Applying Complex-Valued Neural Networks to Acoustic Modeling for Speech Recognition

Daichi Hayakawa*, Takashi Masuko* and Hiroshi Fujimura*
* Corporate Research & Development Center, Toshiba Corporation, Kawasaki, Japan
E-mail: {daichi1.hayakawa, takashi.masuko, hiroshi4.fujimura}@toshiba.co.jp Tel: +81-44-549-2395

***Abstract*—Complex-valued neural networks (CVNNs) are well suited to speech signal processing because they can naturally represent amplitude and phase. In this paper, we explore applying an acoustic model with multiple complex-valued layers (multiple-CVNN-AM) and spliced features to speech recognition. First, we focus on multiple-CVNN-AM with unspliced input features and investigate an appropriate architecture from the viewpoint of the activation function, bias, and number of complex-valued layers. We also propose batch amplitude mean normalization for more quickly and stably training complex-valued layers. We then investigate an appropriate architecture for multiple-CVNN-AM with spliced input features and compare it with a real-valued neural network acoustic model without complex-valued layers (RVNN-AM) and complex linear projection (CLP) models, which can be considered acoustic models with single complex-valued layers. We show that under noise conditions, multiple-CVNN-AM outperforms RVNN-AM and CLP models by up to 7.45% and 11.90%, respectively.**

I. Introduction

Complex-valued neural networks (CVNNs) are neural networks that deal with complex-valued features [1]. CVNNs have better generalization characteristics [2] and potential to enable faster learning [3-5] . Furthermore, CVNNs are well suited to speech signal processing where complex values are often used through fast Fourier transform (FFT), because CVNNs can naturally represent amplitude and phase. CVNNs have previously been shown to be effective in spectrum prediction [5], source localization [6], and automatic music transcription [7]. In this paper, we focus on applying CVNNs to acoustic modeling for speech recognition.

Recently, Variani et al. proposed complex linear projection (CLP) [8]. CLP processes FFT features by inserting a complex-valued linear layer (called a CLP layer) between a complex-valued input and a real-valued neural network acoustic model (RVNN-AM). FFT features are unspliced; that is, CLP processes one frame of FFT features. A CLP layer plays the role of filtering the input signal. The model obtained via joint training of the CLP layer and the RVNN-AM achieves superior performance as compared with the RVNN-AM without the CLP layer, whose input is log Mel-filterbank energy features.

The CLP layer can be considered as a single complex-valued layer. As mentioned above, acoustic models with a single complex-valued layer (single-CVNN-AM) and unspliced input features have been explored. However, to the best of our knowledge, no acoustic models with multiple complex-valued layers (multiple-CVNN-AM) and spliced input features have been explored.

In this paper, we explore applying multiple-CVNN-AM with spliced features to speech recognition and an architecture for complex-valued layers that is appropriate to speech recognition. We propose batch amplitude mean normalization (BAMN) to train complex-valued layers more quickly and stably. We show that the multiple-CVNN-AM approach outperforms RVNN-AM and single-CVNN-AM under noise conditions.

The remainder of this paper is organized as follows. In section II, we describe a typical CVNN architecture for acoustic models and BAMN. In section III, we focus on multiple-CVNN-AM with unspliced input features. We investigate activation functions appropriate to speech recognition, the necessity of bias, and how many complex-valued layers are needed. We also evaluate the effectiveness of BAMN. In section IV, we compare multiple-CVNN-AM with RVNN-AM and single-CVNN-AM under the condition that input features are spliced. We show that multiple-CVNN-AM outperforms the RVNN-AM and single-CVNN-AM models. We also investigate appropriate architectures for multiple-CVNN-AM with spliced input features. Section V concludes this paper.

II. Complex-Valued Neural Network for Acoustic Model

A. Typical CVNN Architecture for Acoustic Model

Fig. 1 shows a typical structure for a multiple-CVNN-AM architecture in an acoustic model. There are three layers in

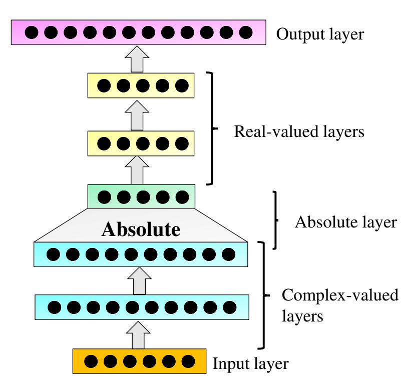


Fig. 1. Typical multiple CVNN architecture for an acoustic model.

a multiple CVNN: complex-valued, absolute, and real-valued layers.

