Adaptive Denoising Autoencoders: A Fine-tuning Scheme to Learn from Test Mixtures

Minje Kim[1] and Paris Smaragdis[2]

[1] Department of Computer Science, University of Illinois at Urbana-Champaign
minje@illinois.edu
[2] University of Illinois at Urbana-Champaign,
Adobe Research,
paris@illinois.edu

Abstract. This work aims at a test-time fine-tune scheme to further improve the performance of an already-trained Denoising AutoEncoder (DAE) in the context of semi-supervised audio source separation. Although the state-of-the-art deep learning-based DAEs show sensible denoising performance when the nature of artifacts is known in advance, the scalability of an already-trained network to an unseen signal with an unknown characteristic of deformation is not well studied. To handle this problem, we propose an adaptive fine-tuning scheme where we define a test-time target variables so that a DAE can learn from the newly available sources and the mixing environments in the test mixtures. In the proposed network topology, we stack an AutoEncoder (AE) trained from clean source spectra of interest on top of a DAE trained from a variety of available mixture spectra. Hence, the bottom DAE outputs are used as the input to the top AE, which is to check the purity of the once denoised DAE output. Then, the top AE error is used to fine-tune the bottom DAE during the test phase. Experimental results on audio source separation tasks demonstrate that the proposed fine-tuning technique can further improve the sound quality of a DAE during the test procedure.

Keywords: Deep Learning, Deep Neural Networks, Autoencoders, Speech Enhancement, Semi-supervised Separation

1 Introduction

Recent advances in the deep learning research greatly improved the single-channel audio source separation performance as well. Most of the time, the Deep Neural Networks (DNN) commonly take a set of frequency coefficients of a short time period of the mixed signal, but there are three different choices for the output. First, the network can produce Ideal Binary Masks (IBM) [11], which are binary labels that tell us whether each frequency coefficient (T-F unit in the cochleagram usually) belongs to the interesting source (e.g. speech) or not (e.g.

noise). Second, a DNN can generate all the spectra of unmixed sources simultaneously [3]. This kind of models is more difficult to learn due to the higher dimension of the output layer, but they tend to produce even reconstruction qualities for all the sources. Third, a Denoising AutoEncoder (DAE) can take a noisy spectrum, and then outputs its cleaned-up version [5] [13]. DAEs are common in deep learning as a feature learning technique, where the inputs are perturbed with some stationary noise [10][7]. In the source separation applications however, a DAE is trained with more realistic acoustic noise types.

There is no good ways for those DNNs for source separation to adapt to an unseen signal, while adapting a half-trained model during the test time has been a common idea in the source separation research in the name of *semi-supervised separation* [1]. It has a merit especially when an established separation model cannot efficiently represent a test signal with some unknown sources in it. To handle this problem, the semi-supervised separation systems build a part of the model for the desired source in advance, and then train the rest of the model from the residual of the test signal.

In this work we propose to vertically stack a pair of a DAE and an AutoEncoder (AE) as a source separation system that adapts to the unknown characteristics of the test signals. First, we train a DAE with an available set of noisy spectra and their corresponding clean spectra as the input and the output, respectively. However, we also consider the fact that the noisy spectra for training might not be diverse enough to cover all the variation of deformation that can happen in the real world, such as different types and levels of additive noise. As in the semi-supervised separation scenario, we improve this imperfectly trained DAE by fine-tuning it to minimize the test-time error we newly define. To this end, we set up another AE that is dedicated to produce a clean spectrum if its input is the clean spectra of a target source as well, which we also call a *purity checker*. It gives us a lower error if its input is clean and higher otherwise. During the separation phase, we first denoise the input using the bottom DAE. Then, we check on the purity of the DAE output by feeding it as an input to the top AE. In this way, instead of a single path feedforward for the denoising job, we measure the quality of the once denoised spectrum and backpropagate the error of the top AE to fine-tune the bottom DAE.

We show that the proposed method can sensibly improve Signal-to-Interference Ratio (SIR), while sacrificing Signal-to-Artifact Ratio (SAR) a little. Therefore, there is a point where we get better Signal-to-Distortion Ratio (SDR).

2 Related Work

This section introduces semi-supervised NMF models and two-stage approaches, which are conceptually and structurally similar to our work, respectively.

