EXPLORING DEEP COMPLEX NETWORKS FOR COMPLEX SPECTROGRAM ENHANCEMENT

Ashutosh Pandey[1] *and DeLiang Wang*[1,2]

[1]Department of Computer Science and Engineering, The Ohio State University, USA
[2]Center for Cognitive and Brain Sciences, The Ohio State University, USA
{pandey.99, wang.77}@osu.edu

ABSTRACT

A recent study has demonstrated the effectiveness of complex-valued deep neural networks (CDNNs) using newly developed tools such as complex batch normalization and complex residual blocks. Motivated by the fact that CDNNs are well suited for the processing of complex-domain representations, we explore CDNNs for speech enhancement. In particular, we train a CDNN that learns to map the complex-valued noisy short-time Fourier transform (STFT) to the clean STFT. Additionally, we propose the complex-valued extensions of the parametric rectified linear unit (PReLU) nonlinearity that helps to improve the performance of CDNN. Experimental results demonstrate that a CDNN using the proposed nonlinearity can give similar or better enhancement results compared to real-valued deep neural networks (DNNs).

Index Terms— complex-valued deep neural networks, CDNN, phase-aware speech enhancement, learning phase

1. INTRODUCTION

Speech enhancement is concerned with improving the intelligibility and quality of a speech signal degraded by additive noise. It has many real-world applications such as robust automatic speech recognition, mobile speech communication, and hearing aids design. Traditional speech enhancement approaches include statistical enhancement methods [1] and computational auditory scene analysis [2].

In recent years, supervised methods for speech enhancement using DNNs have become the mainstream [3]. Some of the most popular deep learning based methods are: feed-forward DNNs [4, 5], deep denoising autoencoders [6], and convolutional neural networks (CNNs) [7].

Typical speech enhancement systems operate in the time-frequency (T-F) domain by enhancing the magnitude response and leaving the phase response unaltered. Some studies in the past have shown that phase is important for the perceptual quality of speech [8]. This has led researchers to develop several phase enhancement algorithms [9, 10, 11]. In [10], the authors propose the complex ideal ratio mask (cIRM) as the training target that, when multiplied with the noisy STFT, gives the clean STFT; they then use a standard DNN to jointly estimate the real and the imaginary part of the cIRM. In [11], the authors employ a CNN to predict the real and the imaginary part of STFT as separate channels in the output layer of the CNN. Since STFT is complex-valued, using a CDNN is a promising alternative for phase-aware enhancement. In [12], the authors employ CDNN for beamforming and find that CDNNs do not give a considerable improvement over the DNNs. Recently, in [13], the authors explore CDNNs for singing source separation, and train it with an additional sparsity constraint to obtain some improvement.

Recently, Trabelsi et al. [14] propose elementary building blocks for CDNNs such as complex batch normalization, complex weight initialization, and complex residual blocks. Additionally, they show that CDNNs can outperform DNNs for speech spectrum prediction.

Motivated by [14], we explore CDNNs for monaural speech enhancement. To our knowledge, CDNNs have not been investigated for this task in the past. Our study is different from the past studies in following aspects. First, we employ CDNN that is trained using real-valued backpropagation whereas the previous studies have explored complex backpropagation. Second, we utilize complex batch normalization inside our model that has not been suggested previously. Finally, we propose complex-valued extensions of PReLU to improve the performance of the CDNN further.

This paper is organized as follows. The next section describes different building blocks inside the employed fully connected CDNN. Section 3 gives the details about the performed experiments. Section 4 concludes the paper.

2. FULLY CONNECTED COMPLEX DEEP NEURAL NETWORKS

In a CDNN, complex vectors are represented using real vectors. The real and the imaginary part of a complex vector are concatenated to form a real vector. Given a complex-valued vector $\mathbf{h} = \mathbf{x} + i\mathbf{y}$, it will be represented inisde the CDNN as:

$$\mathbf{h} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \tag{1}$$

In a CDNN, the length of a vector is always even as it formed by the concatenation of two equal-sized vectors.