The complex- and real-valued layers process and produce complex- and real-valued inputs, respectively.

An acoustic model using deep neural networks produces posterior probabilities over hidden Markov model states. Consequently, output from multiple-CVNN-AM must be real-valued. In this paper, we introduce an absolute layer between the complex- and real-valued layers. The absolute layer outputs absolute values resulting from complex-valued inputs and propagates these output values to the next real-valued layer.

Complex- and real-valued layers are optimized through complex-valued back propagation [9] and real-valued back propagation [10], respectively.

B. Activation Functions for CVNN

Several complex activation functions have been proposed in the literature [11-15] . These active functions are mainly classified into the following three types.

- Real–imaginary function

$$z^{out} = \sigma_R(\Re[z^{in}]) + i\sigma_R(\Im[z^{in}]). \tag{1}$$

- Phase-amplitude function

$$z^{out} = \sigma_R(|z^{in}|)\exp(i\arg(z^{in})). \tag{2}$$

- Complex function

$$z^{out} = \sigma_C(z^{in}). \tag{3}$$

Here, $z^{in} \in \mathbb{C}$ and $z^{out} \in \mathbb{C}$ denote an activation function's input and output, respectively, $\sigma_R(\cdot)$ denotes a real-valued function such as a sigmoid or a complex function, and $\sigma_C(\cdot)$ denotes, for example, a complex sigmoid [11]. In this paper, we focus on real–imaginary and phase–amplitude functions.

C. Batch Amplitude Mean Normalization

Batch normalization [16] helps accelerate and stabilize a training neural network by reducing internal covariate shift.

Conventional batch normalization involves the following steps. Let a mini-batch $\mathcal{B}$ of size m be $\mathcal{B} = \{x_{1...m}\} \in \mathbb{R}$, and let the normalized values be $\hat{x}_{1...m} \in \mathbb{R}$. Then,

$$\mu_{\mathcal{B}} = \frac{1}{m}\sum_{n=1}^{m} x_n,$$
$$\sigma^2_{\mathcal{B}} = \frac{1}{m}\sum_{n=1}^{m}(x_n - \mu_{\mathcal{B}})^2,$$
$$\hat{x}_n = \frac{x_n - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}},$$

where $\epsilon \in \mathbb{R}$ is a constant added to the mini-batch variance for numerical stability and $n = 1, 2, ..., m$. After normalizing the mini-batch, the normalized values are scaled by $\gamma \in \mathbb{R}$ and shifted by $\beta \in \mathbb{R}$. The output of batch normalization x_n^{BN} is therefore

$$x_n^{BN} = \gamma\hat{x}_n + \beta.$$

However, the standard formulation of batch normalization applies only to real values.

Trabelsi et al. proposed complex batch normalization [7], which takes a whitening two-dimensional vectors approach. Complex batch normalization can be performed by multiplying zero-centered data by the inverse square root of the covariance matrix.

Let a complex-valued mini-batch $\mathcal{B}$ of size m be $\mathcal{B} = \left\{ \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} ... \begin{bmatrix} x_m \\ y_m \end{bmatrix} \right\} \in \mathbb{R}^{2\times 1}$. Here, x_n and y_n $(n = 1, ...m)$ denote the real and imaginary parts of the nth components, respectively, and $\begin{bmatrix} \hat{x}_1 \\ \hat{y}_1 \end{bmatrix} ... \begin{bmatrix} \hat{x}_m \\ \hat{y}_m \end{bmatrix}$ denotes the normalized values.

$$\mu_{\mathcal{B}} = \frac{1}{m}\sum_{n=1}^{m} \begin{bmatrix} x_n \\ y_n \end{bmatrix},$$
$$\mathbf{V}_{\mathcal{B}} = \begin{pmatrix} \mathrm{Cov}(x_{1...m}, x_{1...m}) & \mathrm{Cov}(x_{1...m}, y_{1...m}) \\ \mathrm{Cov}(x_{1...m}, y_{1...m}) & \mathrm{Cov}(y_{1...m}, y_{1...m}) \end{pmatrix},$$
$$\begin{bmatrix} \hat{x}_n \\ \hat{y}_n \end{bmatrix} = (\mathbf{V}_{\mathcal{B}})^{-\frac{1}{2}} \left(\begin{bmatrix} x_n \\ y_n \end{bmatrix} - \mu_{\mathcal{B}} \right).$$

