SPEECH FEATURE DENOISING AND DEREVERBERATION VIA DEEP AUTOENCODERS FOR NOISY REVERBERANT SPEECH RECOGNITION

Xue Feng, Yaodong Zhang, James Glass

MIT Computer Science and Artificial Intelligence Laboratory
Cambridge, MA, USA, 02139
{xfeng, ydzhang, jrg}@csail.mit.edu

ABSTRACT

Denoising autoencoders (DAs) have shown success in generating robust features for images, but there has been limited work in applying DAs for speech. In this paper we present a deep denoising autoencoder (DDA) framework that can produce robust speech features for noisy reverberant speech recognition. The DDA is first pre-trained as restricted Boltzmann machines (RBMs) in an unsupervised fashion. Then it is unrolled to autoencoders, and fine-tuned by corresponding clean speech features to learn a nonlinear mapping from noisy to clean features. Acoustic models are re-trained using the reconstructed features from the DDA, and speech recognition is performed. The proposed approach is evaluated on the CHiME-WSJ0 corpus, and shows a 16-25% absolute improvement on the recognition accuracy under various SNRs.

Index Terms— robust speech recognition, feature denoising, denoising autoencoder, deep neural network

1. INTRODUCTION

There is a continuously growing demand for hands-free speech input for various applications [1, 2]. One driving force behind this development is the rapidly increasing use of portable devices such as hands-free mobile telephones, tablets and voice-controlled systems. Another important application where distant speech is of interest is that of hearing aids.

In the above distant-talking speech communication systems, the presence of environmental noise, and/or reverberation, often causes a dramatic performance drop on automatic speech recognition (ASR) systems. To improve the robustness of ASR, different approaches have been investigated: Front-end methods include speech signal pre-processing [3, 4, 5], robust acoustic features [6, 7]; back-end methods include model compensation or adaptation [8], and uncertainty decoding [9] etc. Traditional speech signal front-end techniques focus on spatial speech processing and separation techniques such as non-negative matrix factorization (NMF) [4, 5]. Recently, along with the growing popularity of using deep neural network (DNN)-HMMs for ASR, many researchers have also reported different ways of using DNNs to generate robust speech features. For example, Sainath *et al.* explored using a deep bottleneck autoencoder to produce features for GMM-HMM based ASR and obtained good recognition results [10]. Vinyals *et al.* investigated the effectiveness of DNNs for detecting articulatory features, which combined with MFCC features were used for robust ASR tasks [7].

In this paper we investigate an alternative front-end method to obtain robust features via deep denoising autoencoders (DDAs). The proposed DDA framework involves layers of affine+sigmoid encoding followed by affine decoding to recover speech features from noisy reverberant speech features. The DDA is fine-tuned by clean features, which results in learning a stochastic mapping from noisy to clean. Denoising autoencoders (DAs) have been visited by Vincent *et al.* in [11] and Bengio in [12], and stacked to form stacked denoising autoencoder (SDA) in [13] to generate robust features for images. Moreover, in [14], DAs are applied to reconstruct clean speech spectrum from reverberant speech. The key differences between our proposed framework with this prior work are a) denoising is performed on the feature level, b) multiple frames of noisy features are mapped to single frame output features directly c) the training procedure is differenct since the decoder layers are no longer symmetric with the encoder layers.

We validate the effectiveness of our proposed DDA front-end denoising approach on track 2 of the second CHiME challenge [15]. Track 2 is a 5k medium-vocabulary speech recognition task in reverberant and noisy environment, whose utterances are taken from the Wall Street Journal database (WSJ0).

The rest of the paper is organized as follows. Section 2 presents the proposed DDA feature denoising architecture. Section 3 describes the training procedure. Then we evaluate the performance in Section 4, before concluding.

2. DEEP DENOISING AUTOENCODER MODEL

2.1. Traditional Autoencoder

Autoencoders consist of the encoder and the decoder. The encoder is the deterministic mapping f_θ that transforms a n-

