JOINT OPTIMIZATION OF NEURAL NETWORK-BASED WPE DEREVERBERATION AND ACOUSTIC MODEL FOR ROBUST ONLINE ASR

Jahn Heymann[1], Lukas Drude[1], Reinhold Haeb-Umbach[1], Keisuke Kinoshita[2], Tomohiro Nakatani[2]

[1]Paderborn University, Department of Communications Engineering, Paderborn, Germany
[2]NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan
[1]{heymann, drude, haeb}@nt.upb.de
[2]{keisuke.kinoshita, tnak}@ieee.org

ABSTRACT

Signal dereverberation using the Weighted Prediction Error (WPE) method has been proven to be an effective means to raise the accuracy of far-field speech recognition. First proposed as an iterative algorithm, follow-up works have reformulated it as a recursive least squares algorithm and therefore enabled its use in online applications. For this algorithm, the estimation of the power spectral density (PSD) of the anechoic signal plays an important role and strongly influences its performance. Recently, we showed that using a neural network PSD estimator leads to improved performance for online automatic speech recognition. This, however, comes at a price. To train the network, we require parallel data, i.e., utterances simultaneously available in clean and reverberated form. Here we propose to overcome this limitation by training the network jointly with the acoustic model of the speech recognizer. To be specific, the gradients computed from the cross-entropy loss between the target senone sequence and the acoustic model network output is backpropagated through the complex-valued dereverberation filter estimation to the neural network for PSD estimation. Evaluation on two databases demonstrates improved performance for online processing scenarios while imposing fewer requirements on the available training data and thus widening the range of applications.

***Index Terms*—** dereverberation, speech enhancement, joint optimization, robust ASR

1. INTRODUCTION

Reverberation has a severe impact on the intelligibility of a speech signal and, despite all recent advances in acoustic modeling, still deteriorates the performance of automatic speech recognition (ASR) systems significantly, even when trained on large scale data [1, 2]. In these challenging far-field scenarios, signal processing to dereverberate and thus enhance the signal can help to mitigate the performance losses. Many techniques have been proposed for signal dereverberation, which can be broadly categorized in linear filtering approaches and spectral subtraction like approaches for magnitude or power spectrum manipulation [3].

Weighted prediction error (WPE) dereverberates the signal by estimating an inverse filter which is used to subtract the reverberation tail from the observation and thus falls into the first category. It can operate either on a single channel or in a multiple-input multiple-output fashion on multi-channel data. The quality of the estimated filter coefficients mainly depends on the estimation of the PSD of the "anechoic speech", i.e., the direct speech signal and its early reflections. Since this signal is unknown, the vanilla WPE works iteratively by alternating between two steps: (Step 1) Dereverberating the signal using the current estimate of the anechoic speech PSD, and, (Step 2) estimating the anechoic speech PSD using the current estimate of the dereverberated signal. Alternating these two steps gradually improves the estimate of both, the (dereverberated) target signal and the anechoic speech PSD. This, however, inherently makes the vanilla WPE an offline method and computationally expensive.

To overcome this dependency issue – and enable an online usage of WPE – we recently proposed to utilize a neural network to directly estimate the PSD from the observation [4] [5]. We could show that this leads to improved performance for low-latency solutions compared to a more simple PSD estimation [6]. However, we now need parallel reverberated and non-reverberated data in order to train the PSD estimation network, limiting the applicability of the approach.

In this work, we lift this restriction by combining the WPE front-end with the acoustic model already during training. This allows us to train the estimator directly with a suitable ASR loss. Apart from an expected performance improvement like we saw with e.g. beamforming [7], the motivation for this is threefold:

1. Calculating a training target for the PSD estimator requires a corpus of parallel data. Building such a corpus is almost only possible by simulating the observed data, inevitably leading to a mismatch between training and test data.

2. There is no clear notion which part of the signal should be considered as anechoic or the target signal respectively. Thus, the training target for the PSD estimator is not well-defined. Where as the network can adjust to the acoustic model needs when trained jointly.

3. If other directed noise sources are present, their contribution to the covariance statistics could be optimized by an appropriate weighting factor for the tf-bins in question.

We investigate the performance of such a joint system and compare it with one separately trained on oracle PSD information.

