COMPLEX RECURRENT NEURAL NETWORKS FOR DENOISING SPEECH SIGNALS

Keiichi Osako[1,2]*, Rita Singh*[1]*, and Bhiksha Raj*[1]

1. Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA
{osakok, rsingh, bhiksha}@cs.cmu.edu
2. Sony Corporation, Minato-ku, Tokyo 108-0075, Japan

ABSTRACT

Effective denoising of noise-corrupted speech signals remains a challenging problem. Existing solutions typically employ some combination of noise estimation and noise elimination, either by subtraction or by filtering. The estimation of noise and the denoising are generally treated as independent aspects of the problem. In this paper we propose a new neural-network-based approach for denoising of speech signals. The approach integrates noise estimation and denoising into a single network design, while maintaining many of the aspects of conventional noise estimation and signal denoising through a recurrent gated structure. The network thus operates as a single integrated process that can be trained to jointly estimate noise and denoise the speech signal with minimal artifacts. Noise reduction experiments on noisy speech, both with digitally added synthetic noise and real car noise, show that the proposed algorithm can recover much of the degradation caused by the noise.

***Index Terms*—** Recurrent neural networks, Speech enhancement, Noise reduction

1. INTRODUCTION

The problem of enhancing speech that has been corrupted by noise has continued to receive the attention of researchers for several decades. The challenges are twofold: the noise that corrupts any particular segment of speech must be accurately estimated, and this noise must be effectively attenuated without diminishing the speech in the signal as well.

The first problem arises from the fact the noise in the noisy speech is itself "corrupted" by the speech and cannot be accurately characterized. It must be estimated primarily from past samples of the signal, particularly regions where there is no speech [1]. One must, of course, be able to accurately detect the presence of speech in order for this to work well, and this can be a challenging problem by itself in many situations. It is common to utilize some form of a running estimate of the noise, which is updated relatively quickly in regions of no speech, and either not at all, or at a much lower rate, in regions where speech has been found, possibly with additional, possibly non-linear smoothing to interpolate between long and short-term trends [2].

The latter problem – that of optimal attenuation or cancellation of the noise in the noisy speech signal arises from the fact that we usually only have an *estimate* of the power or magnitude spectrum of the noise, and must nevertheless cancel the noise from the speech signal itself. The simplest approach to this is to simply subtract the estimated noise power from the power in the noisy speech [3]. Alternately, one may cast this is a filtering process. The optimal filter for such filtering is a Wiener filter, whose filter response is the ratio of the estimated power spectrum of the clean speech and that of the noisy speech [4]. Variations of this scheme usually differ in the manner in which the power (or magnitude) spectra of the clean and noisy speech are estimated. Still other methods develop these filters as MMSE estimators based on assumed distributions for the speech and the noise [5, 6].

The above approaches generally make few assumptions of knowledge of the properties of the underlying speech or noise. Other methods make more detailed assumptions about the distributions of the speech and/or the noise. For instance signal-separation techniques such as those based on non-negative matrix factorization [7] or PLSA [8] assume that non-negative compositional factors of both the speech signal and the corrupting noise are known. Kalman or particle filtering approaches assume knowledge of the dynamics of the speech signal or the noise [9]. Yet other approaches assume knowledge of the dynamics of both sources [10]. In all of these cases, the statistical nature of the speech and the noise must be learned from prior training examples.

Meanwhile, neural network based speech enhancement techniques have generally taken the form of mappings that attempt to directly convert noisy speech to clean speech. These mappings are often modeled by simple, time-invariant networks such as feed forward networks, autoencoders, and restricted Boltzmann machines [11, 12, 13]. The time-invariant structure does not explicitly take advantage of the time-series nature of the speech signal. Recurrent neural networks formalisms too have been proposed as an alternative, where the network mapping the noisy speech to clean speech includes a recurrent component that carries over state information [14]. However, even here, the underlying processing, although recurrent, is fixed, in that the manner in which the internal states are updated and the signal is processed remains unchanged regardless of the nature of the incoming signal. In any stream of incoming speech, the signal includes both non-speech segments and segments with speech; this distinction is ignored by the processing.

