2018 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 17–20, 2018, AALBORG, DENMARK

MONAURAL SPEECH SEPARATION USING A PHASE-AWARE DEEP DENOISING AUTO ENCODER

Donald S. Williamson

Department of Computer Science, Indiana University, USA
williads@indiana.edu

ABSTRACT

Traditional deep denoising autoencoders (DDAE) use magnitude domain features and training targets to separate speech from background noise. Phase enhancement, however, has recently been shown to improve perceptual and objective speech quality. We present an approach that uses a DDAE to estimate phase-aware training targets from phase-aware input features. This network is denoted as a phase-aware deep denoising autoencoder (paDDAE). The short-time Fourier transform (STFT) of noisy speech is the network input, and the network estimates a phase-aware time-frequency mask. The proposed approach is evaluated across multiple conditions, including various signal-to-noise ratios (SNRs), noise types, and speakers. The results show that the paDDAE offers improvements over traditional DDAEs in terms of objective speech quality and intelligibility.

Index Terms— Deep denoising autoencoders, phase enhancement, speech separation

1. INTRODUCTION

Monaural speech separation remains a challenging problem, where the main objective is to remove background noise from targeted speech. Statistical signal processing [1] and computational auditory scene analysis (CASA) [2] have traditionally been used to address this problem. Of late, deep learning through various deep neural networks (DNNs) has been used to address speech separation and it has resulted in tremendous gains.

Deep-learning based speech separation can largely be grouped into two categories: mask-based estimation and signal-based estimation, where each approach operates in the time-frequency (T-F) domain. In [3], a DNN estimates the ideal binary mask (IBM), which is a two-dimensional matrix that defines whether a T-F unit is speech dominant (value of 1) or noise dominant (value of 0). More recently, an IBM is estimated using deep clustering (DC), which functions for arbitrary source signals [4]. A binary mask often results in degraded perceptual quality, so alternative T-F masks are used. In [5], a DNN estimates the ideal ratio mask (IRM), which is a soft mask that returns values between 0 and 1 inclusively. The IRM indicates the percentage of speech energy at each T-F unit. Recurrent neural networks have also been used to estimate the IRM [6, 7]. Instead of estimating a T-F mask, other approaches directly estimate the clean magnitude spectra using feed-forward or recurrent neural networks [8, 9, 10].

Deep denoising autoencoders (DDAE) have also been used for monaural speech separation [11, 12, 13, 14, 15]. DDAEs consist of encoder and decoder stages. The encoder maps the input noisy magnitude spectra into a hidden representation using a DNN. The decoder then maps the hidden representation to an estimate of the clean magnitude spectra. Many deep denoising autoencoders have been proposed in the past, where differences amount to variations in input features, training targets, and network architectures. In [12, 15], DDAEs learn a mapping from noisy Mel-frequency cepstral coefficients (MFCCs) to their corresponding clean versions. They also use pre-training with restricted Boltzmann machines (RBMs) to initialize the network parameters. An ensemble of denoising autoencoders are used in [14], where the inputs and outputs are defined in the Mel-spectral magnitude domain. Shivakumar and Georgiou use log-power spectral features as inputs to their denoising decoder, along with a perceptually-inspired cost function [16]. Power magnitude spectrums are also used as inputs and outputs in [11].

Whether DDAEs or DNNs are used, the above approaches typically have one thing in common, they operate on the magnitude spectra and use the phase from the noisy signal to reconstruct an estimate of the clean speech. This is problematic because recent studies have shown that phase is important for perceptual speech quality [17], where signal processing [18, 19, 20] and DNN-based approaches [21, 22, 23] have confirmed this finding.

In this paper, we use DDAEs for phase-aware speech separation. More specifically, we train a deep autoencoder (DAE) to learn a mapping from noisy complex spectra to noisy complex spectra, where only the hidden representations from the DAE calculation are retained. A DDAE is then trained to map the hidden representations of the DAE to phase-aware training targets. In this study, the complex ideal ratio mask (cIRM) is used as the phase-aware training target of the DDAE, since

978-1-5386-5477-4/18/$31.00

it has been shown to outperform other training targets in perceptual and objective quality evaluations [22]. The key difference between this proposed approach and all other approaches is that deep learning is used to map phase-aware features to phase-aware training targets. To the best of our knowledge, phase-aware DDAEs have not been investigated for monaural speech separation.

