predict N filter coefficients per channel. The input to the *FP module* is a concatenation of frames of raw input samples $x_c(k)[t]$ from all the channels, and can also include features typically computed for localization such as cross correlation features [21, 17, 22]. We describe the *FP module* architecture in more detail in Section 3.2. Following Equation 2 the estimated filter coefficients $h_c(k)[t]$ are convolved with input samples $x_c(k)[t]$ for each channel. The outputs of the convolution are summed across channels to produce a single channel signal $y(k)[t]$.

2.2. Acoustic Modeling

The single channel enhanced signal $y(k)[t]$ is passed to the *AM module* shown in Figure 1, which is similar to the CLDNN AM from [4]. The single channel waveform is passed into a "time convolution" layer, denoted as `tConv`, which acts as a time-domain filterbank containing 128 filters. The `tConv` output is decimated in time by max-pooling over the length of the input frame. Finally, a rectifier non-linearity and stabilized logarithm compression are applied to each filter output, to produce a frame-level feature vector at frame k.

Unlike CLDNN models used in [2, 3, 4], we do not include a frequency convolution layer. The feature vector generated by the time convolution layer is directly passed to three LSTM layers with 832 cells and a 512-dimensional projection layer, followed by a fully connected DNN layer of 1,024 hidden units. A 512-dimensional linear output low rank projection layer is used prior to the softmax layer to reduce the number of parameters needed to classify the 13,522 context-dependent state output targets used[23]. After processing the frame k, we shift the window of the overall input signal by a 10 ms hop and repeat this process.

The *AM* and *FP modules* are trained jointly, however the FS block has no trainable parameters. The model is unrolled 20 time steps for training using truncated back-propagation through time. The output state label is delayed by 5 frames, as we have found that using information about future frames improves the prediction of the current frame [18].

2.3. Gated Feedback

Augmenting the network input at each frame with the prediction from the previous frame has been shown to improve performance [24]. To investigate the benefit of feedback in the NAB model, we pass the AM prediction at frame $k-1$ back to the FP model at time frame k (red line in Figure 1). Since the softmax prediction is very high dimensional, we feed back the low-rank activations preceding the softmax to the *FP module* to limit the increase of model parameters [25].

This feedback connection gives the *FP module* high level information about the phonemic content of the signal to aid in estimating beamforming filter coefficients. This feedback is comprised of model *predictions* which may contain errors, particularly early in training, and therefore might lead to poor model training [24]. A gating mechanism [19] is hence introduced to the connection to modulate the degree of feedback. Unlike conventional LSTM gates, which control each dimension independently, we use a global scalar gate to moderate the feedback. The gate $g^{\texttt{fb}}(k)$ at time frame k, is computed from the input waveform samples $\boldsymbol{x}(k)$, the state of the first FP LSTM layer $\boldsymbol{s}(k-1)$, and the feedback vector $\boldsymbol{v}(k-1)$, as follows:

$$g^{\texttt{fb}}(k) = \sigma(\boldsymbol{w}_x^T \cdot \boldsymbol{x}(k) + \boldsymbol{w}_s^T \cdot \boldsymbol{s}(k-1) + \boldsymbol{w}_v^T \cdot \boldsymbol{v}(k-1)) \quad (3)$$

where $\boldsymbol{w}_x$, $\boldsymbol{w}_s$ and $\boldsymbol{w}_v$ are the corresponding weight vectors and σ is an elementwise non-linearity. We use a logistic function for σ which outputs values in the range $[0, 1]$, where 0 cuts off the feedback connection and 1 directly passes the feedback through. The effective FP input is hence $\Big[\boldsymbol{x}(k), \quad g^{\texttt{fb}}(k)\boldsymbol{v}(k-1)\Big]$.