2. SCENARIO AND SIGNAL MODEL

Using D microphones, we observe a signal which is represented as the D-dimensional vector $\mathbf{y}_{t,f}$ at time frame index t and frequency bin index f in the short time Fourier transformation (STFT) domain. In a far-field scenario, this signal is impaired by (convolutive) reverberation. We assume, that for ASR the early part of the room impulse response (RIR) is beneficial whereas the reverberation tail deteriorates the recognition and should therefore be suppressed. Specifically, we consider the first $50\,\mathrm{ms}$ after the main peak of the RIR ($h^{(\mathrm{early})}$) to contribute to the anechoic signal whereas the remaining part ($h^{(\mathrm{tail})}$) is assumed to cause the distortions. In the STFT domain we denote this model as follows:

$$\mathbf{y}_{t,f} = \mathbf{x}_{t,f}^{(\mathrm{early})} + \mathbf{x}_{t,f}^{(\mathrm{tail})}, \tag{1}$$

where $\mathbf{x}_{t,f}^{(\mathrm{early})}$ and $\mathbf{x}_{t,f}^{(\mathrm{tail})}$ are the STFTs of the source signal convolved with the early part of the RIR and with the late reflections, respectively. Note that we explicitly allow RIRs longer than the length of a DFT window.

3. WEIGHTED PREDICTION ERROR

WPE estimates the reverberation tail of the signal from previous samples and subtracts it from the observation to obtain an optimal estimate of the anechoic speech in a maximum likelihood sense:

$$\hat{x}_{t,f,d}^{(\mathrm{early})} = y_{t,f,d} - \mathbf{g}_{f,d}^{\mathsf{H}} \tilde{\mathbf{y}}_{t-\Delta,f}, \tag{2}$$

where $\mathbf{g}_{f,d}$ and $\tilde{\mathbf{y}}_{t-\Delta,f}$ are stacked representations of the filter taps and the observation respectively and d is the filter index.

Using $\Delta \geq 1$ avoids whitening of the speech source. WPE maximizes the likelihood of the model under the assumption that the anechoic signal is a realization of a zero-mean circularly-symmetric complex Gaussian with an unknown time-varying variance $\lambda_{t,f}$.

3.1. Iterative WPE

There is no closed form solution for the likelihood optimization, but an iterative procedure which alternates between estimating the filter coefficients $\mathbf{g}_{fd}$ and the time-varying variance λ_{tf} exists:

$$\text{Step 1)} \quad \mathbf{R}_f = \sum_t \frac{\tilde{\mathbf{y}}_{t-\Delta,f} \tilde{\mathbf{y}}_{t-\Delta,f}^{\mathsf{H}}}{\lambda_{t,f}}, \tag{3}$$

$$\mathbf{p}_{f,d} = \sum_t \frac{\tilde{\mathbf{y}}_{t-\Delta,f} y_{t,f,d}^*}{\lambda_{t,f}}, \tag{4}$$

$$\mathbf{g}_{f,d} = \mathbf{R}_f^{-1} \mathbf{p}_{f,d} \tag{5}$$

$$\text{Step 2)} \quad \lambda_{t,f} = \frac{1}{(\delta + 1 + \delta)\, D} \sum_{\tau=t-\delta}^{t+\delta} \sum_d |\hat{x}_{\tau,f,d}^{(\mathrm{early})}|^2. \tag{6}$$

The heuristic context of $(\delta + 1 + \delta)$ frames helps to improve the variance estimate in this iterative scheme [8].

3.2. Recursive WPE

To derive a recursive formulation, the correlation matrix is estimated with a decaying window:

$$\mathbf{R}_{t,f} = \sum_{\tau=0}^{t} \alpha^{t-\tau} \frac{\tilde{\mathbf{y}}_{\tau-\Delta,f} \tilde{\mathbf{y}}_{\tau-\Delta,f}^{\mathsf{H}}}{\lambda_{\tau,f}}. \tag{7}$$

This leads to a recursive solution with the following rank-one updates [9]:

$$\mathbf{K}_{t,f} = \frac{\mathbf{R}_{t-1,f}^{-1} \tilde{\mathbf{y}}_{t-\Delta,f}}{\alpha \lambda_{t,f} + \tilde{\mathbf{y}}_{t-\Delta,f}^{\mathsf{H}} \mathbf{R}_{t-1,f}^{-1} \tilde{\mathbf{y}}_{t-\Delta,f}} \tag{8}$$

$$\mathbf{R}_{t,f}^{-1} = \frac{1}{\alpha} \left(\mathbf{R}_{t-1,f}^{-1} - \mathbf{K}_{t,f} \tilde{\mathbf{y}}_{t-\Delta,f}^{\mathsf{H}} \mathbf{R}_{t-1,f}^{-1} \right) \tag{9}$$

$$\mathbf{G}_{t,f} = \mathbf{G}_{t-1,f} + \mathbf{K}_{t,f} \tilde{\mathbf{y}}_{t-\Delta,f}^{\mathsf{H}}. \tag{10}$$