After normalizing the mini-batch, the normalized values are scaled by $\boldsymbol{\gamma} = \begin{pmatrix} \gamma_{rr} & \gamma_{ri} \\ \gamma_{ri} & \gamma_{ii} \end{pmatrix} \in \mathbb{R}^{2\times 2}$ and shifted by $\boldsymbol{\beta} \in \mathbb{R}^{2\times 1}$. The output of complex batch normalization $\begin{bmatrix} x_n^{CBN} \\ y_n^{CBN} \end{bmatrix}$ is

$$\begin{bmatrix} x_n^{CBN} \\ y_n^{CBN} \end{bmatrix} = \boldsymbol{\gamma} \begin{bmatrix} \hat{x}_n \\ \hat{y}_n \end{bmatrix} + \boldsymbol{\beta}.$$

However, this approach contaminates any phase information the data have, potentially degrading speech recognition performance. Therefore, we propose BAMN.

BAMN is performed by dividing the mini-batch by its mean absolute value. Let a mini-batch $\mathcal{B}$ of size m be $\mathcal{B} = \{z_{1...m}\} \in \mathbb{C}$ and let its normalized values be $\hat{z}_{1...m} \in \mathbb{C}$. Then,

$$\mu_{\mathcal{B}} = \frac{1}{m}\sum_{n=1}^{m} |z_n|, \tag{4}$$

$$\hat{z}_n = \frac{z_n}{\mu_{\mathcal{B}} + \epsilon}, \tag{5}$$

where $\epsilon \in \mathbb{R}$ is a constant added to the mini-batch mean absolute value $\mu_{\mathcal{B}}$ for numerical stability and $n = 1, 2, ..., m$. After normalizing the mini-batch, we use scaling parameter $\gamma \in \mathbb{R}$. The output of BAMN z_n^{BAMN} is therefore

$$z_n^{BAMN} = \gamma\hat{z}_n. \tag{6}$$

The parameter γ is learned to achieve a desired mean. However, phase information of the mini-batch is inverted if γ is less than 0. To avoid this problem, γ with values less than 0 are clipped at 0. Consequently, BAMN can reduce internal covariant shift while maintaining phase information to help accelerate and stabilize the neural network training.

III. Experiments on Unspliced Input Features

In this section, we focus on multiple-CVNN-AM with unspliced input features. We investigate an activation function appropriate for speech recognition, the necessity of bias, and how many complex-valued layers are needed. We also evaluate the effectiveness of BAMN.

A. Experimental Setup

We conducted experiments using Wall Street Journal (WSJ) [17] datasets. A multiple-CVNN-AM was trained on the SI-284 set, which contains 82 hours of speech data. Evaluation was performed using the November 1992 ARPA WSJ test set, which contains 333 sentences. For training and evaluation, we used a modified version of the Kaldi [18] speech recognition toolkit. FFT was applied to raw signals sampled at 16 kHz after pre-emphasis with parameter 0.97. The FFT was computed with 512 frequency bins using Hanning windows of 25 ms (400 samples) with a 10 ms shift (160 samples) and zero padding. We used 257-dimensional FFT features, which correspond to 0 Hz–8 kHz. The amplitude of the FFT features was normalized to a unit mean.

Fig. 2 shows the architecture of a multiple-CVNN-AM with unspliced input features. Each complex-valued layer has 1024 units (512 units each for the real and imaginary parts), and each real-valued layer has 512 units. The output layer has 3367 units. Both complex- and real-valued layers are fully connected. The total number of complex- and real-valued layers is set to 4. The activation function of real-valued layers is fixed to a sigmoid function. The multiple-CVNN-AM was trained using backpropagation with stochastic gradient descent in a similar manner to section 3.2 in [19]. The initial learning rate was set to 0.0008. Optimal language model weights for decoding were selected for each model.

B. Activation Function Type

In this section, we compare real–imaginary and phase–amplitude functions for use as activation functions. We conducted this experiment by changing the number of complex-valued layers from two to four layers. We use $\sigma_R(x) = \tanh(x)$ for this comparison.

Table I shows word error rates (WERs) for each activation type and number of layers. This table shows that a phase–amplitude function achieves superior performance as compared with a real–imaginary function. A phase–amplitude function can propagate phase information to the next layer without change, so complex-valued layers are able to effectively handle phase information from input features. These results show that phase information is important for training Multiple-CVNN-AM.