The rest of this paper is organized as follows. A description of a traditional deep denoising autoencoder is given in Section 2. Section 3 provides a detailed description of our proposed approach. The experiments and results are shown in Section 4. Finally, a conclusion is given in Section 5.

2. DEEP DENOISING AUTOENCODER

Autoencoders find hidden representations that can be used to estimate the input signal. More specifically, if $|\boldsymbol{X}(t)|$ is a single time frame of the noisy speech magnitude spectra, then the autoencoder uses a function $f(|\boldsymbol{X}(t)|)$ to learn a hidden representation, $\boldsymbol{h}$, for the noisy speech input

$$\boldsymbol{h} = f(|\boldsymbol{X}(t)|) = \sigma(\boldsymbol{W}|\boldsymbol{X}(t)| + \boldsymbol{b}) \tag{1}$$

where σ is the non-linear activation function of the network, and $\boldsymbol{W}$ and $\boldsymbol{b}$ are the network weight and bias terms, respectively. The decoder portion of the autoencoder learns a functional mapping from the hidden representation, $\boldsymbol{h}$, back to the input $|\boldsymbol{X}(t)|$ resulting in an estimate of the input, $|\hat{\boldsymbol{X}}(t)|$. In other words,

$$|\hat{\boldsymbol{X}}(t)| = \sigma(\boldsymbol{W}'\boldsymbol{h} + \boldsymbol{b}') \tag{2}$$

where $\boldsymbol{W}'$ and $\boldsymbol{b}'$ are the weights and bias for the decoding portion of the autoencoder. The back-propagation algorithm that minimizes the mean-square error between $|\boldsymbol{X}(t)|$ and $|\hat{\boldsymbol{X}}(t)|$ is used to compute the network parameters (e.g. $\boldsymbol{W}$, $\boldsymbol{b}$, $\boldsymbol{W}'$, and $\boldsymbol{b}'$).

The autoencoder can use multiple layers of encoding to learn refined hidden representations.

$$\begin{aligned} \boldsymbol{h}_1 &= f(|\boldsymbol{X}(t)|) = \sigma(\boldsymbol{W}_1|\boldsymbol{X}(t)| + \boldsymbol{b}_1) \\ \boldsymbol{h}_2 &= f(\boldsymbol{h}_1) = \sigma(\boldsymbol{W}_2\boldsymbol{h}_1 + \boldsymbol{b}_2) \\ &\vdots \\ \boldsymbol{h}_i &= f(\boldsymbol{h}_{i-1}) = \sigma(\boldsymbol{W}_i\boldsymbol{h}_{i-1} + \boldsymbol{b}_i) \end{aligned} \tag{3}$$

The constant i represents the number of encoding neural network layers, where $i \geq 1$ and $\boldsymbol{h}_0 = |\boldsymbol{X}(t)|$.

Denoising autoencoders use an autoencoder to map an noisy input signal into a hidden representation, and then use a denoising decoder to map the hidden representation to a clean version of the input signal. When denoising autoencoders are used, the decoding portion of the autoencoder is discarded and only the encoding portion is retained. The final hidden representation, $\boldsymbol{h}_i$, that is outputted by the autoencoder is subsequently provided as an input to the denoising decoder function, $g(\boldsymbol{h}_i)$. This function transforms the hidden representation into an estimate of the clean speech magnitude spectra, $|\hat{\boldsymbol{S}}(t)|$. As with the autoencoder, the denoising decoder can have multiple network layers as defined by j, where $j \geq 1$. This is depicted in Eq. (4), where $|\hat{\boldsymbol{S}}_0(t)| = \boldsymbol{h}_i$, and the denoising decoder's weights and bias at the j^{th} layer are denoted by $\boldsymbol{H}_j$ and $\boldsymbol{d}_j$, respectively.