2.1. Complex Matrix Multiplication

Given a complex matrix $\mathbf{W} = \mathbf{A} + i\mathbf{B}$ and a complex vector $\mathbf{h} = \mathbf{x} + i\mathbf{y}$, a complex matrix multiplication is given by:

$$\mathbf{Wh} = (\mathbf{Ax} - \mathbf{By}) + i(\mathbf{Bx} + \mathbf{Ay}) \tag{2}$$

The above equation can be represented using a real matrix and a real vector as:

$$\begin{bmatrix} \Re(\mathbf{Wh}) \\ \Im(\mathbf{Wh}) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{-B} \\ \mathbf{B} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \tag{3}$$

Equation 3 shows how a fully connected layer is defined inside a CDNN. Each matrix is formed by stacking two real matrices, the real and the imaginary part of the complex valued matrix, according to Equation 3. Fig. 1 illustrates a 3-layered fully connected CDNN. Each matrix multiplication in the CDNN is followed by complex batch normalization [14] and $\mathbb{C}$PReLU nonlinearity described in Section 2.2.4.

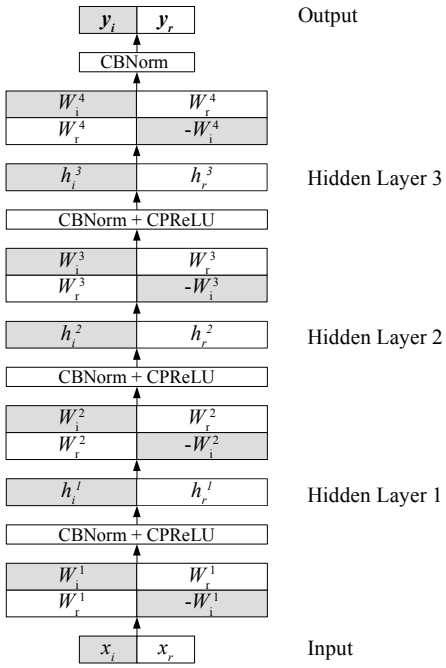

Fig. 1. A diagram illustrating the implementation of a fully connected CDNN.

2.2. Complex-Valued Activations

In this study, we have investigated the following complex-valued nonlinearities:

2.2.1. MODRELU

First proposed in [15], the modReLU is defined as:

$$\text{modReLU}(z) = \text{ReLU}(|z| + b) \tag{4}$$

where b is a trainable parameter. This nonlinearity only modifies the magnitude by creating a dead zone around the origin. The radius of the dead zone is learned at the training time.

2.2.2. zRELU

zReLU was proposed in [16]. It is defined as:

$$z\text{ReLU}(z) = \begin{cases} z, & \text{if } \theta_z \in \left[0, \pi/2\right] \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where θ_z is the phase of complex number z. This nonlinearity only allows to pass the signals that lie in the first quadrant of the complex plane.

2.2.3. $\mathbb{C}$RELU

$$\mathbb{C}\text{ReLU}(z) = \text{ReLU}(\Re(z)) + i\text{ReLU}(\Im(z)) \tag{6}$$

This nonlinearity applies the rectified linear unit (ReLU) nonlinearity separately on the real and the imaginary part of the complex number z.

2.2.4. $\mathbb{C}$PRELU

$$\begin{aligned} \Re(\mathbb{C}\text{PReLU}(z)) &= \begin{cases} \Re(z), & \text{if } \Re(z) \geq 0 \\ \Re(z) * \alpha_R, & \text{otherwise} \end{cases} \\ \Im(\mathbb{C}\text{PReLU}(z)) &= \begin{cases} \Im(z), & \text{if } \Im(z) \geq 0 \\ \Im(z) * \alpha_I, & \text{otherwise} \end{cases} \end{aligned} \tag{7}$$

This nonlinearity is the extension of PReLU [17] to complex domain. It applies PReLU separately to the real and the imaginary part of the complex number. α_r and α_i in Equation 7 are real-valued trainable parameters.

2.2.5. zPRELU

$$z\text{PReLU}(z) = \begin{cases} z, & \text{if } \theta_z \in \left[0, \pi/2\right] \\ z * \alpha, & \text{otherwise} \end{cases} \tag{8}$$

This nonlinearity is similar to zReLU. It has one complex-valued trainable parameter α. The input, z, is multiplied with α if it does not lie in the first quadrant. Note that the multiplication here is a complex multiplication.