TABLE I
WERs for each activation type and number of layers

Activation	Activation type	# of Complex-valued layers: 2 layers	3 layers	4 layers
tanh	Real-imaginary	9.14	11.11	13.24
	Phase-magnitude	**8.28**	**10.08**	**12.26**

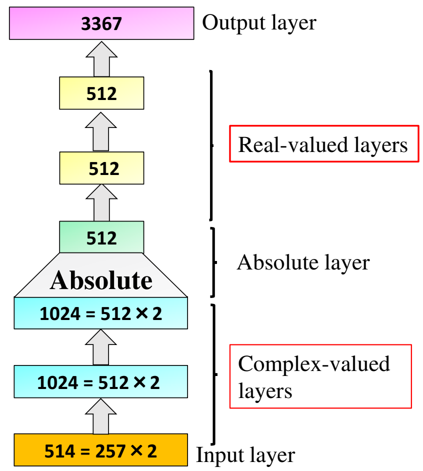


Fig. 2. Architecture of multiple-CVNN-AM with unspliced input features.

C. With or Without Bias

In this section, we investigate the necessity of bias. The multiple-CVNN-AM structure has two complex-valued and two real-valued layers.

We chose the following phase–amplitude functions

- tanh : $\sigma_R(|z|) = \tanh(|z|)$
- squash : $\sigma_R(|z|) = \dfrac{|z|^2}{1+|z|^2}$
- log : $\sigma_R(|z|) = \log(|z|+1)$

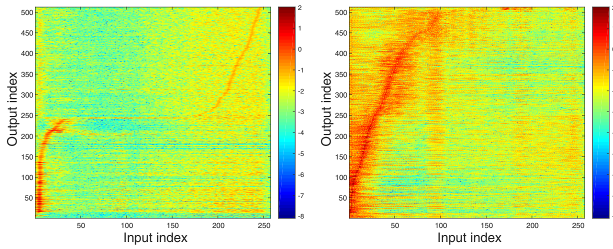
Table II shows WERs for each activation functions with and without bias. This table shows that Multiple-CVNN-AM without bias achieves superior performance as compared with that with bias.

As described in [8], multiplying the complex-valued layer's inputs by a complex-valued weighting matrix is equivalent to convoluting the input signal with a filter followed by weighted average pooling. Therefore, it is possible that training a complex-valued weight matrix with bias will not work because the bias harms phase information.

Fig. 3 shows the logarithm of the magnitude weights of the first complex-valued layer in multiple-CVNN-AM with (Fig. 3, left) and without (Fig. 3, right) bias. The phase–amplitude function is tanh. For the first complex-valued layer without bias, the set of narrowband bandpass filters can be clearly observed. On the other hand, for the first complex-valued layer, more than half of the row vectors do not have clear narrowband bandpass filters. In other words, training the

TABLE II
WERs for each activation function with and without bias

Activation	w/ bias	w/o bias
tanh	11.13	**8.28**
squash	9.43	**8.70**
log	10.35	**8.17**



(a) With the bias (b) Without the bias

Fig. 3. Logarithm of magnitude weights in the first complex-valued layer in multiple-CVNN-AM (a) with bias and (b) without bias. The phase–amplitude function is tanh.

TABLE III
WERS FOR EACH NUMBER OF COMPLEX-VALUED LAYERS

Activation	1 layer	2 layers	3 layers	4 layers
tanh	8.56	**8.28**	10.08	12.26
squash	9.18	**8.70**	10.49	10.47
log	8.44	**8.17**	9.18	Failed

complex-valued layer is difficult when each complex-valued layer has bias.

D. Number of Complex-Valued Layers

In this section, we investigate the number of required complex-valued layers. We choose the three phase–amplitude functions presented in section III-C.

Table III shows WERs for each number of complex-valued layers. As that table shows, increasing the number of complex-valued layers to two improves speech recognition performance, but then performance starts to get worse. These results show that complex-valued layers are effective for speech recognition performance, and that having two complex-valued layers is optimal when the total number of complex and real-valued layers is four. This suggests that multiple complex- and real-valued layers are both required.