$$\begin{aligned} |\hat{\boldsymbol{S}}_1(t)| &= g(\boldsymbol{h}_i) = \sigma(\boldsymbol{H}_1\boldsymbol{h}_i + \boldsymbol{d}_1) \\ |\hat{\boldsymbol{S}}_2(t)| &= g(|\hat{\boldsymbol{S}}_1(t)|) = \sigma(\boldsymbol{H}_2|\hat{\boldsymbol{S}}_1(t)| + \boldsymbol{d}_2) \\ &\vdots \\ |\hat{\boldsymbol{S}}_j(t)| &= g(|\hat{\boldsymbol{S}}_{j-1}(t)|) = \sigma(\boldsymbol{H}_j|\hat{\boldsymbol{S}}_{j-1}(t)| + \boldsymbol{d}_j) \end{aligned} \tag{4}$$

The denoising decoder's weights and bias terms are computed with the backpropagation algorithm that minimizes a cost function (e.g. mean-square error) between $|\hat{\boldsymbol{S}}_j(t)|$ and $|\boldsymbol{S}(t)|$, where $|\boldsymbol{S}(t)|$ is the magnitude spectra of the true clean signal.

3. PHASE-AWARE DEEP DENOISING AUTOENCODER

Traditional deep denoising autoencoders have two major things in common: they operate in the magnitude domain and they estimate clean spectra. Alternatively, we propose a DDAE that uses phase-aware inputs and outputs by operating in the complex domain. Our proposed approach also estimates a time-frequency mask that is subsequently used to estimate the clean spectra. Details about this approach are below.

3.1. Phase-aware deep autoencoder

Our approach begins by training a phase-aware deep autoencoder (paDAE), which uses phase-aware features and training target. More specifically, we define $\boldsymbol{X}$ as the short-time Fourier transform (STFT) of a noisy speech signal. $\boldsymbol{X}$ is a function of time, t, and frequency, f, and it is computed from the time-domain noisy speech signal, $\boldsymbol{x}$, as follows:

$$X(t,f) = \sum_{m=-\infty}^{\infty} x(m)w(t-m)e^{-\frac{j2\pi fm}{N}} \tag{5}$$

where $\boldsymbol{X} \in \mathbb{C}$ is a $F \times T$ matrix with F representing the number of frequency channels and T representing the number of time frames. w represents the time-domain windowing function and N is the length of the discrete Fourier transform (DFT). The STFT uses a complex exponential term, hence $\boldsymbol{X}$ is complex valued with real and imaginary terms (e.g. $X(t,f) = X_r(t,f) + jX_i(t,f)$). Additionally, since $\boldsymbol{X}$ is complex-valued it can be represented by its magnitude, $|\boldsymbol{X}|$, and phase, $\angle\boldsymbol{X}$, where $\boldsymbol{X} = |\boldsymbol{X}|e^{j\angle\boldsymbol{X}}$, which is often done. Obviously, coordinate conversions can convert from

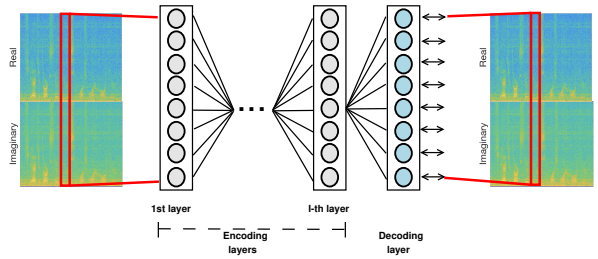

Fig. 1. Depiction of a phase-aware deep autoencoder (paDAE) with multiple encoding layers.