Table 1. Complex-valued nonlinearity comparisons for noise-dependent models.

noises	babble		factory		SSN		oproom		engine		average	
evaluation metrics	STOI(%)	PESQ	STOI(%)	PESQ	STOI(%)	PESQ	STOI(%)	PESQ	STOI(%)	PESQ	STOI(%)	PESQ
unprocessed	66.4	1.75	65.4	1.63	68.9	1.76	69.8	1.82	68.8	1.49	67.8	1.69
$\mathbb{C}$ReLU	77.1	1.96	77.2	1.96	80.8	2.04	83.2	2.47	84.6	2.36	80.6	2.16
zReLU	72.0	1.77	72.7	1.83	75.0	1.82	77.5	2.08	78.6	2.13	75.2	1.93
modReLU	76.2	2.07	77.4	2.14	80.2	2.20	82.3	2.44	83.6	2.39	79.9	2.25
zPReLU	74.6	2.03	75.5	2.16	78.0	2.07	79.9	2.27	81.1	2.26	77.8	2.16
$\mathbb{C}$PReLU	**77.1**	**2.08**	**77.8**	**2.19**	**80.8**	**2.20**	**83.8**	**2.53**	**84.5**	**2.50**	**80.8**	**2.30**
z3PReLU	76.7	**2.08**	77.5	**2.19**	80.4	2.17	82.8	2.44	83.9	2.42	80.3	2.26

Table 2. CDNN and DNN comparisons for noise-dependent models.

noises	babble		factory		SSN		oproom		engine		average	
evaluation metrics	STOI (%)	PESQ	STOI (%)	PESQ	STOI (%)	PESQ	STOI (%)	PESQ	STOI (%)	PESQ	STOI (%)	PESQ
unprocessed	66.4	1.75	65.4	1.63	68.9	1.76	69.8	1.82	68.8	1.49	67.8	1.69
DNN-SM	75.6	2.01	76.3	2.04	79.9	2.09	83.7	2.44	83.9	2.43	79.9	2.20
DNN-RI	76.6	**2.11**	77.5	**2.23**	80.5	**2.21**	83.1	2.51	84.0	2.47	80.3	**2.31**
CDNN-RI	**77.1**	2.08	**77.8**	2.19	**80.8**	2.20	**83.8**	**2.53**	**84.5**	**2.50**	**80.8**	2.30

2.2.6. z3PReLU

$$z3\text{PReLU}(z) = \begin{cases} z, & \text{if } \theta_z \in [0,\ \pi/2] \\ z * \alpha_1, & \text{if } \theta_z \in [\pi/2,\ \pi) \\ z * \alpha_2, & \text{if } \theta_z \in [\pi,\ 3\pi/2) \\ z * \alpha_3, & \text{otherwise} \end{cases} \tag{9}$$

This nonlinearity has three complex-valued trainable parameters. The input, z, is multiplied with different complex numbers depending on its quadrant.

2.3. Complex Batch Normalization

Complex batch normalization was introduced in [14]. In a complex batch normalization, data is first centered using the mean from the batch. The centered data is multiplied with the square root inverse of 2×2 covariance matrix calculated using the batch. These two steps can be written in an equation as:

$$\tilde{\boldsymbol{x}} = (\boldsymbol{V})^{-\frac{1}{2}}(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}]) \tag{10}$$

where,

$$\begin{aligned} \boldsymbol{V} &= \begin{pmatrix} \boldsymbol{V}_{rr} & \boldsymbol{V}_{ri} \\ \boldsymbol{V}_{ir} & \boldsymbol{V}_{ii} \end{pmatrix} \\ &= \begin{pmatrix} \text{Cov}(\Re\{\boldsymbol{x}\}, \Re\{\boldsymbol{x}\}) & \text{Cov}(\Re\{\boldsymbol{x}\}, \Im\{\boldsymbol{x}\})) \\ \text{Cov}(\Im\{\boldsymbol{x}\}, \Re\{\boldsymbol{x}\}) & \text{Cov}(\Im\{\boldsymbol{x}\}, \Im\{\boldsymbol{x}\})) \end{pmatrix} \end{aligned}$$

where $\text{Cov}(\mathbf{x}, \mathbf{y})$ denotes the covariance between **x** and **y**.