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

dimensional input vector $\mathbf{x}$ into a hidden representation $\mathbf{y}$. The typical form is an affine mapping followed by a nonlinearity:

$$f_\theta(\mathbf{x}) = s(\mathbf{W}\mathbf{x} + \mathbf{b}),$$

with parameter set $\theta = \{\mathbf{W}, \mathbf{b}\}$, where $\mathbf{W}$ is a $d \times d$ weight matrix and b is an offset vector of dimensionality d. The resulting hidden representation $\mathbf{y}$ is then mapped back to a reconstructed d-dimensional vector $\mathbf{z}$ in input space, with $\mathbf{z} = g'_\theta(\mathbf{y})$.This mapping is called the decoder. Its typical form is an affine mapping optionally followed by a squashing nonlinearity, that is, either

$$g_{\theta'}(\mathbf{y}) = \mathbf{W}'\mathbf{y} + \mathbf{b}',$$

or

$$g_{\theta'}(\mathbf{y}) = s(\mathbf{W}'\mathbf{y} + \mathbf{b}').$$

with appropriately sized parameters $\theta' = \{\mathbf{W}', \mathbf{b}'\}$. In general, $\mathbf{z}$ is not to be interpreted as an exact reconstruction of $\mathbf{x}$, but rather in probabilistic terms as the parameters (typically the mean) of a distribution $p(X|Z = \mathbf{z})$ that may generate $\mathbf{x}$ with high probability. This yields an associated reconstruction error to be optimized with respect to loss $L(\mathbf{x}, \mathbf{z}) = -\log p(\mathbf{x}|\mathbf{z})$. For real-valued $\mathbf{x}$, this requires $X|\mathbf{z} \sim \mathcal{N}(\mathbf{z}, \sigma^2\mathbf{I})$, which yields $L(\mathbf{x}, \mathbf{z}) = C(\sigma^2)||\mathbf{x} - \mathbf{z}||^2$, where $C(\sigma^2)$ denotes a constant that depends only on σ^2 and thus can be ignored for the optimization. This is the squared error objective found in most traditional autoencoders. In this setting, due to the Gaussian interpretation, it is more natural not to use a squashing nonlinearity in the decoder. For the rest of this paper, we use **affine+sigmoid encoder** and **affine decoder** with **squared error loss**.

2.2. Denoising Autoencoder

The denoising autoencoder (DA) is a straightforward variant of the basic autoencoder. A DA is trained to reconstruct a clean input $\mathbf{x}$ from a corrupted version of it. The corrupted input $\tilde{\mathbf{x}}$ is mapped, as with the basic autoencoder, to a hidden representation $f_\theta(\tilde{\mathbf{x}}) = \text{sigmoid}(\mathbf{W}\tilde{\mathbf{x}} + \mathbf{b})$ from which we reconstruct a $\mathbf{z} = g_{\theta'}(\mathbf{y}) = (\mathbf{W}'y + \mathbf{b}')$. Instead of minimizing the loss function $L(\tilde{\mathbf{x}}, \mathbf{z})$ between the input and the output, parameters θ and θ' are trained to minimize the average reconstruction error over a clean training set, that is, to have $\mathbf{z}$ as close as possible to the uncorrupted input $\mathbf{x}$, with $L(\mathbf{x}, \mathbf{z}) \propto ||\mathbf{x} - \mathbf{z}||^2$. Note that for our speech denoising and dereverberation task, the corrupted feature $\tilde{\mathbf{x}}$ is not of the same dimension as $\mathbf{x}$. Due to the reverberations, information from previous frames is leaked to the current frame, and noises also makes adjacent frame features less independent. We use concatenated MFCCs from fifteen contiguous frames as $\tilde{\mathbf{x}}$ to encode, and use only the corresponding middle frame clean MFCCs as $\mathbf{x}$ to fine-tune.

2.3. Deep Denoising Autoencoder

By using multiple layers of encoder and decoder, the DA can form a deep architecture and become a Deep Denoising Autoencoder (DDA). Note that since we use an affine decoder without nonlinearity, one can easily join the layers of decoders to form one single decoder layer. The system work flow is demonstrated in Figure 1. Specifically, with parallel clean and noisy speech data available, a DDA can be pre-trained on noisy reverberant speech features and fine-tuned by clean speech features. The rich nonlinear structure in the DDA can be used to learn an efficient transfer function which removes noise in speech while keeping enough phonetically discriminative information to generate good reconstructed features.

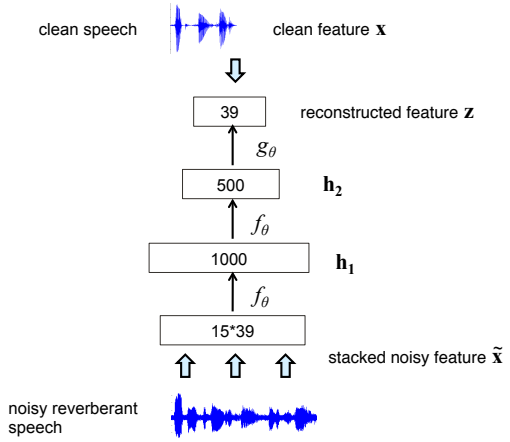


Fig. 1. Deep Denoising Autoencoder Architecture. This figure gives an example of a DDA containing two encoder layers, with 1000 nodes and 500 nodes respectively.

