UNSUPERVISED ADAPTATION OF A DENOISING AUTOENCODER BY BAYESIAN FEATURE ENHANCEMENT FOR REVERBERANT ASR UNDER MISMATCH CONDITIONS

Jahn Heymann[*], *Reinhold Haeb-Umbach*[*]

University of Paderborn
Department of Communications Engineering
Paderborn, Germany

Pavel Golik[*], *Ralf Schlüter*[*]

RWTH Aachen University
Human Language Technology and Pattern Recognition
Computer Science Department
Aachen, Germany

ABSTRACT

The parametric Bayesian Feature Enhancement (BFE) and a data-driven Denoising Autoencoder (DA) both bring performance gains in severe single-channel speech recognition conditions. The first can be adjusted to different conditions by an appropriate parameter setting, while the latter needs to be trained on conditions similar to the ones expected at decoding time, making it vulnerable to a mismatch between training and test conditions. We use a DNN backend and study reverberant ASR under three types of mismatch conditions: different room reverberation times, different speaker to microphone distances and the difference between artificially reverberated data and the recordings in a reverberant environment. We show that for these mismatch conditions BFE can provide the targets for a DA. This unsupervised adaptation provides a performance gain over the direct use of BFE and even enables to compensate for the mismatch of real and simulated reverberant data.

Index Terms— robust speech recognition, deep neuronal networks, feature enhancement, denoising autoencoder

1. INTRODUCTION

Over the last few years, the usage of speech recognition in consumer electronics changed dramatically. Voice controlled personal assistants and systems demand for hands-free usage with larger distances between the speaker and usually a single microphone. These conditions challenge automatic speech recognition (ASR) systems to become more robust against environmental influences like noise and especially against reverberation effects.

The recently held REVERB challenge [1] gave some insights on how ASR systems can become more robust. It revealed that almost all of the best performing systems employ a Deep Neural Network - Hidden Markov Model (DNN-HMM) acoustic model. Additionally, it showed the effectiveness of data-driven feature enhancement methods like Non-negative Matrix Factorization (NMF) [2], a Denoising Autoencoder (DA) [3] or even a Deep Recurrent Neural Network (RNN) [4]. Other systems used parametric feature enhancement methods, e.g. Cross Transform [5] and the Weighted Prediction Error (WPE) algorithm [6]. But the results show a weakness of current DNN respectively data-driven approaches: As the channel mismatch between the training data and the evaluation data becomes larger, the performance drops significantly. A special case is the performance decrease when going from simulated reverberant data to real reverberant data which is visible throughout all approaches. While a certain performance loss due to a mismatch condition is expected to happen for all systems, the data-driven systems do not yet provide a means to adjust them to new conditions using no or very few data. They have to be trained on conditions similar to the ones at decoding time, while parametric methods can be adjusted to new conditions just by an appropriate parameter setting.

We therefore investigate, if it is possible to use a parametric feature enhancement to adapt a data-driven approach to unseen conditions. For our investigations we focus on single-channel audio and choose the data-driven DA [7] and the parametric Bayesian Feature Enhancement (BFE) [8, 9]. We look at the influence of different mismatch situations on the recognition performance which arise from different room sizes (reverberation time), different distances between the speaker and microphone and differences between simulated reverberant data and actually recorded reverberant data.

In the next section, we describe the models used for feature enhancement. Afterwards, we give an overview over the backend and how it is combined with the two enhancement methods. The dataset is described in Sec. 4. We present the results in Sec. 5 and conclude in Sec. 6. We end by relating this paper to previous work and giving an outlook for further research in Sec. 7

2. FEATURE ENHANCEMENT

2.1. Bayesian Feature Enhancement

In a reverberant environment the discrete-time microphone signal $y(k)$ results from a convolution of the clean speech signal $x(k)$ with the acoustic impulse response (AIR) $h(k)$ of finite length L_h and additional noise $n(k)$

$$y(k) = \sum_{l=0}^{L_h-1} h(l)x(k-l) + n(k). \tag{1}$$

We then estimate the sequence of clean Log Mel Power Spectral Coefficients (LMPSCs) $\mathbf{x}_{1:M}$ from the observed sequence $\mathbf{y}_{1:M}$. The estimation is carried out in a Bayesian way [8, 10]. We introduce a state vector