In our work we also employ a recurrent network formulation, however, we explicitly take into account the fact that the incoming signal may or may not include speech. The manner in which the signal must be processed must ideally depend on whether the incoming signal does have speech or not – in the former case, the noise must be filtered out of the signal; in the later, no output is desired; nevertheless the incoming signal may be used to update the internal state of the system that carries information about the noise. We embody this principle into our network by fashioning it upon prior formalisms such as those employed in spectral subtraction or Wiener filtering approaches — we partition our network into three sections, one each designed to track noise (or an internal variable that is analogous to noise) and the speech and a third to perform filtering. Updates to the states of these sections are turned on or off by gates, such as those employed in the long short-term memory (LSTM) principle [15, 16].

One may view the network, in principle, as a variant of a spectral subtraction scheme in which the recurrent estimate of noise and speech, as well as the final filter are all performed by neural networks, rather than simple linear or piecewise-linear functions, and the decision of when to perform the recurrent updates of noise and speech spectra too are governed by a neural network.

Noise reduction experiments on noisy speech demonstrate that the network is able to reduce noise without the spectral holes and musical noise common to most subtraction and filtering schemes. The denoising is observed, not surprisingly to improve on an equivalent implementation of spectral subtraction, indicating that simply modifying existing signal processing structures by compute parameters that are otherwise heuristically obtained through a neural network can provide benefits.

The learned strategy also enables easy transition to *complex* spectra. Conventional spectral subtraction and Wiener filtering operate on magnitude or power spectra, and make no modification to the phase of the signal. The learned neural networks can, however, both incorporate complex values, and be optimized to minimize error in the complex spectrum. We show that this approach does indeed result in improved performance over magnitude spectra.

More importantly, unlike conventional neural-net based sound processing schemes, which attempt to directly estimate the "cleaned" signal, we utilize a different approach where we attempt to learn a *multiplicative* filter that minmizes output error. We believe that this paradigm is a marked novelty that is potentially promising in many problems where multiplicative corrections are more appropriate than simple feed-forward processing.

The rest of this paper is organized as follows. In section 2 we explain the design of the denoising neural networks and present its key formulations for learning. We present our experimental setup and results in section 3. Lastly, we present our conclusions in section 4.

2. THE DENOISING NEURAL NETWORKS

2.1. The basic algorithm: denoising the magnitude spectrum

We base our neural network denoiser on a generalization of optimal ℓ_2 denoising schemes that have previously proven to be effective. In keeping with prior work on speech denoising, we will work with *magnitude* spectral values. We operate on the magnitudes of the short-time Fourier transform derived from the signal. As is well known, the short-time Fourier transform of a signal is derived from the discrete Fourier transform of sliding *frames* of audio, so in the following discussion, all references to time actually refer to frame index. All references to frequency refer to frequency indices of the discrete Fourier transform of the frames.

We assume a simple recurrent formulation for the estimation of the noise, that follows closely from the noise-estimation formulation used in spectral-subtraction like techniques. We make some simple assumptions:

- In regions of no speech, the recorded signal only contains the noise.
- The noise maintains some spectral continuity across frames.

This leads to the following recurrent model for noise:

$$|\mathbf{N}(t)| = \mathbf{g}_1(t) \otimes |\mathbf{N}(t-1)| + \mathbf{g}_2(t) \otimes |\mathbf{X}(t)| \qquad (1)$$