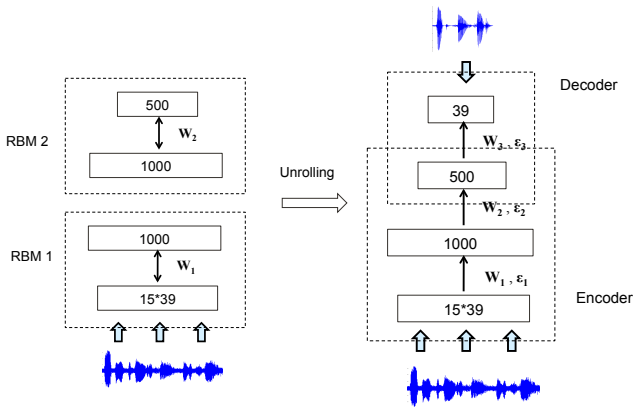
3. TRAINING A DEEP DENOISING AUTOENCODER

3.1. Pre-training

Instead of initializing hidden weights with little guidance, we perform pre-training by adopting an efficient approximation learning algorithm proposed by Hinton *et al.* called one-step contrastive divergence (CD-1) [16]. The generative pre-training not only requires no supervised information, but can also put all hidden weights into a proper range which can be used to avoid local optima in the supervised back-propagation based fine-tuning.

Figure 2(a) illustrates the pre-training of our DDA. Pre-training consists of learning a stack of restricted Boltzmann machines (RBMs), each having only one layer of feature detectors. After learning one RBM, the status of the learned hidden units given the training data can be used as feature vectors for the second RBM layer. The CD-1 method can be used to learn the second RBM in the same fashion. Then, the

status of the hidden units of the second RBM can be used as the feature vectors for the third RBM, etc. This layer-by-layer learning can be repeated for many times. After the pre-training, the RBMs are unrolled to create a deep autoencoder, which is then fine-tuned using back-propagation of error derivatives [17].



(a) Pre-training (b) Fine-Tuning

Fig. 2. Pre-training consists of learning a stack of restricted Boltzmann machines (RBMs). After pre-training, the RBMs are unrolled to create a deep autoencoder, which is then fine-tuned using back-propagation of error derivatives.

3.2. Fine-tuning

Figure 2(b) demonstrates the fine-tuning state of a DDA. The goal of back-propagation fine-tuning is to minimize the squared error loss on the entire dataset between the reconstructed and clean vectors as follows:

$$F = \sum_{i=1}^{U} ||\mathbf{x}^i - \mathbf{z}^i||^2 \tag{1}$$

where U is the total number of training cases, $\mathbf{z}^i$ is the i-th reconstructed feature vector, and $\mathbf{x}^i$ is the corresponding clean feature vector. Suppose we have a DDA with M hidden layers and a decoder layer with N outputs (e.g., 39). By taking partial derivatives, the gradient of weights for the decode layer are

$$\frac{\partial F}{\partial \mathbf{W}_l} = \sum_{i=1}^{U} [Z_l(i)E_l(i)]^T \mathbf{v}_l^i \tag{2}$$

where each $\mathbf{v}_l^i$ represents the output of the l-th hidden layer for the i-th input. and the gradients for the bias are

$$\frac{\partial F}{\partial \epsilon_l} = \sum_{i=1}^{U} [Z_l(i)E_l(i)]^T \tag{3}$$

where Z is the transfer function and E is the error function. For the decoder layer

$$Z_{M+1}(i) = \mathbf{1} \tag{4}$$

$$E_{M+1}(i) = \mathbf{x}^i - \mathbf{z}^i. \tag{5}$$

For the l-th hidden layer ($l \in [1..M]$),

$$Z_l(i) = (\mathbf{W}_l \mathbf{v}_l^i + \epsilon_l) \cdot (1 - \mathbf{W}_l \mathbf{v}_l^i - \epsilon_l) \tag{6}$$

$$E_l(i) = \mathbf{W}_l Z_l(i) E_{l+1}(i) \tag{7}$$

After calculating these gradients, stochastic gradient descent (SGD) is used to update the parameters [12].