Here, $\mathbf{G}_{t,f}$ are the stacked filter taps $\mathbf{g}_{f,d}$ for each microphone. Note that these are now time variant. This is in essence a *Recursive Least Squares* (RLS) adaptive filter for the reverberation estimation. The authors of [6] approximate the PSD of the target signal using a smoothed PSD of the observation averaged over the microphones using a left and right context δ_{L} and δ_{R}:

$$\lambda_{t,f} = \frac{1}{D} \cdot \frac{1}{\delta_{\mathrm{L}} + 1 + \delta_{\mathrm{R}}} \sum_{\tau=t-\delta_{\mathrm{L}}}^{t+\delta_{\mathrm{R}}} \sum_d |y_{\tau,f,d}|^2. \tag{11}$$

4. PROPOSED FRAMEWORK

4.1. PSD estimation

Given the statistics $\lambda_{t,f}$ of the underlying anechoic signal, the optimal filter coefficients for WPE can be calculated in closed

form with Eq. 5 or adaptively with Eq. 10. But since we can only observe the reverberant signal, these statistics have to be estimated. Consistently with [4] and [5], we focus on a deep neural network (DNN) for PSD estimation.

In particular, we use the same network architecture as in the works above. The network consists of a long short-term memory (LSTM) layer with 512 units, two linear layers with 2048 units and ReLU activation functions and a final linear layer with 513 units. It operates on a single channel and the final estimate is obtained by averaging over all channels making it independent of the number of channels and the microphone configuration.

As a baseline, we consider estimating $\lambda_{t,f}$ by a comparably simple smoothing of the spectrum as specified by Eq. 11 which has also shown good performance in [6] and [5] (with $\delta_{\mathrm{L}} = 1$ and $\delta_{\mathrm{R}} = 0$).

4.2. Acoustic model

Our acoustic model is a wide bi-directional residual network (WBRN) as proposed in [5]. It consists of several convolutional layers with residual connections, followed by two BLSTM layers and two linear layers. The hyperparameters as well as the initial training procedure were adapted from [10]. The model is trained on frame-wise senone targets and shows very competitive performance on the task at hand. Note that the acoustic model itself operates offline since we focus on the effects of the front-end but can be replaced by an online version to achieve a fully online operating system.

4.3. Training

The acoustic model is first trained using multi-condition data of the respective corpus until convergence. For the DNN based PSD estimator, we train different variants.

The first one (A1) is our baseline and we train the PSD estimator separately as described in [5] and [4]. The anechoic speech PSD is used as the target for a mean-squared-error (MSE) training.

For the second one (B1), we utilize the acoustic model loss to finetune the estimator trained separately. This model still needs the parallel data (for the initialization) but might result in improved overall system performance as we directly optimize for the ASR target.

Third (C1), we train a PSD estimator with random initialization using the state level cross-entropy (CE) loss but keep all the parameters of the acoustic model fixed, i.e. use it as a loss function w.r.t. the PSD estimator.

Note that we backpropagate either through the offline equations (Eq. 3 – Eq. 5) or the recursive formulation (Eq. 8 – Eq. 10) respectively. Because the backpropagation through the online variant needs a lot of memory, this calculation always runs on the CPU where we can utilize the system memory. Since it is also computationally very expensive, we do not train it directly form scratch but rather first train the offline system from scratch and then switch to the online variant after an initial training phase.

Finally, we finetune the acoustic model for all variants with a learning rate of 10^{-5} using the respective WPE front-end. These systems are referred to as (A2) – (C2), each of which corresponds to the variant (A1) – (C1). If applicable, we jointly optimize both models in this step. Otherwise, just the acoustic model is finetuned. To increase the diversity of the training data, we sample the delay Δ to be in a range between 1 and 4 and the number of taps K to be between 5 and 10 during this step.

4.4. Implementation

All models were implemented in Tensorflow r1.10. We use publicly available[1] WPE implementation [11]. For joint training we found it crucial to use $128\,\mathrm{bit}$ (i.e. $64\,\mathrm{bit}$ for the real and imaginary part) for the complex values involved in the calculation of WPE.

5. EVALUATION

To demonstrate the versatility of the described approach, we evaluate the proposed systems in terms of WERs on the data of the REVERB challenge as well as on WSJ+VoiceHome data.