where $\mathbf{N}(t)$ is the estimate of the noise spectrum at time t, and $\mathbf{X}(t)$ is the spectrum of the recorded signal at time t. The model updates the estimates of the noise at any time as a linear combination of the past estimate and the current incoming signal. The terms $\mathbf{g}_1(t)$ and $\mathbf{g}_2(t)$ represent *gates* to determine *when* the two components, $\mathbf{N}(t-1)$ and $\mathbf{X}(t)$ must contribute to the current estimate. The symbol $\otimes$ represents Schur product, since $\mathbf{g}_1$, $\mathbf{g}_2$, $\mathbf{N}$, $\mathbf{X}$ and $\mathbf{Y}$ are vectors. Equation 1 attempts to capture the following intuition. Ideally, in speech regions, $\mathbf{X}(t)$ cannot be trusted to contribute to the noise estimate, since it carries a significant speech component. The estimate must largely depend on the continuity of what is already known, *i.e.* $\mathbf{N}(t-1)$. On the other hand, in non-speech regions, the current recorded signal must contribute significantly to the noise estimate. The precise behavior however, depends on being able to determine exactly when speech occurs, a challenge in itself. The gates $\mathbf{g}_1(t)$ and $\mathbf{g}_2(t)$ are intended to achieve this objective. The gates in turn are neural network classifiers themselves, since they must determine from available information, which we posit as the current input and past estimates of clean speech and noise, as well as the past value of the gate itself, whether the current frame represents speech or not. The estimated noise is used to calculate the denoising filter $\mathbf{g}_3(t)$, estimating the denoised speech spectrum as below.

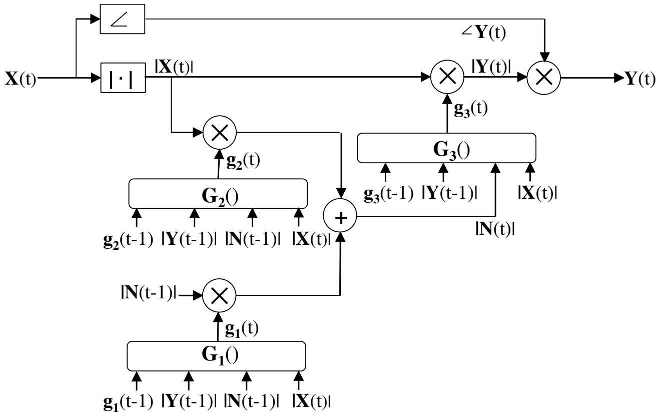

Figure 1: The gated denoising neural networks using the magnitude spectrum (MRNN). Each of the blocked symbols for $\mathbf{N}$, $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{g}_1$ and $\mathbf{g}_2$ actually represent vectors. $\mathbf{G}_1()$, $\mathbf{G}_2()$ and $\mathbf{G}_3()$ functions consist of recurrent neural networks.

$$|\mathbf{Y}(t)| = \mathbf{g}_3(t) \otimes |\mathbf{X}(t)| \qquad (2)$$

$$\mathbf{Y}(t) = |\mathbf{Y}(t)| \otimes \exp(j\angle\mathbf{X}(\omega)) \qquad (3)$$

Here, the gate $\mathbf{g}_3(t)$ above represents a denoising filter, and $\angle\mathbf{X}(\omega)$ is the phase spectrum of the observed signal $\mathbf{X}(t)$.

We incorporate the above intuitions into a recurrent neural network. Figure 1 shows a simplified illustration of the denoising neural network system. The network is recurrent: each stage of the network generates an output that depends both on the current input and the past state of the network. We may draw a linear-system analogy to an autoregressive moving-average system (although the network is decidedly not linear): the current output of the system depends not only on the past state, but also on past inputs. The "moving-average" component is derived from its dependence on $\mathbf{N}(t-1)$ and $\mathbf{Y}(t-1)$. The "autoregressive" component is derived from its dependence on $\mathbf{X}(t-\tau)$, $\tau > 0$. Figure 1 only shows an autoregressive recurrence over one past time instant; in practice longer recurrences may be employed.

The entire operation is performed on magnitude spectra. The phase in the signal is only incorporated after processing, although the actual error that is optimized during training is with respect to the complex output (*i.e.* between the complex $Y(t)$ and the complex spectrum of the corresponding clean speech). Consequently, we will refer to this network as the "MRNN", where "M" stands for "magnitude".

