BEAMNET: END-TO-END TRAINING OF A BEAMFORMER-SUPPORTED MULTI-CHANNEL ASR SYSTEM

Jahn Heymann, Lukas Drude, Christoph Boeddeker, Patrick Hanebrink, Reinhold Haeb-Umbach

Paderborn University
Department of Communications Engineering
Paderborn, Germany
{heymann, drude, haeb}@nt.uni-paderborn.de

ABSTRACT

This paper presents an end-to-end training approach for a beamformer-supported multi-channel ASR system. A neural network which estimates masks for a statistically optimum beamformer is jointly trained with a network for acoustic modeling. To update its parameters, we propagate the gradients from the acoustic model all the way through feature extraction and the complex valued beamforming operation. Besides avoiding a mismatch between the front-end and the back-end, this approach also eliminates the need for stereo data, i.e., the parallel availability of clean and noisy versions of the signals. Instead, it can be trained with real noisy multi-channel data only. Also, relying on the signal statistics for beamforming, the approach makes no assumptions on the configuration of the microphone array. We further observe a performance gain through joint training in terms of word error rate in an evaluation of the system on the CHiME 4 dataset.

***Index Terms*—** Robust ASR, Multi-Channel ASR, Acoustic beamforming, Complex backpropagation

1. INTRODUCTION

The classical approach for mutlti-channel Automatic Speech Recognition (ASR) is statistically optimum beamforming. Using optimization criteria such as the maximization of the output SNR or the Minimum Variance Distortionless Response (MVDR) criterion, an enhanced signal can be produced which is then input to an ASR backend.

With the success of deep neural networks for acoustic modeling it has been proposed to train a large network with the multi-channel data at its input to predict the context-dependent phoneme state, thus eliminating an explicit beamforming stage and letting the neural network figure out the best mapping of the multi-channel input to the state posteriors. Variants of this approach include stacking the input signals to obtain a representation in the feature domain (e.g. [1]). Due to the loss of the phase during the preprocessing step these approaches are hardly en par with regular beamforming systems. Others use the raw waveforms directly as input [2, 3, 4]. An undisputed advantage of this approach is that the network is trained with a criterion, such as Cross-Entropy (CE), which is known to be appropriate for ASR. However, a significant drawback is that the computational complexity is enormous and that large amounts of training data are required to achieve good results. Additionally, these models are bound to a certain number of look directions which are learned by the filters.

Recently we have proposed an alternative which is computationally much more parsimonious and independent of the microphone configuration. This approach combines a neural network mask estimator with a Generalized Eigenvalue (GEV) beamformer and achieved very competitive results in the 4-th CHiME challenge [5]. It has, however, a few drawbacks:

1. We need target masks in order to train the mask estimation network. Stereo data or at least clean speech data is required to generate those targets. This data is much more difficult to collect than noisy data and thus may not be available for many applications. This also means that the mask estimator can only be trained using simulated data which may have some mismatch compared to the real test data.

2. The target masks themselves are heuristic to some extent and merely a very distant proxy for the final objective of high word recognition rate. Manual optimization, e.g., of the threshold below which a time-frequency bin is declared to contain noise only, is required to achieve the best results.

3. The beamforming front-end and the acoustic model are completely separate systems and thus optimized separately. We cannot utilize any information from the acoustic model to improve the mask estimator.

In this paper, we are going to overcome those drawbacks by jointly optimizing the front-end and the back-end under a

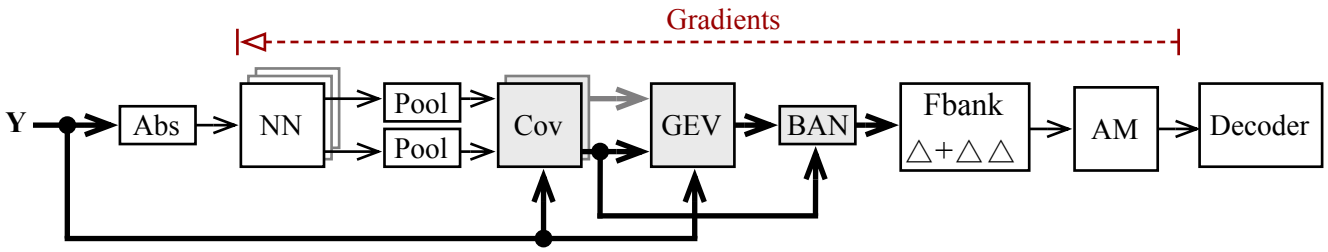

Fig. 1: Overview of the system. Gradients are propagated from the output to the mask estimation network. Bold lines are complex valued signals. Gray blocks operate in the complex domain.

common objective function in an end-to-end training. To this end, we backpropagate gradients from the acoustic model all the way back to the mask estimation stage. While the crucial step of propagating the gradient through the GEV beamformer is detailed in a companion paper [6], this paper focuses on describing the overall processing chain and showing the effectiveness of the approach in terms of recognition performance.