2.4. Regularization with Multitask Learning

Multitask learning has been shown to yield improved robustness [3, 20, 26]. We adopt an *MTL module* similar to [3] during training by configuring the network to have two outputs, one recognition output which predicts CD states and a second denoising output which reconstructs 128 log-mel features derived from the underlying clean signal. The denoising output is only used in training to regularize the model parameters; the associated layers are discarded during inference. In the NAB model the *MTL module* branches off of the first LSTM layer of the *AM module*, as shown in Figure 1. The *MTL module* is composed of two fully connected DNN layers followed by a linear output layer which predicts clean features. During training the gradients back propagated from the two outputs are weighted by α and $1 - \alpha$ for the recognition and denoising outputs respectively.

3. Experiments

3.1. Experimental Setup

Our experiments are conducted on about 2,000 hours of noisy training data consisting of 3 million English utterances. This data set is created by artificially corrupting clean utterances using a room simulator, adding varying degrees of noise and reverberation. The clean utterances are anonymized and hand-transcribed voice search queries, and are representatives of Google's voice search traffic. Noise signals, which include music and ambient noise sampled from YouTube and recordings of "daily life" environments, are added to the clean utterances at SNRs ranging from 0 to 20 dB, with an average of about 12 dB. Reverberation is simulated using the image model [27] with room dimensions and microphone array positions that are randomly sampled from 100 possible room configurations with T_{60}s ranging from 400 to 900 ms, with an average of about 600 ms. The first and last channel of an 8-channel linear microphone array are used, which has a micophone spacing of 14 cm. Both noise and target speaker locations vary across utterances; the distance between the sound source and the microphone array is chosen between 1 to 4 meters. The speech and noise azimuths were uniformly sampled from the range of ± 45 degrees and ± 90 degrees, respectively, for each noisy utterance.

Our evaluation set consists of a separate set of about 30,000 utterances (over 200 hours). It is created similarly to the training set under similar SNR and reverberation settings. Care was taken to ensure that the room configurations, SNR values, T_{60} times, and target speaker and noise positions in the evaluation set are not identical to those in the training set, although the microphone array geometry between the training and test sets is identical.

Input features for raw waveform models are computed using an input window size of 35 ms, with a 10 ms hop between frames, similar to [2, 3]. Unless otherwise indicated, all networks are trained with 128 `tConv` filters and with the cross-entropy criterion, using asynchronous stochastic gradient descent (ASGD) [28]. The sequence-training experiments in this paper also use distributed ASGD, which is outlined in more details in [29]. All networks have 13,522 CD output targets. The weights for CNN and DNN layers are initialized using the Glorot-Bengio strategy described in [30], while all LSTM parameters are uniformly initialized to lie between -0.02 and 0.02. We use an exponentially decaying learning rate, which starts at 4e-3 and has a decay rate of 0.1 over 15 billion frames.

3.2. Filter Prediction Experiments

The baseline NAB model consists of a raw waveform CLDNN AM [4] and a *FP module*, without MTL and feedback. The *FP module* has two 512-cell LSTM layers and one linear output layer to generate 5 ms filter coefficients (i.e. 81 taps at 16kHz sampling rate) per input channel. This gives a word error rate (WER) of 22.2%, while the single-channel raw wavform CLDNN is at 23.5% [3]. In the following subsections, we describe experiments using variations of this baseline to find the best FP setup.

3.2.1. Architecture

First, we explore different architectures for the *FP module* (Figure 1). Each *FP module* has first S "shared" 512-cell LSTM layers, followed by a split stack of P "splitted" channel-dependent 256-cell LSTM layers, to encourage learning an independent filter prediction model for each channel. Channel-dependent linear output layers are then added to generate filter coefficients. The baseline hence has $S = 2$ and $P = 0$.

Table 1 shows the WERs using different *FP module* architectures. The best performance is obtained using one shared and one channel-dependent LSTM layer. Further increasing the total number of LSTM layers does not improve performance, regardless of the configuration.

Total	2			3		
shared (S)	2	1	0	3	2	1
splitted (P)	0	1	2	0	1	2
WER (%)	22.2	**21.8**	22.3	22.4	22.3	22.8

Table 1: WER for different architectures of the *FP module*.