The multiplication with $(V)^{(-\frac{1}{2})}$ is done to make sure that the real and the imaginary parts are decorrelated and have unit variance. The centered and whitened data is multiplied with a 2×2 matrix with three trainable parameters. This matrix multiplication is equivalent of learnable scaling in the real-valued batch normalization. The trainable matrix is given by:

$$\boldsymbol{\gamma} = \begin{pmatrix} \gamma_{rr} & \gamma_{ri} \\ \gamma_{ir} & \gamma_{ii} \end{pmatrix} \tag{11}$$

Finally, the output is shifted by a learnable bias $\boldsymbol{\beta}$ as given in the following equation:

$$\text{BN}(\boldsymbol{x}) = \boldsymbol{\gamma}\tilde{\boldsymbol{x}} + \boldsymbol{\beta} \tag{12}$$

2.4. Complex Weight Initialization

The weight matrices are initialized in such a way that the columns of the matrix are as independent as possible and satisfy the criterion given in [18]:

$$\text{Var}(W) = 2/(n_{in} + n_{out}) \tag{13}$$

where W is the complex weight matrix, n_{in} is the number of input nodes and n_{out} is the number of output nodes. This is achieved as follows. First, two real-valued matrices initialized uniformly between 0 and 1 are used as real and imaginary part to form a complex matrix. Then a singular value decomposition (SVD) is applied to the complex matrix. The diagonal matrix from the SVD is replaced with an identity matrix to construct a new complex matrix. The real and the imaginary part of the obtained complex matrix are scaled separately to have a variance as given in Equation 13.

3. EXPERIMENTS

3.1. Datasets

We compare the models in noise-dependent and noise-independent way on the TIMIT dataset [19]. All 4620 utterances in the training set are used to create the training mixtures. Five noise-dependent models are trained on noises: babble, factory, speech-shaped noise (SSN), oproom and engine. The noisy utterances are generated in the following way. For each utterance, five SNR values are uniformly and randomly selected between the range [-5, 5] and a random segment from the first half of the noise is mixed at the selected SNR. 192 utterances in the TIMIT core test set are

used for the test set. The test utterances are generated at the SNRs -6 dB, -3 dB, 0 dB, 3 dB and 6 dB using segments from the second half of the noises. For noise-independent models, the training utterances are created using the five noises mentioned above and tested on two untrained noises: factory2 and tank. All the used noises are from the Noisex dataset [20].

3.2. Baselines

For the baseline models, we train two three-layered DNN models: DNN-SM, a model that predicts the clean STFT magnitude using the noisy STFT magnitude; DNN-RI, a model that jointly predicts the real and the imaginary part of the STFT. The DNNs use 1024 units with PReLU at the hidden layers. The CDNNs use 724 complex hidden units to keep the number of parameters same. The DNN-SM uses softplus nonlinearity at the output. DNN-RI and CDNN use no activation at the output. All the matrix multiplications are followed by batch normalization in DNNs and complex batch normalization in CDNNs.

3.3. Experimental settings

All the utterances are resampled to 16 kHz. The frames are extracted using the Hamming window with a frame size of 20 ms and frame shift of 10 ms. The real and the imaginary parts of the STFT are whitened, similar to complex batch normalization, using the statistics from the training set.

All the models are trained using a batch size of 4096. The Adam optimizer [21] is used for stochastic gradient descent (SGD) based optimization. The learning rate is set to 0.0002. The DNNs are initialized using Xavier initializer[18] whereas the CDNNs are initialized using the method described in Section 2.4. A dropout of 0.2 is applied after each hidden layer. In CDNNs, same dropout mask is applied to the real and the imaginary part of the tensor. All the networks predict one frame at a time using corresponding features from the noisy frame at the input.

3.4. Experimental results

We compare all the models using short-term objective intelligibility (STOI) [22] and perceptual evaluation of speech quality (PESQ) [23] scores. First, we compare different complex-valued nonlinearities described in section 2.2. The results for noise-dependent and noise-independent models are given in Table 1 and Table 3. In summary, zReLU and zPReLU perform the worst. The proposed $\mathbb{C}$PReLU gives the best overall results followed by z3PReLU. The z3PReLU nonlinearity gives slightly worse performance compared with the $\mathbb{C}$PReLU. A further investigation of z3PReLU is needed for large training-set and deeper models as it uses more parameters and has the potential to learn better representations. A similar performance trend is observed for noise-dependent and noise-independent models.

We also compare the CDNN that uses $\mathbb{C}$PReLU with the baseline models DNN-SM and DNN-RI. The results for noise-dependent and noise-independent models are given in Table 2 and Table 4. We observe that the phase learning models, DNN-RI and CDNN, outperform the magnitude based model, DNN-SM, in all the cases. The CDNN gives slightly better STOI scores in all the scenarios whereas DNN is similar or marginally better regarding PESQ. The performance trend is similar for noise-dependent and noise-independent models.