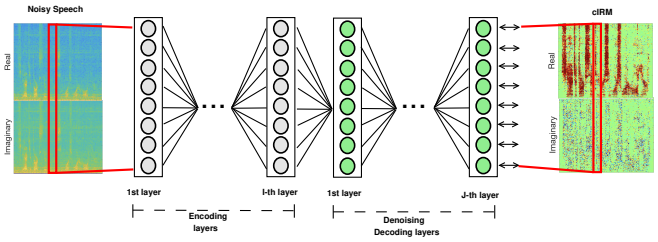

Fig. 2. Proposed phase-aware deep denoising autoencoder (paDDAE) with multiple encoding and decoding layers.

the magnitude-phase representation of $\boldsymbol{X}$ to the complex (e.g. real-imaginary) representation.

$$\begin{aligned}\boldsymbol{X}_r(t) &= |\boldsymbol{X}(t)|\cos(\angle\boldsymbol{X}(t)) \\ \boldsymbol{X}_i(t) &= |\boldsymbol{X}(t)|\sin(\angle\boldsymbol{X}(t))\end{aligned} \tag{6}$$

Typical approaches use the magnitude representation for denoising autoencoders (input feature and target) and discard the phase information. This is problematic since phase has been shown useful for speech enhancement [17]. Rather than discard the phase, we propose to include the phase information in the feature and training targets. More specifically, we train a phase-aware autoencoder that uses the stacked real and imaginary components of the noisy speech STFT, $\boldsymbol{X}$, as inputs and targets.

A depiction of the phase-aware deep autoencoder is shown in Fig 1. The approach starts by computing the real and imaginary STFT components of the noisy speech signal. The real and imaginary components, $\boldsymbol{X}_r$ and $\boldsymbol{X}_i$, are stacked together to form a feature vector, $\boldsymbol{K} = [\boldsymbol{X}_r; \boldsymbol{X}_i]$. Hence, if $\boldsymbol{X}_r$ and $\boldsymbol{X}_i$ are each $F \times T$ dimensional, then $\boldsymbol{K}$ is $2F \times T$ dimensional. Each time frame of the phase-aware feature vector (e.g. $\boldsymbol{K}(t)$) is individually supplied to the multi-layer encoder to produce hidden representations:

$$\boldsymbol{h}_i = f(\boldsymbol{h}_{i-1}) = \sigma(\boldsymbol{W}_i\boldsymbol{h}_{i-1} + \boldsymbol{b}_i), \quad 1 \leq i \leq I \tag{7}$$

where $\boldsymbol{h}_0 = \boldsymbol{K}(t)$, and I is the total number of layers in the encoder. A decoding layer is also used, which maps the final hidden representation, $\boldsymbol{h}_I$, to the stacked real and imaginary components of the noisy speech, $\boldsymbol{K}(t)$.

$$\hat{\boldsymbol{K}}(t) = \sigma(\boldsymbol{W}'\boldsymbol{h}_I + \boldsymbol{b}') \tag{8}$$

3.2. Phase-aware denoising autoencoder

The proposed approach first uses the phase-aware autoencoder to generate a hidden representation, $\boldsymbol{h}_I$, that appropriately represents the phase-aware spectral information. The decoding stage of the phase-aware autoencoder is discarded, hence Eq. (8) is not used once the phase-aware autoencoder is trained. The encoding weights and bias terms (e.g. $\boldsymbol{W}_i$ and $\boldsymbol{b}_i$, $\forall i$, $1 \leq i \leq I$) generate inputs for the denoising decoder. This functionality is depicted in Fig. 2.

Given phase-aware features (e.g. real and imaginary components of the noisy speech STFT), the encoder first generates a hidden representation. Next, a denoising decoding stage, that may consist of multiple neural network layers, uses the hidden representation to estimate a phase-aware training target. Typical, approaches use the clean speech magnitude spectra as a training target, but in this approach, we elect to use the phase-aware complex ideal ratio mask (cIRM) as the training target. The cIRM is a time-frequency mask that can be used to filter noise from speech. Given the noisy speech STFT, $\boldsymbol{X}$, and the clean speech STFT, $\boldsymbol{S}$, the cIRM is computed as follows:

$$\boldsymbol{M}(t) = \frac{\boldsymbol{S}(t)}{\boldsymbol{X}(t)} = \frac{\boldsymbol{S}_r(t) + j\boldsymbol{S}_i(t)}{\boldsymbol{X}_r(t) + j\boldsymbol{X}_i(t)} = \frac{|\boldsymbol{S}(t)|}{|\boldsymbol{X}(t)|}e^{j(\angle\boldsymbol{S}(t) - \angle\boldsymbol{X}(t))} \tag{9}$$

where $|\boldsymbol{S}(t)|$ and $|\boldsymbol{X}(t)|$ denote the magnitude responses of the clean and noisy speech at time frame t, while $\angle\boldsymbol{S}(t)$ and $\angle\boldsymbol{X}(t)$ denote the phase responses of the clean and noisy speech STFTs, respectively. For this project, we denote $\boldsymbol{M}(t)$ as the stacked real and imaginary components of the cIRM. We elect to use a T-F mask for denoising, rather than the clean spectra, since recent work has shown that DNNs estimate T-F masks better than magnitude spectra [5].

The phase-aware denoising decoding stage learns a mapping from the hidden representation to the cIRM. This is accomplished in a layer-wise manner similar to (4), but with the following differences.

$$\begin{aligned}\hat{\boldsymbol{M}}_1(t) &= g(\boldsymbol{h}_i) = \sigma(\boldsymbol{H}_1\boldsymbol{h}_i + \boldsymbol{d}_1) \\ \hat{\boldsymbol{M}}_2(t) &= g(\hat{\boldsymbol{M}}_1(t)) = \sigma(\boldsymbol{H}_2\hat{\boldsymbol{M}}_1(t) + \boldsymbol{d}_2) \\ &\vdots \\ \hat{\boldsymbol{M}}_j(t) &= g(\hat{\boldsymbol{M}}_{j-1}(t)) = \sigma(\boldsymbol{H}_j\hat{\boldsymbol{M}}_{j-1}(t) + \boldsymbol{d}_j)\end{aligned} \tag{10}$$

In Eq. (10), $\hat{\boldsymbol{M}}_j(t)$ denotes the time-frame level estimate of the stacked real and imaginary cIRM components after processing with the j^{th} phase-aware denoising decoding layer. The number of decoding layers (e.g. j) various from 1 to J. Each $\hat{\boldsymbol{M}}_j(t)$ is an $2F$-dimensional vector that represents the

frequency response of the cIRM at time frame t. The first level estimate is computed by using the first layer weights, $\boldsymbol{H}_1$, bias $\boldsymbol{d}_1$ of the denoising decoding stage and the hidden representation from the encoding stage, $\boldsymbol{h}_I$, to perform an affine mapping followed by a non-linear activation function. Estimates using additional decoding layers are computed in a similar manner, but using the estimate of the prior layer as input and with a different set of weights and bias terms.

Once the cIRM mask is estimated, it is unstacked, represented as a complex number, and subsequently applied in an element-wise fashion to the noisy speech STFT, to generate an estimate of the clean speech STFT (e.g. $\hat{\boldsymbol{S}}(t) = \hat{\boldsymbol{M}}_J(t) \odot \boldsymbol{X}(t)$), where complex multiplication is performed at each frequency bin. Overlap and add synthesis is then used to generate a time-domain estimate.

4. EXPERIMENTS AND RESULTS

Our approach is evaluated on the TIMIT speech corpus [24], which consists of utterances from 630 male and female speakers. Training data is generated by combining 500 utterances (10 utterances from 50 speakers), with five noise signals (speech-shaped noise, cafeteria, babble, tank and engine) at signal-to-noise ratios (SNRs) of -5, 0, and 5 dB. Random segments of the first half of each noise are used to generate the noisy speech mixtures. A development set is also used for model selection via early stopping. The development set is generated from 60 different clean speech utterances, and random segments of the first half of each of the above noises at the defined SNRs. 1500 testing signals are used to evaluate the approach, where the signals are generated from 60 different clean speech utterances, random segments of the above five noises, using 5 different SNRs (-10, -5, 0, 5, and 10 dB), hence two SNRs are unseen during training. All signals are downsampled to 16kHz sampling rates prior to any computations.