2.1 Semi-supervised Source Separation

In the semi-supervised source separation methods, Nonnegative Matrix Factorization [4], or Probabilistic Latent Component Analysis (PLCA) [6] as an audio

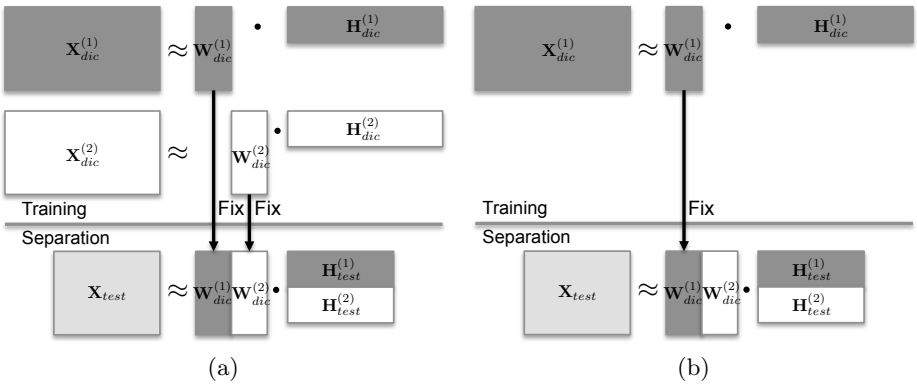


Fig. 1: (a) Supervised separation (b) Semi-supervised separation.

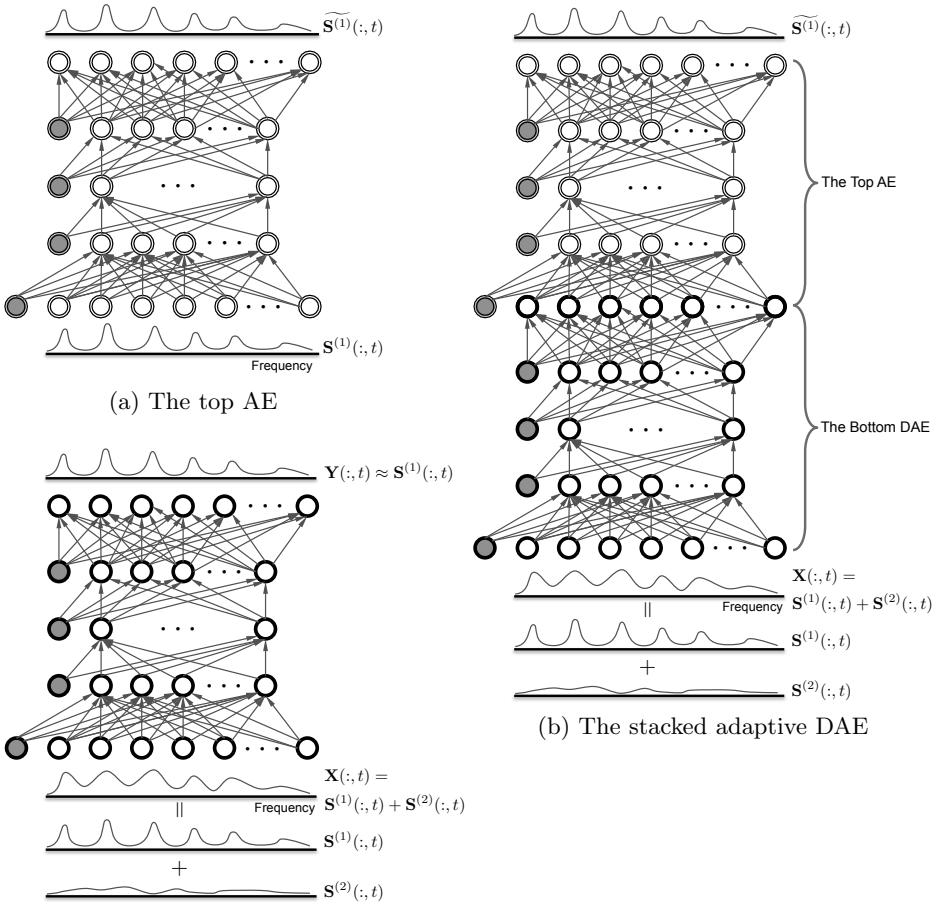
analogy of topic modeling, is a popular tool to discover latent structures of a mixed signal[3]. Fig. 1 compares the two different strategies. In (a) we assume that a set of magnitudes of Fourier spectra, e.g. $\mathbf{X}_{dic}^{(1)}$ from clean speech, is available for the NMF algorithm to train the source specific bases, or a dictionary, $\mathbf{W}_{dic}^{(1)}$ and their temporal activations, $\mathbf{H}_{dic}^{(1)}$. As a fully supervised case, we also learn the second source's dictionary $\mathbf{W}_{dic}^{(2)}$ (e.g. "babble" noise). For the separation, we fix the two dictionaries during the final NMF learning, while both sets of their activations $\mathbf{H}_{test}^{(1)}$ and $\mathbf{H}_{test}^{(2)}$ are learned to best describe the unseen test input $\mathbf{X}_{test}$. Finally, we recover the source by multiplying its corresponding dictionary matrix and activations, e.g. $\mathbf{W}_{dic}^{(1)}\mathbf{H}_{test}^{(1)}$ for the first source.