While the idea of optimizing the beamformer w.r.t. an ASR back-end related criterion is not new (e.g. [7]), this is the first to combine statistically optimum beamforming with an end-to-end trained system of neural networks without the need for any additional information like the generalized cross correlation (GCC) [8].

2. MULTI-CHANNEL ASR

Fig. 1 gives an overview of the system considered in this paper. The multi-channel input consists of D microphone signals, to each of which the short-time Discrete Fourier Transform (STFT) is applied. The resulting D components are gathered in a vector $\mathbf{Y}_{f,t}$, where t is the time frame and f the frequency bin index, which consists of a speech component $\mathbf{X}_{f,t}$ and a noise component $\mathbf{N}_{f,t}$:

$$\mathbf{Y}_{f,t} = \mathbf{X}_{f,t} + \mathbf{N}_{f,t}. \tag{1}$$

The goal of the acoustic front-end is to remove, or at least suppress the noise by means of an acoustic beamformer. This is done by multiplying the observed signal with a beamforming vector $\mathbf{w}_f$

$$\hat{S}_{f,t} = \mathbf{w}_f^{\mathrm{H}} \cdot \mathbf{Y}_{f,t}. \tag{2}$$

where $\hat{S}_{f,t}$ is either an estimate of the speech component as observed at a reference microphone (e.g. microphone #1) or an estimate of the speech signal at the signal source, depending on how exactly the beamforming criterion is defined.

Statistically optimum beamformers, such as the MVDR beamformer or the GEV beamformer, need the knowledge of the power spectral density matrices of speech, $\mathbf{\Phi}_{\mathbf{XX}}$, and of noise, $\mathbf{\Phi}_{\mathbf{NN}}$, to compute the beamforming coefficient vector $\mathbf{w}_f$. As depicted in Fig. 1 these Cross-Power Spectral Density (PSD) matrices are computed by placing masks on the input signal, where the masks are estimated by a neural network. The mask estimation is carried out on each channel separately, and the D masks are joined to a single mask by means of mean or median operation.

The back-end operates on the enhanced signal and consists of a feature extraction stage, a neural network to estimate the acoustic model probabilities and the decoder to infer the spoken word sequence.

The goal of this work is to jointly optimize the overall system using a common objective function to achieve best possible ASR performance. The objective function is the commonly used CE between the context-dependent state labels predicted by the acoustic model neural network and the target state labels. In particular we would like to train the front-end neural network for mask estimation with the very same objective function. To be able to do so, we need to compute the gradient of the objective function w.r.t. the parameters of the mask estimator. This requires propagating the gradient through the complete processing chain depicted in Fig. 1.

In the following we discuss the individual processing blocks and the involved computations, starting from the end of the processing chain.

3. ERROR BACKPROPAGATION

3.1. Acoustic Model

Our acoustic model is based on a Wide Residual Network (WRN) [9] and is a smaller version of the one described in detail in [5].

As a trade off between modeling capacity and training time we choose a depth (d) of 10, a width (k) of 5 and dismissed the recurrent layers. The model operates on the whole utterance instead of a window of a few frames. This helps with the Batch-Normalization [5] and makes it easier to integrate the mask estimator which also operates on a whole utterance.

The training of the model is carried out according to standard error backpropagation procedures, and therefore need no further discussion.

3.2. Feature Extraction

Our acoustic model works with 80 dimensional log-mel filterbank features with their delta and delta-deltas. To connect the beamforming model with the acoustic model, we model the feature extraction using basic building blocks of neural networks. To compute the delta and delta-delta features we use a one dimensional convolution layer with filter size 5 and 9 respectively with a corresponding initialization. To apply the filterbank we use a linear layer with no bias and a fixed matrix reassembling the filter banks. For these standard operations the gradient computation is again straightforward.

3.3. Acoustic Beamformer

In earlier work we have shown that the GEV beamformer [10] is particularly suitable for use with an ASR backend, resulting in consistently better recognition results than a MVDR beamformer [11].