We first evaluate the performance of the phase-aware autoencoder to generate an estimate of the noisy speech signal. In order to accomplish this, we vary the number of encoding layers, I, from 1 to 7 to determine the impact the number of layers has on autoencoding-decoding performance. Each hidden layer has 1024 neurons with rectified linear (ReLU) activation functions. Linear activation functions are used in the output layer. The mean-square cost function is also used. The STFT of the noisy speech signal is computed with a 32ms window with a 8ms hopsize. A 512-point fast Fourier transform (FFT) is used as well. As described in Section 3.1, the real and imaginary components of the STFT are extracted and stacked on top of each other and are used as both the training input and target for the phase-aware autoencoder.

Table 1. Objective evaluation of the phase-aware autoencoder as a function of the number of encoding layers.

	number of encoding layers						
	1	2	3	4	5	6	7
PESQ	3.01	2.91	2.84	2.80	2.75	2.70	2.67
STOI	0.7	0.66	0.58	0.55	0.51	0.46	0.43

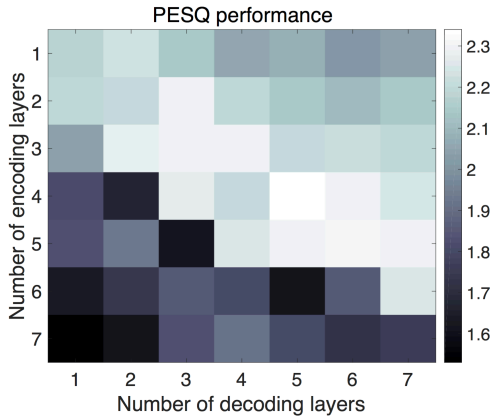

Fig. 3. PESQ performance as a function of the number of encoding layers and denoising decoding layers for paDDAE.

Table 1 shows the average objective performance of the phase-aware autoencoder (paDAE) for different number of encoding layers, where the perceptual evaluation of speech quality (PESQ) [25] and the short-time objective intelligibility (STOI) [26] are used for evaluation. Hence, we are testing the paDAE's ability to estimate the noisy speech STFT, and we are not performing denoising. PESQ and STOI each require a reference signal, and in this case the reference signal is the unprocessed noisy speech signal. As you can see by the results, the objective performance gradually degrades as the number of encoding layers increases.

Autoencoding performance may not directly corresponding to denoising performance, so we evaluate our proposed approach (paDDAE) using varying numbers of encoding and decoding layers. Hence, we vary the values of I and J each from 1 to 7, respectively, which produces 49 different combinations. Figure 3 shows the average PESQ performance of the proposed approach using varying number of encoding and decoding layers. The figure shows that phase-aware denoising decoders with higher number of encoding layers (e.g. 6 and 7) do not perform well, regardless of the number of denoising decoding layers. Using a mid-range of encoding layers (3-5) produces much better results when the number of denoising decoding layers is approximately equal (or greater than) the number of encoding layers. Performance using a small number of encoding layers (1 or 2) is generally better than using a higher number of encoding layers, but worse than using a mid-level number of encoding layers. The best performance occurs when 4 encoding layers are used with 5 denoising decoding layers. Its interesting to note that the ability to repre-

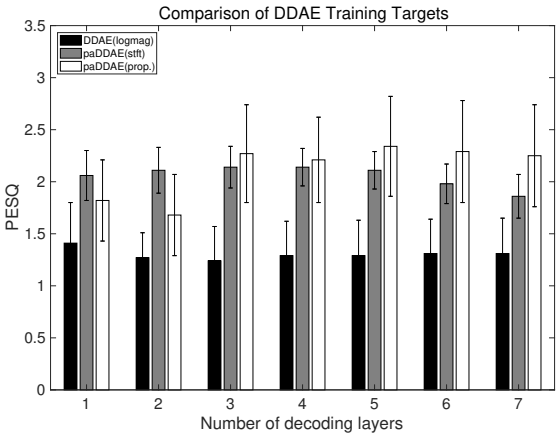

Fig. 4. Comparison of different DDAE approaches as the number of denoising decoding layers increases.

sent a noisy speech signal with the phase-aware autoencoder did not directly correspond to the ability to perform phase-aware denoising. For instance, one encoding layer gives the best autoencoding performance, but a mid-level number of encoding layers gives better denoising performance.