The actual detailed equations governing the action of the network are as follows. The gates $\mathbf{g}_1(t)$, $\mathbf{g}_2(t)$ and $\mathbf{g}_2(t)$ are given by:

$$\mathbf{g}_1(t) = \mathbf{G}_1(\mathbf{g}_1(t-1); |\mathbf{Y}(t-1)|; |\mathbf{N}(t-1)|; |\mathbf{X}(t)|), \quad (4)$$

$$\mathbf{g}_2(t) = \mathbf{G}_2(\mathbf{g}_2(t-1); |\mathbf{Y}(t-1)|; |\mathbf{N}(t-1)|; |\mathbf{X}(t)|), \quad (5)$$

$$\mathbf{g}_3(t) = \mathbf{G}_3(\mathbf{g}_3(t-1); |\mathbf{Y}(t-1)|; |\mathbf{N}(t)|; |\mathbf{X}(t)|). \quad (6)$$

Here, for brevity, we have used vector notation. Thus, $\mathbf{g}_1(t)$ and $\mathbf{g}_2(t)$ are actually vectors with F components, where F is the number of frequency bands in the STFT of the signal. $\mathbf{G}_1(\cdot)$ and $\mathbf{G}_2(\cdot)$ represent multi-layer perceptrons, each with F outputs. Similarly, $\mathbf{N}(t)$, $\mathbf{X}(t)$ and $\mathbf{Y}(t)$ are all F-dimensional vectors. The function $\mathbf{G}_3(\cdot)$ too is a multi-layer perceptron that takes in $\mathbf{Y}(t-1)$, $\mathbf{N}(t)$ and $\mathbf{X}(t)$ as inputs, and has F outputs, one per frequency component of the desired output spectrum.

The gate networks $\mathbf{G}_1(\cdot)$ and $\mathbf{G}_2(\cdot)$ are gating functions which must ideally identify regions of speech and non-speech, to control the update of the noise estimate. In practice, they make "soft" decisions between 0 and 1 to indicate the degree to which past noise and the current speech must contribute to the noise estimate. Consequently, the activations of the neurons in the gate networks are chosen to be squashing functions such as the hyperbolic tan and logistic functions.

The output network $\mathbf{G}_3(\cdot)$ effectively performs a filtering operation, which must produce an estimated clean speech signal. Similar to $\mathbf{G}_1(\cdot)$ and $\mathbf{G}_2(\cdot)$, soft decisions are employed in $\mathbf{G}_3(\cdot)$.

In all cases, the neurons operate on affine combinations of their inputs. Thus, each neuron has the form $h(i_j,\ j = 1 \cdots J) = h(\sum_j w_j i_j + b)$, where $i_j,\ j = 1 \cdots J$ represent the inputs to the neuron, $h(\cdot)$ represents the activation function of the neuron, and $w_j,\ j = 1 \cdots J$ and b represent the weights assigned to the inputs and the bias. The input weights and biases of the various neurons are the parameters of the network, which must be learned from training data.

2.2. Training the network

We use error back propagation through time (BPTT) to train the network. Since our network includes multiplicative gating components, the update rules are somewhat anlogous to those for long short-term memory (LSTM) networks [15, 16]. The backpropagation algorithm incorporates a feed-forward and a back-propagation phase.

Since the network is recurrent, the *initial* state of the network must be specified. In particular, $|\mathbf{N}(-1)|$ and $|\mathbf{Y}(-1)|$ must be specified. We assume that the initial few frames of input contain no speech and represent only the noise. Thus $|\mathbf{N}(-1)|$ is set to be the average of the first several frames of input, and the initial gain values $\mathbf{g}_1(-1)$ and $\mathbf{g}_2(-1)$ are set to 0.1 and 0.9. The feedforward phase subsequently proceeds through direct evaluation of the network.