Its objective is to maximize the a posteriori signal-to-noise ratio (SNR):

$$\mathbf{w}_f^{\mathrm{GEV}} = \arg\max_{\mathbf{w}_f} \frac{\mathbf{w}_f^{\mathrm{H}} \mathbf{\Phi}_{\mathbf{XX}f} \mathbf{w}_f}{\mathbf{w}_f^{\mathrm{H}} \mathbf{\Phi}_{\mathbf{NN}f} \mathbf{w}_f} \tag{3}$$

Solving (3) leads to the Generalized Eigenvalue problem

$$\mathbf{\Phi}_{XX} \mathbf{W} = \mathbf{\Phi}_{NN} \mathbf{W} \mathbf{\Lambda}, \tag{4}$$

where the desired beamforming vector $\mathbf{w}_f$ is given by the eigenvector corresponding to the largest eigenvalue. $\mathbf{W}$ is a matrix, whose columns are the eigenvectors, and $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues. Since the GEV beamformer can introduce arbitrary distortion, we use Blind Analytic Normalization (BAN) as a post-filter [10].

While the backpropagation of the gradient through the BAN operation is relatively easy, the most crucial step is the derivative of the eigenvalue problem w.r.t. the speech and noise PSDs. Note that the beamforming vector is complex-valued, and thus the complex gradient is given by [12]

$$\nabla_{\mathbf{\Phi}^*} = \left((\nabla_{\mathbf{W}^*})^* \frac{\partial \mathbf{W}}{\partial \mathbf{\Phi}^*} + \nabla_{\mathbf{W}^*} \left(\frac{\partial \mathbf{W}}{\partial \mathbf{\Phi}} \right)^* \right). \tag{5}$$

In a companion paper we have submitted to this conference we have shown that the derivative of some real-valued cost function J w.r.t. $\mathbf{\Phi}^*$ of an Eigenvalue Problem can be expressed as [13] [14] [6]

$$\frac{\partial J}{\partial \mathbf{\Phi}^*} = \mathbf{W}^{-\mathrm{H}} \left[\frac{\partial J}{\partial \mathbf{\Lambda}^*} + \mathbf{F}^* \circ \mathbf{W}^{\mathrm{H}} \frac{\partial J}{\partial \mathbf{W}^*} \right] \mathbf{W}^{\mathrm{H}}. \tag{6}$$

This equation holds if subsequent calculations do not depend on the magnitude of the eigenvectors and if $\mathbf{\Phi}$ is hermitian. For the GEV beamformer however, we have $\mathbf{\Phi} = \mathbf{\Phi}_{\mathbf{NN}}^{-1} \mathbf{\Phi}_{\mathbf{XX}}$ and $\mathbf{\Phi}$ is not hermitian. To solve this problem we normalize the eigenvectors to have a magnitude of one. This removes the degree of freedom from the eigendecomposition. Including this normalization results in the following gradient:

$$\frac{\partial J}{\partial \mathbf{\Phi}^*} = \mathbf{W}^{-\mathrm{H}} \left(\frac{\partial J}{\partial \mathbf{\Lambda}^*} + \mathbf{F}^* \circ \left(\mathbf{W}^{\mathrm{H}} \frac{\partial J}{\partial \mathbf{W}^*} \right) \right) \mathbf{W}^{\mathrm{H}}$$
$$- \mathbf{W}^{-\mathrm{H}} \left(\mathbf{F}^* \circ \mathbf{W}^{\mathrm{H}} \mathbf{W} \left(\mathrm{Re} \left\{ \mathbf{W}^{\mathrm{H}} \frac{\partial J}{\partial \mathbf{W}^*} \right\} \circ \mathbf{I} \right) \right) \mathbf{W}^{\mathrm{H}}.$$

For a complete derivation we again refer the reader to [6] and to our technical report [14].

3.4. PSD Computation

We estimate the covariance matrices in Eq. 4 using a masking based approach where the masks $M_{f,t}^{\nu}$ are estimated by a neural network and $\nu \in \{\mathbf{X}, \mathbf{N}\}$:

$$\mathbf{\Phi}_{\nu\nu f} = \sum_{t=1}^{T} M_{f,t}^{\nu} \mathbf{Y}_{f,t} \mathbf{Y}_{f,t}^{\mathrm{H}}. \tag{7}$$

The computation of the derivative of the PSD matrices w.r.t. the masks is straightforward.