In the semi-supervised scenario on the other hand, we assume that only a part of the sources is known (the first source is known in Fig. 1 (b), while the other sources are not). However, by fixing only the first part of the dictionary matrix with the trained one $\mathbf{W}_{dic}^{(1)}$, and learning the other part $\mathbf{W}_{dic}^{(2)}$ along with the activations from the test spectra, we can still recover the test mixture in a semi-supervised fashion. This approach is particularly useful if we are not sure about the quality of the second source's training set, or if it does not exist at all.

2.2 Two-stage Approaches

In the two-stage approaches [12], an additional NMF approximation helps improve the masking-based DNN results. First, a two-stage system uses its usual DNN-based T-F masking module as a front-end to denois the noisy speech signal. In the second stage of the system, NMF is employed to further improve the estimated speech signal to discover a lower-rank approximation of the denoised speech. The NMF part of the system works, because its job is to re-synthesize the speech estimate with a fixed set of clean NMF bases.

[3] In this section we use terminologies from NMF-based models without the loss of generality in the other latent variable models.



(a) The top AE

(b) The stacked adaptive DAE

(c) The bottom DAE

Fig. 2: The proposed system and DNNs as its building blocks.

If we consider NMF as an AE with a single hidden layer, we can expand the two-stage approaches by replacing the NMF module with a deeper AE trained from the clean source. However, the series of runs of DNN and NMF do not guarantee the adaptation we seek in this work, since the second stage merely works as a post-processor, and there is no chance to fine-tune the main DNN-based separation module.

3 The Proposed Adaptive Denoising Autoencoders

We propose the adaptive source separation system in this section. First, we will present the network structure and the test-time error function we define in

Section 3.1. Then, we provide the details about the network settings in Section 3.2.

3.1 The Proposed Network Structure

There are two DNNs in the proposed adaptive DAE system. First, the bottom DAE is trained to take a mixture input magnitude spectrum $|\mathbf{X}(:,t)|$ and produce an output spectrum $\mathbf{Y}(:,t)$, which we express in the matrix notation as follows[4]:

$$\mathbf{Y} = g(\mathbf{W}_{DAE}^{L+1} \cdots g(\mathbf{W}_{DAE}^{2} \quad \cdot g(\mathbf{W}_{DAE}^{1} \cdot \mathbf{X}))), \tag{1}$$

where $\mathbf{W}_{DAE}^{l}$ is the weight matrix between l-th hidden layer units and their input. Bias terms are absolved into $\mathbf{X}$ as an additional row of 1's. $g(\cdot)$ denotes a nonlinearity function. We also use the MATLAB® notation for a column vector in a matrix, e.g. $\mathbf{X}(:,t)$. Since the input vectors $\mathbf{X}$ are the mixtures of multiple source spectra, i.e. $\mathbf{X} = \mathbf{S}^{(1)} + \mathbf{S}^{(2)}$, this network can be a DAE if we set up an error function with respect to the desired source, e.g. $\mathbf{S}^{(1)}$, as the target,

$$\mathcal{E}_{DAE} = \frac{1}{2} \sum_{f,t} \left(\mathbf{Y}(f,t) - \mathbf{S}^{(1)}(f,t) \right)^2. \tag{2}$$

Fig. 2 (c) depicts this procedure. In the usual DAE-based separation scenario, the separation is done by a single forward pass on this trained network.