3.2.2. Filter Inputs

Cross-correlation features [21] are often used for localization, and were also adopted in [17] to predict frequency domain beamforming filters. For comparison, we also trained a two channel NAB model passing the unweighted cross correlation features extracted from 100 ms frames with 10 ms shift as inputs to the *FP module*. With the same baseline structure ($S = 2$, $P = 0$), this model gave a WER of 22.3%, which is similar to the 22.2% obtained using waveform samples as inputs. Providing more explicit localization information in the form of cross correlation features does not help, suggesting that the *FP module* is able to learn good spatial filters directly from waveform samples.

3.2.3. Filter Size

The maximum delay between two microphones spaced 14 cm apart is less than 0.5 ms, suggesting that filters no shorter than 0.5 ms should be sufficient to align the two channels. In this section we explore varying the length of predicted filters with the baseline *FP module* ($S = 2$ and $P = 0$). The results are shown in Table 2. The best performance is obtained using a 1.5 ms filter. It can also be seen that making the filter size too large hurts performance.

Size (ms)	1.0	1.5	2.0	3.0	5.0	10.0
Size (samples)	17	25	33	49	81	161
WER (%)	22.1	**22.0**	22.2	22.3	22.2	22.5

Table 2: WER for different beamforming filter sizes.

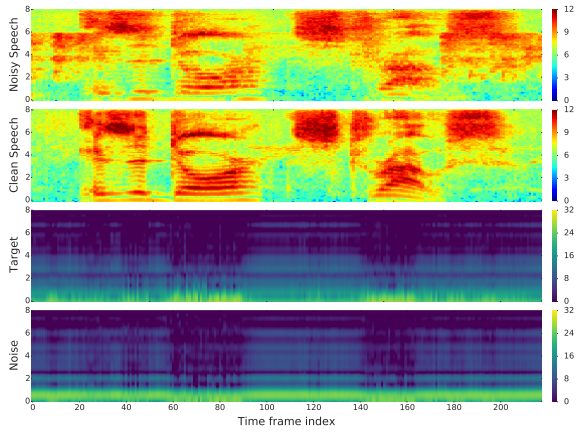

Figure 2: *Visualizations of the predicted beamformer responses at different frequency (Y-axis) across time (X-axis) at the target speech direction (3rd) and interfering noise direction (4th) with the noisy (1st) and clean (2nd) speech spectrograms.*

3.2.4. Multitask Learning and Feedback

The NAB adopts multitask learning to improve robustness by training part of the network to reconstruct 128 dimensional clean log-mel features as a secondary objective to the primary task of CD state prediction. An interpolation weight $\alpha = 0.9$ is used to balance the two objectives. Using MTL, the baseline NAB ($S = 2$, $P = 0$ and 5.0 ms filter) reduces WER from 22.2% to 21.2%. To further improve performance, we add a gated feedback connection described in Section 2.3 which results in another 0.2% absolute reduction to yield a final WER of 21.0%.

3.2.5. Final NAB Setup

A final NAB model with the best configurations is hence built, which has: a) the FP structure of $S = 1$ and $P = 1$; b) raw waveform inputs; c) output filter size of 1.5 ms; d) MTL objective interpolation weight of $\alpha = 0.9$; e) gated feedback connections. Instead of using 128 filters for the spectral filtering layer (`tConv` in Figure 1), we use 256 filters as it has been shown to give further improvements [2]. With the final configurations, the NAB model achieves a WER of 20.5%, a 7.7% relative improvement over the original NAB model at 22.2% without these modifications. Among them, MTL and gated feedback together give the most error reductions. Figure 2 illustrates the frequency responses of the predicted beamforming filters at the target speech and interfering noise direcitons. The SNR for this utterance is 12dB. The responses in the target speech direction have relatively more speech-dependent variations than those in the noise direction. This may indicate that the predicted filters are attending to the speech signal. Besides, the responses at high speech-engergy regions are generally lower than others, which suggests the automatic gain control effect of the predicted filters.