3.5. Mask estimation

The mask estimator network is the same as in our previous works [11, 15]. It consists of one bi-directional Long Short-Term Memory (BLSTM) layer and three feed-forward layers.

The estimator outputs the masks for the target as well as the one for the noise given the magnitude spectrum of one microphone at its input. Each microphone is treated independently but with the same network parameters. This allows us to stay independent of the microphone configuration. The beamforming operation works better when the same mask is used for each channel [16]. To condense the masks into one, we use median pooling during decoding and mean during training. The median is resistant to a channel failure, but its gradient is sparse and not always well defined which lead us to use the mean at training time. This also more closely reassembles our previous approach where each channel gets a gradient from an ideal binary mask.

One major difference compared to our previous contributions are the different parameters used for the Short Time Fourier Transform (STFT) transformation. Instead of using a window size of 1024 and a shift of 256 we use a window size of 400 and a shift of 160. These parameters are common for speech recognition, so choosing them avoids transformations between the beamformer and the acoustic model. Preliminary experiments showed that the different transformation does not have an impact on the performance.

4. EXPERIMENTS

4.1. Database

The dataset from the 4th CHiME challenge [17] is used for all of our experiments. It features real and simulated audio data of prompts taken from the 5k WSJ0-Corpus [18] with 4 different types of real-world background noise. We only consider the multi-channel track with six channels here.

4.2. Setups

The 4th CHiME challenge provides a baseline system which uses BeamformIt! [19] in the front-end, a DNN-HMM acoustic model trained with sMBR and a combination of a 5-gram Kneser-Ney and recurrent neural network language model [17] (BFIT+Kaldi). Alignments from this system are used for all subsequent trainings. The decoding pipeline is the same for all experiments. We train the WRN acoustic model on all six noisy channels to replace the DNN-HMM model and a mask estimator with ideal binary masks as described in [11] and replace BeamformIt! with the GEV beamformer. These three results serve as a baseline.

We aim to answer the following questions: Can end-to-end training reduce the mismatch of a combined system? Can we train a mask estimator without parallel clean and noisy data? And can we even train the system from scratch? To answer these questions, we vary which component we initialize randomly (scratch) and which we initialize with the respective pre-trained model (finetune/fixed). We then train the system using the backpropagation described in the previous section.

For training we use ADAM [20] with $\alpha = 10^{-5}$. Dropout with $p = 0.5$ and an L2 regularization of 10^{-6} is used in each layer. We also employ Batch-Normalization [21] in each layer. This helped to improve performance as well as convergence speed in our previous works[1].

4.3. Results

The results of our experiments are displayed in Tab. 1. They show that our down-sized acoustic model performs as good as the baseline acoustic model and even somewhat better on the real test set (2nd results line). Replacing BeamformIt! with the GEV beamformer with a pre-trained mask estimator ("fixed") leads to a significant gain (3rd results line).

Simultaneously finetuning the mask estimator and the acoustic model provides the best overall performance (last results line). The gain compared to just finetuning the acoustic model on the beamformed data is small (2nd last to last line). This shows that the mismatch between the front-end trained on simulated data and the back end finetuned on real noisy recordings is small for this dataset, because not much is gained by finetuning the mask estimator on the real data.

[1] Due to computational limitations we were unable to do an extensive hyperparameter search and thus relied on experience from previous works.

Table 1: Average WER (%) for the described systems.

Training		Dev		Test	
BF	AM	real	simu	real	simu
BFIT+Kaldi		5.76	6.77	11.51	10.90
BFIT+WRN		5.53	6.67	9.44	10.18
fixed	fixed	4.26	4.29	5.85	4.59
scratch	scratch	5.51	5.19	8.76	5.61
scratch	finetune	4.14	4.09	5.86	4.06
fixed	finetune	4.09	3.96	5.56	**3.9**
finetune	finetune	**3.77**	**3.89**	**5.42**	3.95

The table also shows that if we initialize the mask estimator randomly we can get slightly better results than by just combining both pre-trained models. This result, which can be found on the 3rd to last line of the table, is the most important outcome of this study, because it shows that we indeed were able to eliminate the need for any parallel clean and noisy data for mask estimation and achieve even slightly better performance by the proposed end-to-end training.

If we train the whole model completely from scratch the results get worse. We see two reasons for this. First, the hyper-parameters might not be optimal for this setting as jointly learning to classify the state posteriors and the mask estimation for an optimal look direction is a hard task for the model. Second, the amount of training data for the acoustic model is only one sixth of the data compared to using each channel separately. This has already been shown to lead to decreased performance [22]. Nevertheless, this model still performs better than the baseline model or the pre-trained acoustic model combined with BeamformIt!.