We additionally define the top AE that encodes the identity mapping between the input and the output for the desired source. It basically has the same structure with the bottom DAE (See Fig. 2 (a)) except that it takes the spectra of the desired source $\mathbf{S}^{(1)}$ as input, and tries to minimize the error between its output $\mathbf{U}$ and the input $\mathbf{S}^{(1)}$ as its target:

$$\mathbf{U} = g(\mathbf{W}_{AE}^{L+1} \cdots g(\mathbf{W}_{AE}^{2} \quad \cdot g(\mathbf{W}_{AE}^{1} \cdot \mathbf{S}^{(1)}))), \tag{3}$$

$$\mathcal{E}_{AE} = \frac{1}{2} \sum_{f,t} \left(\mathbf{U}(f,t) - \mathbf{S}^{(1)}(f,t) \right)^2. \tag{4}$$

Having this additional AE as a purity checker, we feed $\mathbf{Y}$ to it as an input. It will give us a smaller AE error $\mathcal{E}_{AE}$ if its input $\mathbf{Y}$ has produced a smaller DAE error $\mathcal{E}_{DAE}$ than bigger. It means the bottom DAE did its job well. On the other hand, we should expect a bigger $\mathcal{E}_{AE}$ value if $\mathbf{Y}$ was significantly different from $\mathbf{S}^{(1)}$. Using this concept, we can judge the quality of $\mathbf{Y}$ by checking on $\mathcal{E}_{AE}$ during the test-time without any help from the ground truth. Moreover, we can fine-tune the bottom DAE so that it can further reduce the error between its output $\mathbf{Y}$ and the top AE output $\mathbf{U}$,

$$\mathcal{E}_{AE} = \frac{1}{2} \sum_{f,t} \left(\mathbf{U}(f,t) - \mathbf{Y}(f,t) \right)^2, \tag{5}$$

[4] We drop the absolute function $|\cdot|$ from now on for brevity, but the readers should be aware that mixing in the time domain does not hold in the magnitude Fourier domain.

which is nothing but the AE error. At every epoch, we backpropagate this error to update the bottom DAE to better separate the unseen mixture probably with unfamiliar sources and mixing environment. Finally, we construct a stacked adaptive DAE as in Fig. 2 (b), whose bottom and top parts are the DAE to be fine-tuned and the AE as a purity checker, respectively.

3.2 The Proposed Network Setting

For both AEs, we train dropout networks [9] with dropout rates 0% for the input and 20% for the other hidden units. We use momentum with parameter 0.95. The main optimization is done by Stochastic Gradient Descent (SGD) with initial step size set to be 10^{-6}. Weights are bound to be between -1 to 1. Once we train both AEs, we first feedforward the new test mixture in the bottom DAE. With its output $\mathbf{Y}$, we do another feedforward in the top AE to get $\mathbf{U}$. Then, we calculate the AE error as in (5) to fine-tune the bottom DAE. We found that the same optimization setting works well for this fine-tuning job, too. The activation is a modified maxout function as suggested in [5],

$$g(x) = \begin{cases} x & \text{if} \quad x \geq \epsilon \\ \frac{-\epsilon}{x-1-\epsilon} & \text{if} \quad x < \epsilon \end{cases} . \tag{6}$$

4 Numerical Experiments

4.1 Speech and Noise

We train the bottom DAE with 10 random female speakers from TIMIT training data, each of which has 10 utterances. Thy are mixed with four different noise types used in [5], i.e. "Babble", "Airport", "Train" and "Subway." Eventually, there are 400 noisy utterances for training, which amount to 80,864 frames after short-time Fourier transform with 1024 pt of the frame size and a 75% overlap. A square-root of Hann window is used for both analysis and synthesis. With the same setting, we also train an AE, but on the clean speech spectra as its input and target. Both networks have two hidden layers with 2048 hidden units per each layer. As for the test signals, we randomly choose 5 female speakers from the TIMIT test part, and add them up with eight different noise types: "Piano", "Drill", "Bus", "Birds", "Computer keyborad", "Frogs", "Machinegun", and "Street" (400 noisy utterances). We try two input Signal-to-Noise Ratio (SNR) choices: 0 and -5dB. Note that since the networks are trained on 0dB mixtures only, the -5dB test mixtures are more difficult for them to separate. Finally, we either use the spectrum as it is or after concatenating five consecutive frames to test the network with inputs with temporal structures.

The bottom DAE is not perfect, because we deliberately chose different kinds of test noise. Fine-tuning the bottom DAE is to reduce the top AE error in (5). By doing so, we get better DAE outputs $\mathbf{Y}$, in the sense of reducing the level of interfering sources, but in the meantime the recovered speech can also lose