$$\mathbf{z}_m := \left((\mathbf{x}_m)^T, \ldots, (\mathbf{x}_{m-L_c+1})^T, (\mathbf{n}_m)^T \right) \tag{2}$$

containing the last L_c LMPSCs of the clean speech and the current noise LMPSCs. Its *a posteriori* probability density function (PDF) $p(\mathbf{z}_m|\mathbf{y}_m)$ is computed recursively with a prediction step

$$p(\mathbf{z}_m|\mathbf{y}_{1:m-1}) = \int p(\mathbf{z}_m|\mathbf{z}_{m-1}, \mathbf{y}_{1:m-1}) p(\mathbf{z}_{m-1}|\mathbf{y}_{1:m-1}) \mathrm{d}\mathbf{z}_{m-1}$$

[*]This work has been supported by the DFG under contracts no. Ha3455/11-1 and no. SCHL2043/1-1

and an update step

$$p(\mathbf{z}_m|\mathbf{y}_{1:m}) \propto p(\mathbf{y}_m|\mathbf{z}_m, \mathbf{y}_{1:m-1})p(\mathbf{z}_m|\mathbf{y}_{1:m-1}) \qquad (3)$$

The prediction step requires an *a priori* model $p(\mathbf{z}_m|\mathbf{z}_{m-1}, \mathbf{y}_{1:m-1})$ for the clean speech and noise LMPSCs. We employ a switching linear dynamical model (SLDM) for the speech and assume the noise signal to be a realization of a stationary white Gaussian stochastic process. The update step calls for an observation model $p(\mathbf{y}_m|\mathbf{z}_m, \mathbf{y}_{1:m-1})$ which we choose to be a multivariate Gaussian with time-variant mean vector and covariance matrix.

The observation model relates the LMPSCs of clean speech and noise to the LMPSCs of noisy reverberant speech. As such, it requires a model of the AIR. In [10] we proposed to use the model by Polack [11]. This model assumes the AIR to be a realization of a white Gaussian noise process with an exponentially decaying envelope. Although only a coarse approximation to a real AIR, it has the advantage that it is characterized by a single parameter: The time constant of the exponential decay. This time constant is proportional to the room reverberation time (T_{60}), which is the time it takes until the energy in the tail of the AIR decays to -60 dB of the total energy. It depends on the room properties and is independent of the distance between the speaker and the microphone. The parameter can be estimated blindly from reverberant speech with a precision well below ± 100 ms (e.g. [12][13]) which is sufficient for our model to deliver good results.

Note that the direct signal and early reflections are not modelled well with this coarse model. BFE has been designed with distant speech in mind and it is to be expected that it is not that effective if the distance between the speaker and the microphone becomes small.

For an in-depth description of this approach we refer to [8, 9].

2.2. Denoising Autoencoder

A (stacked) DA is a network with multiple encoding layers, followed by one affine linear decoding layer when dealing with real-valued data like speech features in this case [7].

The encoding layers have the form

$$h_i(\mathbf{z}_i) = \mathrm{s}\left(\mathbf{W}_i\mathbf{z}_i + \mathbf{b}_i\right) \qquad (4)$$

$\mathbf{z}_i$ is the input to the i-th hidden layer, $\mathbf{W}_i$ its weight matrix and $\mathbf{b}_i$ its bias vector. s () is a non-linearity like a *sigmoid*, *tanh* or *ReLU*.

The goal of the DA is to reconstruct clean (speech) features $\mathbf{x}$ from corrupted input features $\tilde{\mathbf{x}}$. Here, the corruption is caused by reverberation and additional background noise. When providing these corrupted features as the input $\mathbf{z}_0$ to the first layer, we want the output $\hat{\mathbf{x}}$ to be highly similar to the clean features. To learn the required mapping between noisy and clean features, a loss function describing the mean squared error $||\hat{\mathbf{x}} - \mathbf{x}||^2$ is minimized during training.