4. EVALUATION

The effectiveness of the proposed framework is evaluated on ChiME-WSJ0 corpus [15] using MFCC features. A DDA is first trained on the training set. Next, raw test speech features are processed by the trained DDA. Acoustic model is retrained using the processed features, and the retrained model is utilized for speech recognition.

4.1. Dataset

Track 2 data from second CHiME challenge is a 5k-vocabulary task in reverberant and noisy environment, whose utterances are taken from the Wall Street Journal database (WSJ0). The training data set (si_tr_s) contains 7,138 utterances from 83 speakers, the evaluation data set (si_et_05) contains 330 utterances from 12 speakers (Nov92), and the development set (si_dt_05) contains 409 utterances from 10 speakers. Acoustic models were trained using si_tr_s and some of the parameters (e.g., language model weights) were tuned based on the WERs of si_dt_05. This database simulates a realistic environment. We use the type of data called *Isolated*, which is created as follows: First, clean speech is convolved with binaural room impulse responses corresponding to a frontal position at a distance of 2 m from the microphones in a family living room; Second, real-world noises recorded in the same room are added, with the noise excerpts selected to obtain signal-to-noise ratios (SNRs) of -6, 3, 0, 3, 6, and 9 dB without rescaling. Noises are non-stationary such as other speakers utterances, home noises, or background music.

4.2. Acoustic features

Both the clean and noisy reverberant speech waveforms are parameterized into a sequence of standard 39-dimensional Mel-frequency cepstral coefficient (MFCC) vectors: 12 Mel-cepstral coefficients processed by cepstral mean normalization (CMN), plus logarithmic frame energy and delta and acceleration coefficients. The MFCCs are extracted from 25ms time frames with a step size of 10ms. Prior to feature extraction, the input binaural signals are down-mixed to mono by averaging the two channels together. Although this down-mixing operation leads to a small degradation of WER, we decided to use it in order to focus on the evaluation of the front-end processing technique.

4.3. Experimental Setup

Table 1 presents the results for different DDA configurations and their resulting average WER over 6 SNR scenarios. In the first column, 500 indicates a DDA that has one encoder layer with 500 hidden units, while 500x500 denotes a DDA with two hidden layers each of which has 500 hidden units. All DDA configurations have an input layer with 15*39 units and an affine encoder output layer with 39 units. The pre-training in each configuration was set to stop at the 25th iteration with a learning rate of 0.004 and a batch size of 256. The fine-tuning using back-propagation was set to stop at the 50th iteration using the line search stochastic gradient decent method with a batch size of 256. As we can see from Table 1, the average WER is not very sensitive to the DDA configurations. For the following experiments, the 500x500 configuration is used.

DDA	Average WER
500	35.69%
500x500	34.04%
1000x1000	34.51%
500x500x500	34.22%

Table 1. *WER vs. different DDA configurations.*

4.4. Speech Recognition

The HMM/GMM training follows the recipe in [18]. The number of phonemes is 41: 39 phones plus 1 silence (sil) and 1 short pause (sp) model. The output distributions of sp and sil have their parameters tied. The number of clustered triphone HMM states is 1860 and is relatively smaller than the conventional setup (more than 2000 states). Each HMM has three output states with a left-to-right topology with self-loops and no skip. Each HMM state is represented by a GMM with 8 components for phoneme-based HMMs and 16 for silence-based HMMs. The standard WSJ 5K non-verbalized closed bigram language model is considered. We only re-estimate the HMM/GMM parameters from a clean speech acoustic model, and do not change the model topology for simplicity. Decoding is performed using HVite [19] with a pruning threshold.

4.5. Results

Table 2 reports the performance of the system trained on *Isolated* (noisy and reverberated) dataset as a function of the SNR. We compare the WER with and without the proposed front-end processing. We can see that over six SNR scenarios, the proposed method improved the recognition accuracy by 16.68%, 19.88%, 25.05%, 22.87%, 21.8%, and 20.51% respectively. The baseline system is trained on matching noisy reverberant data with the exact same setting. This improvement shows the impact of front-end denoising and dereverberation. The improvement is more dramatic in the 0 dB and 3 dB cases. This should be affected by the fact that we did not train our DDA for various SNR degrees. Thus the feature mapping tends to either overfit or underfit for high or low SNR cases. In comparing our results against those obtained by the actual participants of the CHiME Challenge [20], ours are among the top two. Note that the CHiME challenge participants employed strategies at the spatial signal, feature and model levels [21] while we only utilize a front-end feature denoising technique. If we were to combine our proposed method with the spatial information and back-end techniques, the results would have been better.