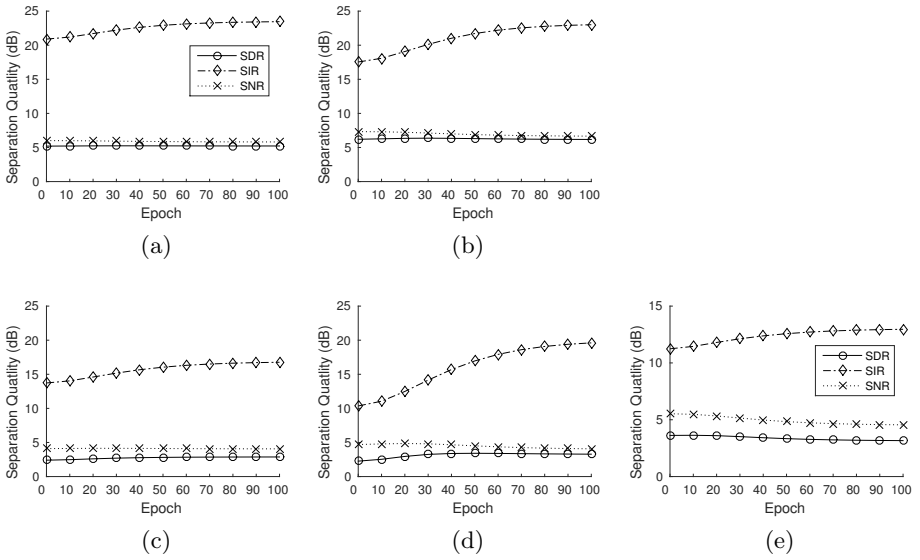


Fig. 3: The separation performance of speech enhancement experiments with different settings (a) Frame-by-frame; input SNR 0dB (b) 5 frames; input SNR 0dB (c) Frame-by-frame; input SNR -5dB (d) 5 frames; input SNR -5dB. (e) The performance on singing voice separation task.

some of its energy. Fig. 3 shows that the average SIR values increase as we keep fine-tuning, where 0-th epoch means no fine-tuning has been done. Since the SAR decreases more slowly, there is a better SDR value after several epochs in all cases. In (d), where the test mixture was -5dB and the input is the vectorized 5 frames, both SIR and SDR improvements are most significant.

4.2 Singing Voice Separation

MIR-1K is a dataset with a thousand karaoke clips played by 19 amateur singers [2]. We followed the basic setting in [8] where only 175 clips from two singers are allowed to be used as the training set, while there are 825 test clips available. We consider the voice part as our desired source, and train the networks with three hidden layers and 2048 units per a layer. Three frames are vectorized to form an input. This lack of training data makes the top AE less reliable. However, we can still see in Fig. 3 (e) that the proposed fine-tuning scheme can improve the average SIR values for the test samples.

5 Conclusion

We developed an adaptive source separation system, which consists of a bottom DAE and a top AE for the main separation module and a purity checker for the

fine-tuning job, respectively. Although a good target variables are not usually available for the use during the test-time separation, we found that this additional well-trained AE on the clean spectra of the desired source can provide the main separation module with some alternative quality measurements.

References

1. Duan, Z., Mysore, G.J., Smaragdis, P.: Online PLCA for real-time semi-supervised source separation. In: Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA). pp. 34–41 (2012)
2. Hsu, C.L., Jang, J.S.: On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset. IEEE Transactions on Audio, Speech, and Language Processing 18(2)
3. Huang, P., Kim, M., Hasegawa-Johnson, M., Smaragdis, P.: Deep learning for monaural speech separation. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (May 2014)
4. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems (NIPS). vol. 13. MIT Press (2001)
5. Liu, D., Smaragdis, P., Kim, M.: Experiments on deep learning for speech denoising. In: Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech) (Sep 2014)
6. Raj, B., Smaragdis, P.: Latent variable decomposition of spectrograms for single channel speaker separation. In: Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics. pp. 17–20 (2005)
7. Salakhutdinov, R., Hinton, G.: Semantic hashing. International Journal of Approximate Reasoning 50(7), 969 – 978 (2009)
8. Sprechmann, P., Bronstein, A., Sapiro, G.: Real-time online singing voice separation from monaural recordings using robust low-rank modeling. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR) (2012)
9. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15(1), 1929–1958 (Jan 2014)
10. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 1096–1103 (2008)
11. Wang, Y., Wang, D.L.: Towards scaling up classification-based speech separation. IEEE Transactions on Audio, Speech, and Language Processing 21(7), 1381–1390 (July 2013)
12. Williamson, D.S., Wang, Y., Wang, D.L.: Reconstruction techniques for improving the perceptual quality of binary masked speech. Journal of the Acoustical Society of America 136, 892–902 (2014)
13. Xu, Y., Du, J., Dai, L.R., Lee, C.H.: An experimental study on speech enhancement based on deep neural networks. IEEE Signal Processing Letters 21(1), 65–68 (2014)