Lastly, we compare our proposed phase-aware DDAE to other DDAE-based speech separation approaches. Mainly we compare to a traditional DDAE, which maps noisy log-magnitude spectral inputs to clean log-magnitude spectral targets, which is similar to the approach in [16]. Once the DDAE is trained, the phase from the noisy speech signal is combined with the estimated magnitude response, to generate a time-domain signal. This approach is denoted as DDAE. We also compare against another phase-aware DDAE, but in this case, the target is the stacked real and imaginary components of the clean speech STFT. Hence, this phase-aware DDAE learns a mapping from the stacked real and imaginary components of the noisy speech STFT to the equivalents from the clean speech STFT, thus a T-F mask is not used. We denote this approach as paDDAE(stft). These two approaches are trained using the same data that is described above. Since 4 encoding layers worked well for our proposed approach, we also use 4 encoding layers for these approaches, but vary the number of decoding layers.

The average PESQ scores, with error bars (e.g. standard deviation), of all approaches are shown in Figure 4. When the number of denoising decoding layers is small (i.e. 1 or 2), the phase-aware DDAE with the STFT training target performs best. The proposed approach, however, consistently performs better than all other approaches as the number of denoising decoding layers increases. This likely occurs because more layers are needed to map the hidden representation of the noisy speech STFT to the cIRM. Notice that the traditional log-magnitude DDAE consistently performs the worse.

Table 2. Mean PESQ scores (standard deviation in parentheses) for log-magnitude DDAE using fewer layers.

		# of dec. layers		
		1	**2**	**3**
# of enc. layers	**1**	1.77 (0.4)	1.95 (0.41)	1.73 (0.41)
	2	1.34 (0.5)	1.68 (0.39)	-

A one-way analysis of variance (ANOVA) test is performed to determine statistical significance. The PESQ scores of the proposed paDDAE are separately compared with DDAE(logmag) and paDDAE(stft), where these two comparisons are performed separately for each number of decoding layers. The resulting p-values are all less than 0.05 (e.g. 5% confidence interval) indicating that the results are statistically significant for all configurations (e.g. number of decoding layers).

Note that earlier research has shown that less layers are needed for DDAEs that operate in the magnitude domain [12]. Hence, the poor performance of the log-magnitude results could be due to the usage of too many layers. To determine if this is the case, we conducted additional experiments where the number of encoding and denoising decoder layers are restricted to a combined maximum of 4 layers in total. The results for the log-magnitude DDAE are shown in Table 2. Notice that these results are much improved, but they still are not better than the results from the proposed phase-aware approach, as shown in Figure 4. These results are consistent with the results reported in [16] for log-magnitude based features and targets. A one-way ANOVA test shows that the proposed results are statistically significant from the best performing log-magnitude results.

5. CONCLUSION

We present an approach to monaural speech separation that uses a phase-aware denoising autoencoder. The approach uses the phase-aware noisy speech spectra as inputs and uses a DDAE to map to the phase-aware complex ideal ratio mask (cIRM). The results show that the proposed approach outperforms a traditional DDAE approach and a different phase-aware DDAE. Ultimately, the results reveal that phase-aware processing is important for speech separation performance, and that deep learning can successfully process phase-aware features and targets.

6. REFERENCES

[1] P. C. Loizou, *Speech Enhancement: Theory and Practice*, Boca Raton, FL, USA: CRC, 2007.

[2] D. L. Wang and G. Brown, Eds., *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*, Hoboken, NJ: Wiley-IEEE Press, 2006.

[3] Y. Wang and D. L. Wang, "Towards scaling up classification-based speech separation," *IEEE Trans. Audio Speech and Lang. Process.*, vol. 21, pp. 1381–1390, 2013.