The REVERB challenge dataset [12] contains simulated and real utterances. For simulated data WSJCAM0 utterances [13] are convolved with measured RIRs. Noise is added with $\sim$20 dB signal to noise ratio (SNR). Reverberation times (T60) are in the range of 0.25 – $0.7\,\mathrm{s}$. The real data consists of utterances from the MC-WSJ-AV corpus [14] which are recorded in a noisy reverberant room with a reverberation time of $\sim 0.7\,\mathrm{s}$. The corpus is known for its mismatch between the simulated data used during training and the real recordings for evaluation. To reduce this discrepancy, we randomly sample the SNR to be in the range of $5\,\mathrm{dB}$ – $30\,\mathrm{dB}$ and scale the signal with 0.2 for the training of the PSD estimators and all finetuning experiments [4]. The initial acoustic model is trained on unscaled data without SNR perturbation.

For WSJ+VoiceHome we convolve WSJ utterances ($5\,\mathrm{k}$ vocabulary) with VoiceHome RIRs and VoiceHome background noise [15] with reverberation times (T60) in the range of 395 – $585\,\mathrm{ms}$. This is similar to the simulation setup proposed by Bertin et al. [16]. The VoiceHome background noise is very dynamic and typically found in households e.g. vacuum cleaner, dish washing or interviews on television and the SNR ranges from $0\,\mathrm{dB}$ – $10\,\mathrm{dB}$.

We evaluate the performance for two and eight microphone channels. Since WPE preserves the number of channels, we always take the first one and use it for decoding with

[1] https://www.github.com/fgnt/nara_wpe

Table 1. WERs/% for all systems evaluated on REVERB (*real* data eval set, averaged over near and far) and WSJ+VoiceHome. For the MSE loss parallel data is needed to calculate the anechoic PSD target, while the CE loss uses senone targets.

	Model	Joint	PSD		AM	Reverb				WSJ+VoiceHome			
						Offline		Online		Offline		Online	
			Init	Loss		2 ch	8 ch	2 ch	8 ch	2 ch	8 ch	2 ch	8 ch
	Unprocessed	–	–	–	–	17.6				24.3			
	Iteration	–	–	–	finetune	14.4	10.9	–	–	18.7	**17.2**	–	–
	Smooth	–	–	–	finetune	16.1	13.0	17.4	16.2	20.3	18.6	20.9	20.0
(A1)	DNN	–	scratch	MSE	fix	16.1	13.0	18.3	17.6	22.1	20.9	23.7	22.5
(A2)	DNN	–	scratch	MSE	finetune	**14.3**	**10.8**	15.6	14.6	18.9	18.1	19.8	19.3
(B1)	DNN	X	(A1)	CE	fix	15.4	11.9	16.8	14.6	19.3	18.5	20.4	19.7
(B2)	DNN	X	(A1)	CE	finetune	15.1	11.8	**15.0**	**13.4**	**18.4**	17.7	19.2	**18.4**
(C1)	DNN	X	scratch	CE	fix	15.2	12.1	16.9	14.5	19.4	18.5	20.2	19.4
(C2)	DNN	X	(C1)	CE	finetune	14.6	11.8	15.4	13.8	18.5	17.6	**19.1**	**18.4**

the acoustic model. For decoding, we use the 3-gram language model from the WSJ0 corpus without any rescoring afterwards.

For WPE we use a DFT window size of 512 (32 ms) and a shift of 128 (8 ms). For the recursive WPE variant, we set $\alpha = 0.9999$ and for vanilla WPE the number of iteration to 3. For all variants, we vary the delay parameter in a range between 1 and 4 and the number of filter taps is set to either 5 or 10. These values are determined on the development set and can be different for each configuration.

Our baselines for evaluation are *Unprocessed*, *Iteration* and *Smooth*. These use the first channel without any enhancement, vanilla WPE iterations and the smoothing PSD estimator (see. Eq. 11) respectively. These are compared with the neural network variants trained as described in Subsec. 4.3.

These systems are evaluated for two different latency constraints (where applicable): offline and online. Offline means, that the whole utterance is available for processing and this is our baseline scenario. For the online setting, which is our target scenario, we use the recursive formulation of WPE (Eq. 8 – Eq. 10) and the system operates on a frame-by-frame basis.

All results are shown in Tbl. 1. First note, that the unprocessed baseline itself already achieves very good performance. For comparison, the recently updated Kaldi system achieves a WER of around 19.7 % [2] on the Reverb dataset. All fine-tuned systems improve upon the unprocessed baseline irrespective of the PSD estimator or the latency constraint showing the effectiveness of a WPE front-end. As another general tendency we can see that the DNN supported systems outperform the baselines, especially in the online case we were focusing on. For the offline use-case, using the vanilla WPE formulation with iterations seems to be most suitable though, especially when considering the system complexity.