3.3. Comparisons to Other Models

3.3.1. NAB in Frequency Domain

Since adaptive beamforming was first proposed in the frequency domain [17], we compare the NAB model in both time and frequency domains. In the frequency domain NAB setup, we have an LSTM which predicts complex FFT (CFFT) filters for both channels. Given a 257-pt FFT input, this amounts to predicting 4×257 frequency points for real and imaginary components for 2 channels, which is much more than the size in the time domain from Table 2. After the complex filters are predicted for each channel, element-wise product is done with the FFT of the input for each channel, equivalent to the convolution in Equation 2 in the time domain. The output of this is given to a single channel CLDNN in the frequency domain, which does both spectral decomposition with a complex linear projection (CLP) and acoustic modeling. We refer the reader to [31] for more details about the CLP model. Table 3 shows the WER and computational complexity of the raw waveform and CFFT NAB models. While using CFFT features greatly reduces computational complexity, the performance is worse than the raw waveform model. One hypothesis we have is that CFFT requires predicting a higher dimensional filter, which we can see from Table 2 leads to a degradation in performance.

Model	**WER (%)**	**Param (M)**	**MultAdd (M)**
raw	20.5	24.6	35.3
CFFT	21.0	24.7	25.1

Table 3: Comparison between time and frequency NAB models.

3.3.2. Comparison to Other Multichannel Methods

We also compare the performance of the NAB model to the unfactored [2] and factored raw waveform models [3], which have been shown to offer superior performance to single channel models and other signal processing techniques such as DS and FS. Table 4 shows that compared to the unfactored model, predicting filters per time frame to handle different spatial directions in the data helps. Second, while the factored model can potentially handle different directions by enumerating many look directions in the spatial filtering layer, the adaptive model can achieve similar performance with much less computational complexity.

Model	**WER (%)**		**Param**	**MultAdd**
	CE	**Seq.**	**(M)**	**(M)**
unfactored [2]	21.7	17.5	18.9	27.2
factored [3]	20.4	17.1	18.9	35.1
NAB	20.5	17.2	24.0	28.8

Table 4: Comparison between factored and adaptive models.

4. Conclusions

In this paper we have presented a NAB architecture for multichannel waveform signals. This model implements adaptive filter-and-sum beamforming jointly with AM training. Unlike previous work, the beamforming filters adapt to the current input signal and also account for AM's previous predictions through gated feedback connections. To improve the generalization of the model, MTL is adopted to regularize the training. Experimental results show that incorporating explicit FS structure is beneficial and the proposed NAB has similar performance to the factored model [3] but with much lower computational cost. In future work, we will further explore this adaptive architecture in conditions such as moving sources or interfering speakers.

5. Acknowledgements

Thank you to Vincent Vanhoucke for the suggestion on feedback for filter prediction, and to Arun Narayanan and Kean Chin for cross correlation related feature preparations.

6. References

[1] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech Acoustic Modeling from Raw Multichannel Waveforms," in *Proc. ICASSP*. IEEE, 2015, pp. 4624–4628.

[2] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani, and A. Senior, "Speaker Localization and Microphone Spacing Invariant Acoustic Modeling from Raw Multichannel Waveforms," in *Proc. ASRU*. IEEE, 2015, pp. 30–36.

[3] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, and M. Bacchiani, "Factored Spatial and Spectral Multichannel Raw Waveform CLDNNs," in *Proc. ICASSP*. IEEE, 2016, to appear.

[4] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Senior, and O. Vinyals, "Learning the Speech Front-end with Raw Waveform CLDNNs," in *Proc. Interspeech*. ISCA, 2015, pp. 1–5.

[5] G. Hinton, L. Deng, D. Yu, G. 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," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[6] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 4, pp. 745–777, 2014.

[7] P. Swietojanski, A. Ghoshal, and S. Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *Proc. ASRU*. IEEE, 2013, pp. 285–290.