5. CONCLUSION & OUTLOOK

This work describes a system where the beamformer front-end is jointly trained with the acoustic model using the CE criterion. Relying on statistical beamforming, this system is independent of the array geometry. We show that such a system is able to further improve performance compared to just combining both components without joint training. Most importantly it eliminates the need for parallel clean and noisy data as well as heuristic hand-tuned masks to train the mask estimator. In future work we will focus on improving the performance of the model trained from scratch.

6. ACKNOWLEDGMENTS

This research was supported by the Deutsche Forschungsgemeinschaft (DFG) under Contract No. Ha3455/11-1. Computational resources were provided by the Paderborn Center for Parallel Computing.

7. REFERENCES

[1] P. Swietojanski, A. Ghoshal, and S. Renals, "Convolutional Neural Networks for Distant Speech Recognition," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1120–1124, Sept 2014.

[2] B. Li, T. Sainath, R. Weiss, K. Wilson, and M. Bacchiani, "Neural Network Adaptive Beamforming for Robust Multichannel Speech Recognition," in *Proc. Interspeech*, 2016.

[3] T. Sainath, R. Weiss, K. Wilson, A. Narayanan, M. Bacchiani, and A. Senior, "Speaker Location and Microphone Spacing Invariant Acoustic Modeling from Raw Multichannel Waveforms," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec 2015, pp. 30–36.

[4] T. Sainath, R. Weiss, K. Wilson, A. Narayanan, and M. Bacchiani, "Factored Spatial and Spectral Multichannel Raw Waveform CLDNNs," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[5] J. Heymann, L. Drude, and R. Haeb-Umbach, "Wide Residual BLSTM Network with Discriminative Speaker Adaptation for Robust Speech Recognition," in *Computer Speech and Language*, 2016, to appear.

[6] C. Boeddeker, P. Hanebrink, L. Drude, J. Heymann, and R. Haeb-Umbach, "Optimizing Neural-Network Supported Acoustic Beamforming by Algorithmic Differentiation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[7] M. L. Seltzer, B. Raj, and R. M. Stern, "Likelihood-Maximizing Beamforming for Robust Hands-Free Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 12, 2004.

[8] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep Beamforming Networks for Multi-Channel Speech Recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[9] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," *CoRR*, vol. abs/1605.07146, 2016.

[10] E. Warsitz and R. Haeb-Umbach, "Blind Acoustic Beamforming based on Generalized Eigenvalue Decomposition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, 2007.

[11] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural Network Based Spectral Mask Estimation for Acoustic Beamforming," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[12] L. Drude, B. Raj, and R. Haeb-Umbach, "On the Appropriateness of Complex-Valued Neural Networks for Speech Enhancement," in *Proc. Interspeech*, 2016.

[13] M. Giles, "An Extended Collection of Matrix Derivative Results for Forward and Reverse Mode Automatic Differentiation," 2008.

[14] C. Boeddeker, P. Hanebrink, L. Drude, J. Heymann, and R. Haeb-Umbach, "On the Computation of Complex-valued Gradients with Application to Statistically Optimum Beamforming," *arXiv:1701.00392 [cs.NA]*, 2017.

[15] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, "BLSTM supported GEV Beamformer Front-End for the 3rd CHiME Challenge," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015.

[16] H. Erdogan, J. Hershey, S. Watanabe, M. Mandel, and J. Le Roux, "Improved MVDR Beamforming using Single-Channel Mask Prediction Networks," in *Proc. Interspeech*, 2016.

[17] E. Vincent, S. Watanabe, A. Nugraha, J. Barker, and R. Marxer, "An Analysis of Environment, Microphone and Data Simulation Mismatches in Robust Speech Recognition," in *Computer Speech and Language*, 2016, to appear.

[18] J. Garofalo et al., "CSR-I (WSJ0) complete," 2007.

[19] X. Anguera, C. Wooters, and J. Hernando, "Acoustic Beamforming for Speaker Diarization of Meetings," vol. 15, 2007.

[20] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[21] Sergey I. and Christian S., "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *CoRR*, vol. abs/1502.03167, 2015.

[22] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi, S. Araki, and T. Nakatani, "The NTT CHiME-3 System: Advances in Speech Enhancement and Recognition for Mobile Multi-Microphone Devices," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015.