Table 3. Complex-valued nonlinearity comparisons for noise-independent models.

noises	factory2		tank		average	
evaluation metrics	STOI(%)	PESQ	STOI(%)	PESQ	STOI(%)	PESQ
unprocessed	74.9	1.93	76.8	2.06	75.8	1.99
$\mathbb{C}$ReLU	82.9	2.29	83.8	2.34	83.3	2.31
zReLU	75.7	1.94	76.7	2.02	76.2	1.98
modReLU	81.8	2.35	82.9	2.40	82.3	2.37
zPReLU	80.9	2.33	81.7	2.44	81.3	2.39
$\mathbb{C}$PReLU	**83.3**	**2.47**	**84.3**	**2.51**	**83.8**	**2.49**
z3PReLU	83.1	2.45	**84.3**	**2.51**	83.7	2.48

Table 4. CDNN and DNN comparisons for noise-independent models.

noises	factory2		tank		average	
evaluation metrics	STOI (%)	PESQ	STOI (%)	PESQ	STOI (%)	PESQ
unprocessed	74.9	1.93	76.8	2.06	75.8	1.99
DNN-SM	81.9	2.14	82.8	2.17	82.3	2.16
DNN-RI	83.0	**2.48**	83.9	**2.5**1	83.5	**2.49**
CDNN	**83.3**	2.47	**84.3**	**2.51**	**83.8**	**2.49**

4. CONCLUSIONS

In this study, we have explored complex-valued deep neural networks for complex spectrogram enhancement. Recently developed complex batch normalization and complex weight initialization are utilized for effective network training. Although CDNNs perform comparably to the corresponding real-valued networks, due to their inherent affinity to spectral enhancement, they have the potential to outperform their real-valued counterparts after effective operations developed in regular DNNs are successfully transferred to the complex domain. Additionally, we have proposed and explored different complex nonlinearities and found that $\mathbb{C}$PReLU that applies PReLU separately on the real and the imaginary part outperforms the other complex-valued nonlinearities. Future work includes exploring the proposed nonlinearities for different architectures and tasks.

5. ACKNOWLEDGEMENTS

This research was supported in part by two NIDCD (R01 DC012048 and R01 DC015521) grants and the Ohio Supercomputer Center.

6. REFERENCES

[1] P. C. Loizou, *Speech Enhancement: Theory and Practice*, CRC Press, Boca Raton, FL, USA, 2nd edition, 2013.

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

[3] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, pp. 1702–1726, 2018.

[4] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1381–1390, 2013.

[5] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 23, no. 1, pp. 7–19, 2015.

[6] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder.," in *Proceedings of Interspeech*, 2013, pp. 436–440.

[7] A. Pandey and D. Wang, "A new framework for supervised speech enhancement in the time domain," in *Proceedings of Interspeech*, 2018, pp. 1136–1140.

[8] K. Paliwal, K. Wojcicki, and B. Shannon, "The importance of phase in speech enhancement," *Speech Communication*, vol. 53, no. 4, pp. 465 – 494, 2011.

[9] M. Krawczyk and T. Gerkmann, "STFT phase reconstruction in voiced speech for an improved single-channel speech enhancement," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 12, pp. 1931–1940, 2014.

[10] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 24, no. 3, pp. 483–492, 2016.

[11] S.-W. Fu, T.-y. Hu, Y. Tsao, and X. Lu, "Complex spectrogram enhancement by convolutional neural network with multi-metrics learning," in *27th International Workshop on Machine Learning for Signal Processing*. IEEE, 2017, pp. 1–6.

[12] L. Drude, B. Raj, and R. Haeb-Umbach, "On the appropriateness of complex-valued neural networks for speech enhancement.," in *Proceedings of Interspeech*, 2016, pp. 1745–1749.

[13] Y.-S. Lee, C.-Y. Wang, S.-F. Wang, J.-C. Wang, and C.-H. Wu, "Fully complex deep neural network for phase-incorporating monaural source separation," in *Proceedings of ICASSP*, 2017, pp. 281–285.

[14] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," *arXiv preprint arXiv:1705.09792*, 2017.

[15] M. Arjovsky, A. Shah, and Y. Bengio, "Unitary evolution recurrent neural networks," in *International Conference on Machine Learning*, 2016, pp. 1120–1128.

[16] N. Guberman, "On complex valued convolutional neural networks," *arXiv preprint arXiv:1602.09046*, 2016.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[18] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[19] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. nist speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, 1993.

[20] A. Varga and H. J. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, no. 3, pp. 247–251, 1993.

[21] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[23] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs," in *Proceedings of ICASSP*, 2001, pp. 749–752.