Because this loss function is highly non-linear in the weights of the network, stochastic gradient descent is used to find a solution for this optimization problem. One which generalizes well needs a *good* initialization of the weights $\mathbf{W}_i$. This is achieved with a pre-training. The probably most common approach for this is the one presented by Hinton et al. [14]. In this paper, we use another approach, namely greedy supervised pre-training [15] with the extension of corrupting the input for the trainable layer as described in [16]. The main reason for this choice is that the latter requires no additional generative model and uses the same function as the fine-tuning. Nevertheless, the results are comparable.

We pre-train every layer for 15 epochs with a learning rate of 0.1 while fixing the weights of the already trained layers. During pre-training the clean corpus is used and the input for the trainable layer is corrupted by randomly setting 50% of it to zero. Afterwards we perform fine-tuning with an initial learning rate of 0.1 and a *Newbob* learning strategy. The input is the multicondition and the output the clean training data. Note that we do not apply additional corruption during the fine-tuning since the multicondition data is already a corrupted version of the training data.

The autoencoder is implemented using Theano [17] and features 3 hidden layers with 2048 sigmoid units. The input spans over 7 consecutive MFCC frames with 13 components. The output also consists of 7 frames during training. For the decode, we average over all seven appearances for every single frame as proposed in [18].

2.3. Adapting the DA with BFE

As mentioned in the introduction and as the results will show, the performance of the DA (and DNN alone) drops significantly when there is a larger mismatch between the training data and the data to be decoded. Therefore we adapt the DA to the new condition: We first enhance the data to be decoded with BFE, resulting in a cleaned-up version of the MFCC features. These are then used as a target to retrain the DA with an initial learning rate of 0.01. After this process, the DA can be used to enhance the data for the new condition.

Note that this approach is not limited to BFE but can be used with any parametric feature enhancement method, making it possible to adapt a DA to new environmental conditions.

3. BACKEND

For the backend we use the freely available RASR toolkit [19, 20]. We chose a hybrid approach for the recognition, where a DNN calculates the posteriors for each generalized triphone state. The number of decision tree based generalized triphone states is chosen to be 3000 for every setup. The network itself consists of 6 hidden layers with 2048 sigmoid units each. It has a factorized linear bottleneck structure where each weight matrix is factorized into two matrices of dimensions 2048×256 and 256×2048. This corresponds to adding an additional linear hidden layer with 256 units between each pair of hidden layers. Such a structure reduces the number of free parameters, accelerates the decoding and training process and was found to be effective against overfitting [21, 22].

To train the network, we first initialize the weights with a greedy layer-wise pre-training. Each layer is pre-trained for 2 epochs with the learning rate set to 0.016. Afterwards, the network is fine-tuned with an initial learning rate of 0.016. The *Newbob* strategy is used to control the learning rate for the following epochs. The batch size is set to 512. Instead of stochastic gradient descent, we use an optimization algorithm called Mean-Normalized Stochstic Gradient Descent (MN-SGD). This shows faster convergence properties and enables the training of a factorized network without further decomposition methods [22]. Finally, we decode a held-out set with the weights gained from each iteration. The weights with the best WER on the held-out set are then used for the experiments. For decoding, we use the bigram language model supplied with the database (see Sec. 4).

3.1. Combining BFE and the DNN backend

We also experimented with a combination of BFE with a DNN backend, without a DA. The BFE enhanced features are used as the input for the DNN trained on the clean training data. Though we tried different setups (DNN trained on multicondition, DNN retrained on

BFE features), the described method delivered the best results. Since the BFE is a parametric model, we also have to determine its parameters. As mentioned before, the crucial parameter is the reverberation time (T_{60}). For the SimData set, the parameter is known (see Sec. 4). For the RealData set we perform a grid search to find the best parameter. We also confirmed the parameters for the SimData using this technique. But again, this parameter could also be determined in an unsupervised fashion from the single-channel speech input. The SLDM necessary for the BFE is trained on the clean utterances of the training set.

3.2. Combining the DA and DNN backend

For the combination of the DA with the DNN backend, we take the DNN trained on the clean data and retrain it with the DA output of its training data and an initial learning rate of 0.01. In contrast to [3], this method delivered the best results for us. But the results without a retraining are only slightly worse. This indicates that the DA outputs features similar to clean ones, but apparently with small differences. The retraining allows the DNN to adapt to these.

For the adapted DAs we use the DNN trained on clean data.