In the backpropagation process, we define the error as the sum of the squared errors at all time instants. Let $|\mathbf{Y}(t)|,\ t = 1 \cdots T$ and $|\mathbf{S}(t)|,\ t = 1 \cdots T$ represent the sequences of spectral vectors of the denoised speech and clean speech components of training utterances. Here, index T represents the total number of frames of the data. The error at time instant t is given by

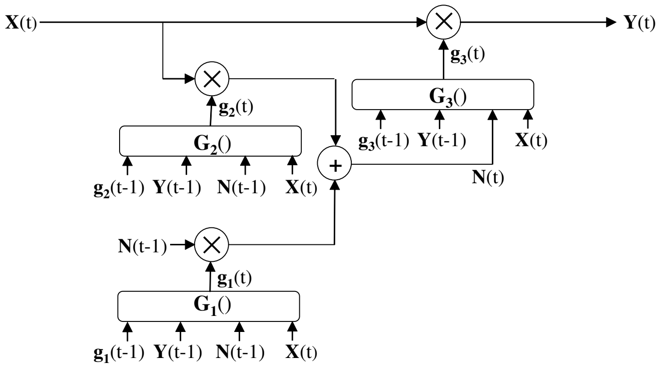

Figure 2: The gated denoising neural networks extended to complex values (CRNN).

$$E(t) = \sum_{f=1}^{F} (|S(t, f|) - |Y(t, f)|)^2. \quad (7)$$

The total error that we actually minimize is the sum of the error at all time instants:

$$E = \sum_{t=1}^{T} E(t) \quad (8)$$

The backpropagation rules minimize this error with respect to the network parameters, namely the weights and biases of the neurons.

2.3. Denoising the Complex Spectrum

In the MRNN explained in the previous section, the network operates on magnitude spectral values. In order to resynthesize the denoised speech, the phase information is carried over from the observed noisy signal $\mathbf{X}(t)$.

In our second model, we attempt to directly estimate a *complex* multiplicative correction term, by operating directly on the complex spectrum of the speech [17]. The input to the network is now the complex spectrum of the incoming speech. All weights are complex. The neurons are also complex, and operate on complex inputs to generate complex outputs: the conventional real function $y = f(x)$ specifying the activation functions is now replaced by $y_r = f(x_r)$, $y_i = f(x_i)$, $y = y_r + iy_i$.

Figure 2 shows the complex valued denoising neural networks. We refer to this network as a "CRNN", where the "C" stands for "Complex". The individual components of the extended complex denoising algorithm are described as below.

$$\mathbf{N}(t) = \mathbf{g}_1(t) \otimes \mathbf{N}(t-1) + \mathbf{g}_2(t) \otimes \mathbf{X}(t) \quad (9)$$

$$\mathbf{Y}(t) = \mathbf{g}_3(t) \otimes \mathbf{X}(t) \quad (10)$$

$$\mathbf{g}_1(t) = \mathbf{G}_1(\mathbf{g}_1(t-1); \mathbf{Y}(t-1); \mathbf{N}(t-1); \mathbf{X}(t)) \quad (11)$$

$$\mathbf{g}_2(t) = \mathbf{G}_2(\mathbf{g}_2(t-1); \mathbf{Y}(t-1); \mathbf{N}(t-1); \mathbf{X}(t)) \quad (12)$$

$$\mathbf{g}_3(t) = \mathbf{G}_3(\mathbf{g}_3(t-1); \mathbf{Y}(t-1); \mathbf{N}(t); \mathbf{X}(t)) \quad (13)$$

The objective we minimize to train the is the ℓ_2 error between the complex output of the network and the desired complex clean spectrum. Since the ℓ_2 error is not analytic and is hence not differentiable, we use the approximation from [18] to obtain the derivatives for back propagation.