[8] C. Weng, D. Yu, S. Watanabe, and B.-H. F. Juang, "Recurrent deep neural networks for robust speech recognition," in *Proc. ICASSP*. IEEE, 2014, pp. 5532–5536.

[9] M. Brandstein and D. Ward, *Microphone arrays: signal processing techniques and applications*. Springer Science & Business Media, 2013.

[10] J. Benesty, J. Chen, and Y. Huang, *Microphone array signal processing*. Springer Science & Business Media, 2008, vol. 1.

[11] J. H. DiBiase, H. F. Silverman, and M. S. Brandstein, "Robust localization in reverberant rooms," in *Microphone Arrays*. Springer, 2001, pp. 157–180.

[12] E. Warsitz and R. Haeb-Umbach, "Acoustic filter-and-sum beamforming by adaptive principal component analysis," in *Proc. ICASSP*, vol. 4. IEEE, 2005, pp. 797–800.

[13] B. D. Van Veen and K. M. Buckley, "Beamforming: A Versatile Approach to Spatial Filtering," *IEEE ASSP magazine*, vol. 5, no. 2, pp. 4–24, 1988.

[14] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third CHiME'Speech Separation and Recognition Challenge: Dataset, task and baselines," in *Proc. ASRU*. IEEE, 2015, pp. 504–511.

[15] T. Hori, Z. Chen, H. Erdogan, J. R. Hershey, J. Roux, V. Mitra, and S. Watanabe, "The MERL/SRI system for the 3rd CHiME challenge using beamforming, robust feature extraction, and advanced speech recognition," in *Proc. ASRU*. IEEE, 2015, pp. 475–481.

[16] M. L. Seltzer, B. Raj, and R. M. Stern, "Likelihood-maximizing beamforming for robust hands-free speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 12, no. 5, pp. 489–498, 2004.

[17] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M. L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep beamforming networks for multi-channel speech recognition," in *Proc. ICASSP*. IEEE, 2016, to appear.

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

[19] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," *arXiv preprint arXiv:1502.02367*, 2015.

[20] R. Giri, M. L. Seltzer, J. Droppo, and D. Yu, "Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning," in *Proc. ICASSP*. IEEE, 2015, pp. 5014–5018.

[21] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 24, no. 4, pp. 320–327, 1976.

[22] X. Xiao, S. Zhao, X. Zhong, D. L. Jones, E. S. Chng, and H. Li, "A learning-based approach to direction of arrival estimation in noisy and reverberant environments," in *Proc. ICASSP*. IEEE, 2015, pp. 2814–2818.

[23] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-Rank Matrix Factorization for Deep Neural Network Training with High-Dimensional Output Targets," in *Proc. ICASSP*. IEEE, 2013, pp. 6655–6659.

[24] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. NIPS*, 2015, pp. 1171–1179.

[25] Y. Zhang, E. Chuangsuwanich, and J. R. Glass, "Extracting deep neural network bottleneck features using low-rank matrix factorization." in *Proc. ICASSP*, 2014, pp. 185–189.

[26] Z. Chen, S. Watanabe, H. Erdoğan, and J. R. Hershey, "Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks," in *Proc. Interspeech*. ISCA, 2015, pp. 3274–3278.

[27] J. B. Allen and D. A. Berkley, "Image Method for Efficiently Simulation Room-Small Acoustics," *Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, April 1979.

[28] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng, "Large Scale Distributed Deep Networks," in *Proc. NIPS*, 2012, pp. 1223–1231.

[29] G. Heigold, E. McDermott, V. Vanhoucke, A. Senior, and M. Bacchiani, "Asynchronous Stochastic Optimization for Sequence Training of Deep Neural Networks," in *Proc. ICASSP*. IEEE, 2014, pp. 5587–5591.

[30] X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *Proc. AISTATS*, 2010, pp. 249–256.

[31] E. Variani, T. N. Sainath, and I. Shafran, "Complex Linear Projection (CLP): A Discriminative Approach to Joint Feature Extraction and Acoustic Modeling," in *Proc. ICML*, 2016, submitted.