4. DATASET

4.1. Evaluation set

Experiments are carried out on the datasets SimData and RealData from the Reverb Challenge [1]. The vocabulary size is 5 k.

For the SimData set utterances by 28 different speakers are taken from the WSJCAM0 corpus [23] and convolved with three different room impulse responses (RIRs). Noise is added at a signal-to-noise ratio of 20 db. The rooms are named Room1, Room2 and Room3. Additionally, there are two distances between the microphone and the speaker: 50 cm for the *near* condition and 200 cm for the *far* condition. For each condition there are 363 utterance and 6 k words.

Table 1 shows two important acoustic parameters for the dataset. The first one is the C_{50} parameter (**C**larity index). It describes the ratio between the early signal energy (< 50 ms) and the rest and is thus related to the distance between the speaker and the microphone. The second parameter is the reverberation time T_{60} (compare Sec. 2.1).

The RealData set consists of 372 utterances with 6.1 k words in total. The utterances are from the MC-WSJ-AV corpus [24] and spoken by 10 different speakers. These are a set of WSJCAM0 utterances rerecorded with real speakers in a noisy and reverberant room. The set is also divided into a far and a near set but the distances are ~ 100 cm respectively ~ 250 cm this time.

4.2. Training set

The training set of the WSJCAM0 corpus is used. In the case of *clean* training, the 7861 utterances by 92 speakers are left untouched. For the *multicondition* training the utterances are convolved with RIRs from up to three different room sizes (small, medium and large) and background noise with a SNR of 20 dB is added. These RIRs are different from the ones used to generate SimData, but have comparable room reverberation times. Note that the RIRs change from one utterance to the other in the same order, so the size of the training set remains the same for all experiments and there is an equal number of utterances with different RIRs. The same acoustic parameters as for the evaluation set are shown in Table 1.

		Distance	C_{50}	T_{60}
Training	Room1	near	29 - 31 dB	250 ms
		far	21 - 22 dB	
	Room2	near	14 - 17 dB	500 ms
		far	6 - 7 dB	
	Room3	near	14 - 16 dB	700 ms
		far	6 - 7 dB	
Evaluation	Small	near	25 - 30 dB	~ 250 ms
		far	19 - 21 dB	
	Medium	near	14 - 18 dB	~ 500 ms
		far	6 - 10 dB	
	Large	near	12 - 16 dB	~ 700 ms
		far	4 - 7 dB	

Table 1. Acoustic parameters for the different conditions.

5. RESULTS

Table 2 summarizes the results obtained with different training conditions on SimData and RealData. The respective training condition is denoted as a subscript of the model. DA+DNN$_{\text{LargeFar}}$ means for example that only the RIR of the large room with a distant speaker is used[1]. If no distance is denoted, both, near and far, have been included. Further, MC (multicondition) includes all conditions and Clean are the clean training utterances.

5.1. Baseline

We start by discussing the baseline results for each model in the first five rows of Table 2. The combination of the DA and the DNN backend outperforms the other setups in every single condition when trained with full multicondition data (DA+DNN$_{\text{MC}}$). It also slightly improves the results when trained on clean data only (DA+DNN$_{\text{Clean}}$) compared to DNN$_{\text{Clean}}$. Additionally, the severe performance degradation caused by a channel mismatch becomes obvious. The word error rate (WER) doubles for more reverberant conditions when the model is trained with clean data only. In this case, BFE significantly improves the performance, achieving results competitive to DNN$_{\text{MC}}$ for *far* conditions and even outperforming it for the RealData set. Only for *near* conditions it breaks down for the reason described in sec. 2.1.

5.2. DA and DNN with channel mismatch

Next, we look at the results for a DNN and a DA+DNN for different channel mismatch conditions. This is the second block of Table 2 If the models have only seen reverberant speech ($\cdot_{\text{LargeFar}}$) the WER increases significantly for conditions with less reverberation. If they have only seen slightly reverberated speech (such as $\cdot_{\text{Small}}$), the performance decreases for highly reverberated speech. But reverberation is not the only important mismatch. Performance degradation due to a mismatch in the C_{50} parameter is also noticeable. On the other hand, the results show that there is some tolerance for the case where the model saw reverberated speech but with a different reverberation time ($\cdot_{\text{Medium}}$ vs. $\cdot_{\text{Large}}$). Additionally the DA is again able to increase the performance in all but one case (DA+DNN$_{\text{Small}}$), mostly by a significant margin. It makes it possible to use only the data from the large room for training and achieve better results than the model

[1]Although not listed in the results, we also considered MediumFar, MediumNear, SmallFar and SmallNear, but no new insights can be gained from these setups.