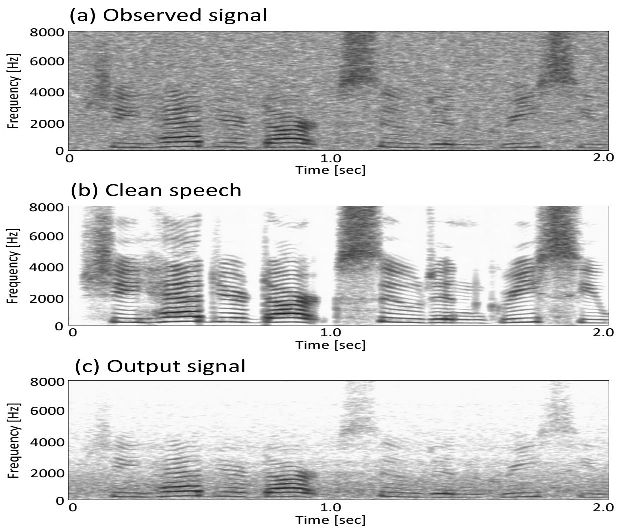

Figure 3: Log spectrogram of signals. (a) observed signal with white noise 10dB, (b) clean speech and (c) output signal by CRNN.

3. EXPERIMENTAL RESULTS

3.1. Experimental setup

For experimental evaluation, we conducted two sets of experiments. In one, we corrupted a speech database with digitally added synthetic white noise, and in the second set, we corrupted the data with noise recorded in a car. In each case we trained the network with clean and noisy pairs of utterances from the training set, and subsequently tested denoising performance by using the trained network to clean the test set. For the experiments, we used the TIMIT [19] dataset. This dataset is pre-partitioned into a training and a test set. We corrupted the dataset with noises at SNRs ranging from -5 to 15dB at intervals of 5dB SNR. Spectral vectors were computed from the clean and noisy signals using a 256-point Short Time Fourier Transform (STFT).

The key parameters of recurrent neural networks in our experiments were a) Depth of network: 3 to 4 layers, b) Breadth of network: 32 to 258 units, c) Learning rate: 0.03, d) Activation function: logistic. Also, the dimensions of the input and output layers were set to 516 and 129 respectively.

In the training phase, errors aggregated over all data in the training set were used to re-estimate the weights in each iteration (or epoch) of the training. In the testing phase, input spectral vectors from noisy signal are operated on by the trained network. Each 129-dimensional output corresponded to the spectrum of one frame of the estimated cleaned signal. From these, we reconstruct the clean signal through an inverse short-time Fourier transform.

We evaluate the denoising performance in terms of signal-to-distortion ratio (SDR) [20], which computes the ratio of the signal energy in the true (clean) signal to the energy in the distortion between the true and reconstructed signals. For comparison, traditional spectral subtraction [3] was employed, using an empirically determined over-subtraction factor of 2.0 and noise floor of 0.01.

3.2. Results

Table 1 shows the denoising performance using various hidden units parameters. White gaussian noise at 5dB SNR was used in this experiment. The best results obtained are 9.51dB SDR with 129 units in the CRNN. In all patterns of various hidden units, the CRNN is better than the MRNN.

Hidden Layer	SDR [dB]	
size	MRNN	CRNN
32	9.26	9.50
32-32	9.27	9.50
64	9.29	9.51
64-64	9.32	9.50
129	9.32	9.51
258	9.29	9.51

Table 1: Denoising performance for various hidden units on TIMIT testset with white gaussian noise at 5dB SNR.

Table 2 shows the results the denoising performance for various SNRs on the TIMIT testset. The hidden-layer sizes of MRNN and CRNN are set to 129 units. In this result too, the proposed algorithms are significantly better than conventional spectral subtraction for both, white and car noises.

Noise	Input SNR	SDR [dB]		
type	[dB]	SS	MRNN	CRNN
White	-5	3.98	4.03	4.04
	0	5.77	6.47	6.61
	5	7.61	9.32	9.51
	10	9.32	12.48	12.71
	15	10.86	16.09	16.36
Car	-5	4.49	6.90	7.12
	0	6.72	10.01	10.40
	5	8.61	13.39	13.75
	10	10.22	16.79	17.23
	15	11.59	20.30	20.97

Table 2: Denoising performance for various SNR on TIMIT testset. Hidden units size of MRNN and CRNN are 129.