SNR	WER	
	Baseline using MFCC features	Using proposed DDA reconstructed feature
-6dB	70.43%	53.75%
-3dB	63.09%	44.21%
0dB	58.42%	33.37%
3dB	51.06%	28.19%
6dB	45.32%	23.52%
9dB	41.73%	21.22%

Table 2. *WER under different SNRs.*

5. CONCLUSION AND FUTURE WORK

In this paper we presented a front-end speech feature denoising and dereverberation method based on deep denoising autoencoders. The proposed framework is unsupervised and learns a stochastic mapping from the corrupted features to the clean ones. Speech recognition experiments on the 5k noisy reverberant CHiME-WSJ0 corpus showed a 16 to 25% absolute improvement compared to the provided baseline. Our results are also among the top two for task 2 in the second CHiME Challenge, without using any backend technique. In the future, we plan to train a noise adaptive DDA for feature denoising by feeding estimated SNR as a model parameter. Also, since this feature denoising method does not preclude the use of many other front-end or back end methods, we would like to combine this approach together with array speech processing and back-end model adaptation. Experiments on larger vocabulary tasks, and with languages other than English will also be performed, since this DDA based speech feature denoising framework is language independent.

6. ACKNOWLEDGEMENTS

Thank you to Ekapol Chuangsuwanich for useful discussions. Thank you to Shinji Watanabe for providing the data and baseline scripts.

7. REFERENCES

[1] C. Nikias and J. Mendel, "Signal processing with higher-order spectra," *Signal Processing Magazine*, vol. 10, no. 3, pp. 10–37, 1993.

[2] R. A. Wiggins, "Minimum entropy deconvolution," *Geoexploration*, vol. 16, no. 1, pp. 21–35, 1978.

[3] Z. Koldovskỳ, J. Málek, J. Nouza, and M. Balık, "Chime data separation based on target signal cancellation and noise masking," *Proc. CHiME*, 2011.

[4] K. Wilson, B. Raj, P. Smaragdis, and A. Divakaran, "Speech denoising using nonnegative matrix factorization with priors," in *Proc. ICASSP*, 2008.

[5] F. Weninger, M. Wollmer, J. Geiger, B. Schuller, J. Gemmeke, A. Hurmalainen, T. Virtanen, and G. Rigoll, "Non-negative matrix factorization for highly noise-robust asr: To enhance or to recognize?," in *Proc. ICASSP*, 2012.

[6] H. K. Maganti and M. Matassoni, "An auditory based modulation spectral feature for reverberant speech recognition.," in *Proc. INTERSPEECH*, 2010.

[7] O. Vinyals and S. V. Ravuri, "Comparing multilayer perceptron to deep belief network tandem features for robust asr," in *Proc. ICASSP*, 2011.

[8] J. Du and Q. Huo, "A feature compensation approach using high-order vector taylor series approximation of an explicit distortion model for noisy speech recognition," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2285–2293, 2011.

[9] F. R. Astudillo, *Integration of short-time fourier domain speech enhancement and observation uncertainty techniques for robust automatic speech recognition*, Ph.D. thesis, Technische Universit at Berlin, 2010.

[10] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *Proc. ICASSP*, 2012.

[11] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. ICML*, 2008.

[12] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends in Machine Learning*, vol. 2, no. 1, 2009.

[13] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[14] T. Ishii, H. Komiyama, T. Shinozaki, Y. Horiuchi, and S. Kuroiwa, "Reverberant speech recognition based on denoising autoencoder," *Proc. INTERSPEECH*, 2013.

[15] E. Vincent, J. Barker, S. Watanabe, J. L. Roux, F. Nesta, and M. Matassoni, "The second chime speech separation and recognition challenge: Datasets, tasks and baselines," in *Proc. ICASSP*, 2013.

[16] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[17] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[18] K. Vertanen, "Baseline WSJ acoustic models for HTK and Sphinx: Training recipes and recognition experiments," Tech. Rep., Technical report, University of Cambridge, 2006.

[19] Steve Young *et. al.*, "The HTK book (Version 3.2)," *Cambridge Univ. Press*, 2002.

[20] "The 2nd chime speech separation and recognition challenge track 2 results," `http://spandh.dcs.shef.ac.uk/chime_challenge/track2_results.html`.

[21] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second 'chime' speech separation and recognition challenge: An overview of challenge systems and outcomes," in *Proc. ASRU*, 2013.