	Model	SimData								RealData		
		Room 1		Room 2		Room 3		Avg.		near	far	Avg.
		near	far	near	far	near	far	near	far			
Baseline	$\text{DNN}_{\text{Clean}}$	15.6	22.5	48.6	80.2	57.7	86.4	40.6	63.0	87.1	83.0	85.0
	DNN_{MC}	18.2	18.4	21.5	32.3	25.7	37.0	21.8	29.2	52.9	49.7	51.3
	$\text{BFE+DNN}_{\text{Clean}}$	19.8	17.4	30.0	29.0	42.2	43.8	30.7	30.0	48.2	47.2	47.7
	$\text{DA+DNN}_{\text{Clean}}$	15.3	22.4	46.9	79.1	54.3	85.7	38.8	62.4	83.4	80.6	82.0
	$\text{DA+DNN}_{\text{MC}}$	14.7	15.6	16.6	26.4	20.8	31.3	**17.4**	24.4	44.3	45.2	44.8
Mismatch	$\text{DNN}_{\text{Small}}$	16.2	18.0	24.8	56.3	31.3	63.9	24.0	46.1	71.4	69.0	70.2
	$\text{DA+DNN}_{\text{Small}}$	13.3	14.4	24.3	63.9	31.7	70.2	23.1	49.5	72.4	72.8	72.6
	$\text{DNN}_{\text{Medium}}$	17.8	18.2	25.2	37.2	27.3	38.4	23.4	31.2	53.4	50.6	52.0
	$\text{DNN+DA}_{\text{Medium}}$	14.6	16.2	22.8	33.8	23.7	32.8	24.5	29.9	52.3	50.0	46.5
	$\text{DNN}_{\text{Large}}$	20.5	22.2	23.7	30.9	29.3	36.6	24.5	29.9	52.3	50.0	51.1
	$\text{DA+DNN}_{\text{Large}}$	15.6	17.5	16.8	24.4	21.1	29.6	17.9	**23.8**	43.4	43.2	43.3
	$\text{DNN}_{\text{LargeFar}}$	40.0	34.3	39.5	38.0	40.9	39.9	40.1	37.4	55.0	52.2	53.6
	$\text{DA+DNN}_{\text{LargeFar}}$	25.7	23.4	30.3	31.6	31.7	32.0	29.2	29.0	44.3	44.4	44.4
	$\text{DNN}_{\text{LargeNear}}$	16.9	18.2	25.2	47.2	29.0	50.7	23.7	38.7	63.3	57.7	60.5
	$\text{DA+DNN}_{\text{LargeNear}}$	13.3	16.0	21.8	43.7	23.2	44.8	19.4	34.8	55.9	53.4	54.7
Adapt	$\text{DA+DNN}^{\text{Sim}}_{\text{Small}}$	18.6	18.7	22.4	29.4	28.4	41.9	23.1	30.0	48.2	47.8	48.0
	$\text{DA+DNN}^{\text{Sim}}_{\text{LargeFar}}$	20.8	20.7	24.4	29.5	29.6	41.2	24.9	30.3	48.0	48.2	48.1
	$\text{DA+DNN}^{\text{Sim}}_{\text{Clean}}$	18.4	18.6	24.1	32.1	29.3	43.7	24.0	31.5	51.9	50.7	51.3
	$\text{DA+DNN}^{\text{Real}}_{\text{Clean}}$	22.7	24.2	32.3	46.4	36.8	54.4	30.6	41.7	45.7	46.1	45.9
	$\text{DA+DNN}^{\text{Real}}_{\text{MC}}$	23.6	23.7	26.3	34.4	30.7	43.4	26.9	33.8	42.0	41.0	**41.5**

Table 2. Word error rates for SimData and RealData for different system combinations and training data. The training condition is denoted as a subscript. In case of adaptation, the data used for adaptation is denoted as a superscript.

trained with all conditions ($\text{DA+DNN}_{\text{Large}}$ vs. $\text{DA+DNN}_{\text{MC}}$). Note that there is no mismatch for BFE since it is a parametric model and can be adjusted to new conditions. All results must be compared to the baseline ($\text{BFE+DNN}_{\text{Clean}}$).