Figure 3 shows the spectrogram of a noisy signal corrupted to 10dB by white noise, the reference clean speech and the output signal of a CRNN. It should be noted that the high frequency component of consonants is not destroyed by the process, nevertheless high frequency noise is reduced clearly. The noise in the low-frequency bands has remained. We believe that this too can be eliminated by appropriate modification of the objective minimized to train the network.

4. CONCLUSIONS

In this paper, we have proposed the use of gated denoising recurrent neural networks as neural equivalents of spectral subtraction for speech enhancement. The results of the noise reduction experiments revealed that the noise reduction performance of the proposed algorithm was superior to that of conventional spectral subtraction. We have also extended the algorithm into complex-valued recurrent neural networks. Experimental results demonstrated that the complex-valued network was superior to its real-valued counterpart. While better results may be obtained through more sophisticated models, we believe that the chief take-home lesson is that even conventional signal-processing algorithms may benefit significantly through direct translation into analogous neural-net formalisms.

5. REFERENCES

[1] S. Jongseo, K. N. Soo, and S. Wonyong, "A statistical model-based voice activity detection," *Signal Processing Letters, IEEE*, vol. 6, no. 1, pp. 1–3, 1999.

[2] P. Lockwood, J. Boudy, and M. Blancher, "Non-linear spectral subtraction (nss) and hidden markov models for robust speech recognition in car noise environments," in *Proc. IEEE ICASSP*, vol. I, 1992, pp. 265–268.

[3] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. on Acoustic, Speech and Signal Processing*, vol. 27, pp. 113–120, 1979.

[4] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.

[5] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 32, pp. 1109–1121, 1984.

[6] ——, "Speech enhancement using minimum mean square log spectral amplitude estimator," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 33, pp. 443–445, 1985.

[7] M. N. Schmidt, O. Winther, and L. K. Hansen, "Bayesian non-negative matrix factorization," in *Independent Component Analysis ans Signal Separation*, 2009, pp. 540–547.

[8] P. Smaragdis, "From learning music to learning to separate," in *Forum Acusticum*, 2005.

[9] R. Singh and B. Raj, "Tracking noise via dynamical systems with a continuum of states," in *Proc. ICASSP*, 2003.

[10] A. P. Varga and R. K. Moore, "Hidden markov model decomposition of speech and noise," in *Proc. ICASSP*, 1990, pp. 845–848.

[11] T. Fechner, "Nonlinear noise filtering with neural networks: comparison with weiner optimal filtering," *Proceedings of third International Conference of Neural Networks, IEEE Conference Publication*, pp. 143–147, 1993.

[12] X. G. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising auto-encoder," in *Proc. Interspeech*, 2013, pp. 436–440.

[13] Y. Xu, J. Du, and C. H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2015.

[14] F. Weninger, J. R. Hershey, J. L. Roux, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *Proc. IEEE GlobalSIP 2014 Symposium on Machine Learning Applications in Speech Processing*, 2014.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] J. F. Gers and J. Cummins, "Learning to forget: continual prediction with lstm," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[17] T. Nitta, "An extension of the back-propagation algorithm to complex numbers," *Neural Networks*, vol. 10, no. 8, pp. 1391–1415, 1997.

[18] H. G. Zimmermann, A. Minin, and V. Kusherbaeva, "Comparison of the complex valued and real valued neural networks trained with gradient descent and random search algorithms," in *European Symposium on Artificial Neural Networks*, 2011.

[19] J. Garofolo, "Timit acoustic-phonetic continuous speech corpus ldc93s1," *Web Download. Philadelphia: Linguistic Data Consortium*, 1993.

[20] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.