[4] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: discriminative embeddings for segmentation and separation," in *Proc. ICASSP*, 2016.

[5] Y. Wang, A. Narayanan, and D. L. Wang, "On training targets for supervised speech separation," *IEEE/ACM Trans. Audio Speech and Lang. Process.*, vol. 22, pp. 1849–1858, 2014.

[6] F. Weninger, J. Le Roux, J. R. Hershey, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *Proc. IEEE GlobalSIP*, 2014.

[7] P. S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 23, pp. 2136–2147, 2015.

[8] Y. Xu, J. Du, L. Dai, and C. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Process. Letters*, vol. 21, pp. 65–68, 2014.

[9] F. Weninger, S. Watanabe, J. Le Roux, J. R. Hershey, Y. Tachioka, J. Geiger, B. Schuller, and G. Rigoll, "The MERL/MELCO/TUM system for the REVERB Challenge using deep recurrent neural network feature enhancement," in *Proc. REVERB*, 2014.

[10] K. Han, Y. Wang, D. L. Wang, W. S. Woods, I. Merks, and T. Zhang, "Learning spectral mapping for speech dereverberation and denoising," *IEEE/ACM Trans. Audio Speech and Lang. Process.*, vol. 23, pp. 982–992, 2015.

[11] B. Xia and C. Bao, "Speech enhancement with weighted denoising auto-encoder," in *Proc. INTERSPEECH*, 2013, pp. 3444–3448.

[12] X. Feng, Y. Zhang, and J. Glass, "Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition," in *Proc. ICASSP*, 2014, pp. 1778–1782.

[13] F. Weninger, S. Watanabe, Y. Tachioka, and B. Schuller, "Deep recurrent de-noising auto-encoder and blind dereverberation for reverberated speech recognition," in *Proc. ICASSP*, 2014, pp. 4623–4627.

[14] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Ensemble modeling of denoising autoencoder for speech spectrum restoration," in *Proc. INTERSPEECH*, 2014, pp. 885–889.

[15] K. H. Lee, S. J. Kang, W. H. Kang, and N. S. Kim, "Two-stage noise aware training using asymmetric deep denoising autoencoder," in *Proc. ICASSP*, 2016, pp. 5765–5769.

[16] P. G. Shivakumar and P. Georgiou, "Perception optimized deep denoising autoencoders for speech enhancement," in *Proc. INTERSPEECH*, 2016, pp. 3743–3747.

[17] K. Paliwal, K. Wójcicki, and B. Shannon, "The importance of phase in speech enhancement," *Speech Commun.*, vol. 53, pp. 465–494, 2010.

[18] P. Mowlaee, R. Saeidi, and R. Martin, "Phase estimation for signal reconstruction in single-channel speech separation.," in *Proc. INTERSPEECH*, 2012, pp. 1–4.

[19] T. Gerkmann, M. Krawczyk-Becker, and J. Le Roux, "Phase processing for single-channel speech enhancement: History and recent advances," *IEEE Signal Process. Mag.*, vol. 32, pp. 55–66, 2015.

[20] F. Mayer, D. S. Williamson, P. Mowlaee, and D. L. Wang, "Impact of phase estimation on single-channel speech separation based on time-frequency masking," *J. Acoust. Soc. Am.*, vol. 141, pp. 4668–4679, 2017.

[21] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. ICASSP*, 2015, pp. 708–712.

[22] D. S. Williamson, Y. Wang, and D. L. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Trans. Audio Speech and Lang. Process.*, vol. 24, pp. 483–492, 2016.

[23] D. S. Williamson and D. L. Wang, "Time-frequency masking in the complex domain for speech dereverberation and denoising," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, pp. 1492–1501, 2017.

[24] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT acoustic phonetic continuous speech corpus," *Available: http://www.ldc.upenn.edu/Catalog/LDC93S1.html*, 1993.

[25] ITU-R, "Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs," p. 862, 2001.

[26] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time frequency weighted noisy speech," *IEEE Trans. Audio Speech and Lang. Process.*, vol. 19, pp. 2125–2136, 2011.