5.3. Adaption with BFE

Finally, we show how a parametric method can be used to adapt the DA to unseen conditions, leading to a significant WER reduction. We conduct two different adaptations. One is an adaptation to the SimData (superscript $^{\text{Sim}}$), the other is an adaptation to the RealData (superscript $^{\text{Real}}$). The first one is carried out using all sets of the SimData, while the later one uses all sets of the RealData.

The last block of Table 2 shows the results for different adapted models. The first three rows are related to SimData where one model has only seen slightly reverberated speech, one only highly reverberated far speech and one no reverberated speech at all. Retraining these models brings a significant gain compared to the results with the unadapted models. The biggest improvement can be seen for $\text{DA+DNN}_{\text{Clean}}$ (vs. $\text{DA+DNN}^{\text{Sim}}_{\text{Clean}}$), with a relative WER reducion of nearly 50%. Importantly, all adapted models deliver better results than the BFE baseline. Especially the ability to improve features for the *near* condition remains untouched. This means that the DA does not just learn the same mapping the BFE performs but rather keeps a part of its original mapping while adjusting it to the new condition. Also, even though adapted with SimData only, the gain is also visible for the RealData set. This indicates that the DA still generalizes instead of only working on the adapted condition.

The last two rows show an adaptation with RealData. Again, the performance increases significantly. Even more interesting, the adapted model $\text{DA+DNN}^{\text{Real}}_{\text{MC}}$ outperforms its unadapted equivalent $\text{DA+DNN}_{\text{MC}}$. This shows, that the proposed adaptation is able to reduce a mismatch between simulated and real reverberant data.

6. CONCLUSIONS

The combination of a DA and a DNN delivers good performance when the conditions at decode time have been included in the training data. It can even compensate for a mismatch in reverberation time to a certain extend. But when the mismatch becomes too large, the WER grows quickly. BFE on the other hand can be adjusted to unseen conditions by parameter selection, though the performance is behind the one a DA can achieve in a matched condition. But while it is possible to avoid a mismatch caused by the acoustic properties of the room and the speaker position by just generating enough artificially reverberated training data using different RIRs, there is still a gap between real recordings and simulated data. The results show that BFE as a parametric method can produce cleaned-up feature vectors which serve as targets for the DA for an unsupervised adaptation to the decoding conditions, bridging that gap.

7. RELATION TO PRIOR WORK AND OUTLOOK

This work builds on the DA [3, 16, 18, 25] and BFE [8, 9] as feature enhancement techniques. It also uses recent advancements in the acoustic modeling for the backend [19, 22]. We propose to make use of the advantages of both feature enhancement techniques: The possibility to be adjusted to new conditions of a parametric model (BFE) and the modelling power of a neural network (AE) by using the BFE to adapt the DA. In future research, we will try other parametric methods as we suspect that this is not limited to BFE. Further, a combination of different parametric methods for the adaptation might bring an additional performance gain.

8. REFERENCES

[1] Keisuke Kinoshita, Marc Delcroix, Takuya Yoshioka, Tomohiro Nakatani, Emanuel Habets, Reinhold Haeb-Umbach, Volker Leutnant, Armin Sehr, Walter Kellermann, Roland Maas, Sharon Gannot, and Bhiksha Raj, "The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2013.

[2] Heikki Kallasjoki, Jort F Gemmeke, Kalle J Palom, Amy V Beeston, and Guy J Brown, "Recognition of Reverberant Speech by Missing Data Imputation and NMF," in *REVERB Challange*, 2014.

[3] Masato Mimura, Shinsuke Sakai, Tatsuya Kawahara, and Media Studies, "REVERBERANT SPEECH RECOGNITION COMBINING DEEP NEURAL NETWORKS AND DEEP AUTOENCODERS," 2014.