For the online use-case however, using a DNN-based PSD estimator and training it with the ASR criterion is more effective. One reason for this might be that the PSD can be implicitly tuned for faster convergence of the filter estimation in this case. The best results are achieved if the PSD estimator is pre-trained using parallel data and then finetuned jointly with the acoustic model. If no parallel data is available, the PSD estimator can also be trained from scratch with little to no loss in performance. All this applies to both tested corpora and therefore for highly reverberant as well as reverberant and noisy household-like scenarios.

6. CONCLUSIONS

In this paper we demonstrate that jointly optimizing a DNN based PSD estimator and the acoustic model improves the performance for online dereverberation with WPE by 8 % – 18 % in highly reverberant as well as noisy reverberant scenarios compared to a baseline smoothing PSD estimator. Being able to backpropagate through the filter estimation further lifts the requirement for parallel training data and further allows to potentially train the PSD estimator on real data to reduce the mismatch between training and inference with minimal impact on the overall performance.

7. ACKNOWLEDGEMENTS

This work was in part supported by a Google Faculty Research Award. Computational resources were provided by the Paderborn Center for Parallel Computing.

[2] https://github.com/kaldi-asr/kaldi/blob/64d5cf269321883c4031cf12a62374e01acd4b51/egs/reverb/s5/RESULTS#L165 mean of near and far conditions for real data

8. REFERENCES

[1] B. Li, T. Sainath, A. Narayanan, J. Caroselli, M. Bacchiani, A. Misra, I. Shafran, H. Sak, G. Pundak, K. Chin, et al., "Acoustic modeling for Google home," 2017.

[2] M. Harper, "The automatic speech recognition in reverberant environments (ASpIRE) challenge," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015.

[3] J. Li, L. Deng, R. Haeb-Umbach, and Y. Gong, *Robust Automatic Speech Recognition – a Bridge to Practical Applications*, Elsevier, 2015.

[4] K. Kinoshita, M. Delcroix, H. Kwon, T. Mori, and T. Nakatani, "Neural network-based spectrum estimation for online WPE dereverberation," *Interspeech*, 2017.

[5] J. Heymann, L. Drude, R. Haeb-Umbach, K. Kinoshita, and T. Nakatani, "Frame-online DNN-WPE dereverberation," in *International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2018.

[6] J. Caroselli, I. Shafran, A. Narayanan, and R. Rose, "Adaptive multichannel dereverberation for automatic speech recognition," in *Interspeech*, 2017.

[7] J. Heymann, L. Drude, C. Boeddeker, P. Hanebrink, and R. Haeb-Umbach, "Beamnet: End-to-end training of a beamformer-supported multi-channel asr system," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[8] T. Yoshioka and T. Nakatani, "Generalization of multichannel linear prediction methods for blind MIMO impulse response shortening," *IEEE Transactions on Audio, Speech, and Language Processing*, 2012.

[9] T. Yoshioka, H. Tachibana, T. Nakatani, and M. Miyoshi, "Adaptive dereverberation of speech signals with speaker-position change detection," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009.

[10] J. Heymann, L. Drude, and R. Haeb-Umbach, "Wide residual BLSTM network with discriminative speaker adaptation for robust speech recognition," in *CHiME-4 workshop*, 2016.

[11] Lukas Drude, Jahn Heymann, Christoph Boeddeker, and Reinhold Haeb-Umbach, "NARA-WPE: A Python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing," in *13. ITG Fachtagung Sprachkommunikation (ITG 2018)*, 2018.

[12] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, A. Sehr, W. Kellermann, and R. Maas, "The REVERB challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2013.

[13] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, "WSJCAM0: a British English speech corpus for large vocabulary continuous speech recognition," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1995.

[14] M. Lincoln, I. McCowan, J. Vepa, and H. K. Maganti, "The multi-channel Wall Street Journal audio visual corpus (MC-WSJ-AV): Specification and initial experiments," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2005.

[15] S. Sivasankaran, E. Vincent, and I. Illina, "A combined evaluation of established and new approaches for speech recognition in varied reverberation conditions," *Computer Speech & Language*, 2017.

[16] N. Bertin, E. Camberlein, E. Vincent, R. Lebarbenchon, S. Peillon, E. Lamand, S. Sivasankaran, F. Bimbot, I. Illina, A. Tom, S. Fleury, and E. Jamet, "A french corpus for distant-microphone speech processing in real homes," in *Interspeech*, 2016.