[4] Felix Weninger, Shinji Watanabe, Jonathan Le Roux, John R Hershey, Yuuki Tachioka, Gerhard Rigoll, and Mitsubishi Electric, "Deep Recurrent Neural Network Feature Enhancement," in *REVERB Challange*, 2014.

[5] Xiong Xiao, Shengkui Zhao, Duc Hoang, Ha Nguyen, Xionghu Zhong, Douglas L Jones, Eng Siong Chng, and Haizhou Li, "The NTU-ADSC Systems for Reverberation Challenge 2014," in *REVERB Challange*, 2014.

[6] Marc Delcroix, Takuya Yoshioka, Atsunori Ogawa, Yotaro Kubo, Masakiyo Fujimoto, and Nobutaka Ito, "Linear Prediction-based Dereverberation with Advanced Speech Enhancement and Recognition Technologies for the REVERB Challenge," in *REVERB Challange*, 2014.

[7] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol, "Extracting and composing robust features with denoising autoencoders," 2008, pp. 1096–1103.

[8] Volker Leutnant, Alexander Krueger, and Reinhold Haeb-umbach, "Bayesian Feature Enhancement for Reverberation and Noise Robust Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 8, pp. 1640–1652, 2013.

[9] Volker Leutnant, Alexander Krueger, and Reinhold Haeb-Umbach, "A New Observation Model in the Logarithmic Mel Power Spectral Domain for the Automatic Recognition of Noisy Reverberant Speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 1, pp. 95–109, 2014.

[10] Alexander Krueger and Reinhold Haeb-Umbach, "Model-based feature enhancement for reverberant speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1692–1707, 2010.

[11] Jean-Dominique Polack, "La transmission de l'energie sonore dans les salles," 1988.

[12] Nikolay D Gaubitch, Heinrich W Loellmann, Marco Jeub, Tiago H Falk, Patrick A Naylor, Peter Vary, and Mike Brookes, "Performance comparison of algorithms for blind reverberation time estimation from speech," in *Acoustic Signal Enhancement; Proceedings of IWAENC 2012; International Workshop on*. VDE, 2012, pp. 1–4.

[13] Heiner Löllmann, Emre Yilmaz, Marco Jeub, and Peter Vary, "An improved algorithm for blind reverberation time estimation," in *Proceedings of International Workshop on Acoustic Echo and Noise Control (IWAENC)*, 2010, pp. 1–4.

[14] Geoffrey E Hinton and Ruslan R Salakhutdinov, "Reducing the dimensionality of data with neuronal networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[15] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems 19 (NIPS'06)*, 2007, pp. 153–160.

[16] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[17] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010, Oral Presentation.

[18] Takaaki Ishii, Hiroki Komiyama, Takahiro Shinozaki, Yasuo Horiuchi, and Shingo Kuroiwa, "Reverberant speech recognition based on denoising autoencoder.," in *INTERSPEECH*, 2013, pp. 3512–3516.

[19] David Rybach, Stefan Hahn, Patrick Lehnen, David Nolden, Martin Sundermeyer, Zoltan Tüske, Siemon Wiesler, Ralf Schlüter, and Hermann Ney, "Rasr-the rwth aachen university open source speech recognition toolkit," in *ASRU*, 2011.

[20] Jonas Lööf, Christian Gollan, Stefan Hahn, Georg Heigold, Björn Hoffmeister, Christian Plahl, David Rybach, Ralf Schlüter, and Hermann Ney, "The rwth 2007 tc-star evaluation system for european english and spanish.," in *INTERSPEECH*, 2007, pp. 2145–2148.

[21] 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 *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2013, pp. 6655–6659.

[22] Simon Wiesler, Alexander Richard, Ralf Schluter, and Hermann Ney, "Mean-normalized Stochastic Gradient for Large-scale Deep Learning," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2014, pp. 180–184.

[23] Tony Robinson, Jeroen Fransen, David Pye, Jonathan Foote, and Steve Renals, "Wsjcam0: A british english speech corpus for large vocabulary continuous speech recognition," in *In Proc. ICASSP 95*. 1995, pp. 81–84, IEEE.

[24] 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 *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, Nov 2005, pp. 357–362.

[25] James Glass Xue Feng, Yaodong Zhang, "Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2014, pp. 1778–1782.