E. Effect of Batch Amplitude Mean Normalization

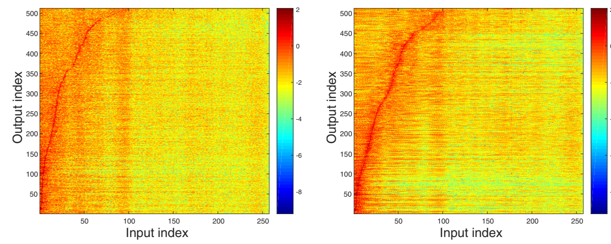
In this section, we evaluate the effect of BAMN. We choose two phase–amplitude functions, tanh and log, as described in section III-C.

Table IV shows WERs for each combination of BAMN and activation function. "BAMN→Activ" and "Activ→BAMN" designations in that table denote whether BAMN is applied before or after the activation function. The table shows that BAMN has little effect when the activation function is tanh, because the output of tanh is bounded above by 1, and hence the influence from the internal covariant shift is small. On the other hand, BAMN is effective when the activation function is log, because the log function output is unbounded. It is therefore possible that there is influence from the internal mean absolute value shift. The experimental results suggest that BAMN can reduce the influence of covariant shift.

Fig. 4 shows a logarithm of magnitude weights for the first complex-valued layer in the multiple-CVNN-AM without

TABLE IV
WERS FOR EACH COMBINATION OF BAMN AND ACTIVATION FUNCTION

Activation	no BAMN	BAMN→Activ	Activ→BAMN
tanh	8.28	**8.24**	8.26
log	8.17	**7.80**	8.01



(a) Without BAMN (b) With BAMN

Fig. 4. Logarithm of magnitude weights in the first complex-valued layer in multiple-CVNN-AM with and without bias. The phase–amplitude function is log.

(Fig. 4, left) and with (Fig. 4, right) BAMN. The phase–amplitude function is log. Without BAMN, the first one-third of low vectors have similar filters, with center frequency at 0, and some rows do not have narrowband bandpass filters. This implies that information related to input features is not effectively utilized. In contrast, narrowband bandpass filters are clearly observed with BAMN. These results suggest that BAMN improves the effectiveness of spectral modeling with a phase–amplitude log activation function.

IV. EXPERIMENTS ON SPLICED INPUT FEATURES

In this section, we compare multiple-CVNN-AM with RVNN-AM and single-CVNN-AM under the condition that input features are spliced. We choose an acoustic model with CLP (CLP-AM) [8] for the single-CVNN-AM. We also investigate an appropriate architecture for multiple-CVNN-AM with spliced input features.

A. Experimental Setup

We conducted experiments on noisy WSJ sets. The training data was the SI-284 set mixed with pedestrian area (PED) noise used in the third CHiME Speech Separation and Recognition Challenge (CHiME-3) [20]. Training data were divided into three parts, mixed with PED noise at 5, 10, and 15 dB. The training method was that described in section III-A. Evaluation was performed using the November 1992 ARPA WSJ test set mixed with PED noise. The test set was mixed with noise at signal-to-noise ratios (SNRs) of 0, 5, 10, 15, and 20 dB, resulting in 1665 mixtures (333 utterances $\times$ 5 SNRs).

The RVNN-AM has four real-valued layers and an output layer. Each real-valued layer has 512 nodes and the output layer has 3367 nodes. The activation function for real-valued layers is a sigmoid function. The input features of the RVNN-AM are 40-dimensional log Mel-filterbank energy features extracted every 10 ms on 25 ms windows from speech signals. Log Mel-filterbank energy features were normalized to zero mean and unit variance, and time-spliced with a context size of 11 frames.

Extracting a log Mel-filterbank is performed by multiplying the Mel-filterbank matrix by a power spectrum. Thus, extracting a log Mel-filterbank can be interpreted as a single real-valued layer without bias. We therefore compare the five-layer CLP-AM and the five-layer multiple-CVNN-AM with RVNN-AM as a fair comparison.

The CLP-AM has one complex-valued layer without bias and four real-valued layers. There is a logarithmic compression layer next to the absolute layer. The activation function for the real-valued layers is the sigmoid function. The multiple-CVNN-AM has two complex-valued layers without bias and three real-valued layers. The activation function for the complex-valued layers is the phase–amplitude log function described in section III-C, and the activation function for the real-valued layers is sigmoid function. The input features for CLP-AM and multiple-CVNN-AM are the FFT spectrum. FFT features are extracted and normalized in a similar manner to the description in section III-A. Normalized FFT features are spliced in time taking a context size of eleven frames.

All real-valued layers in CLP-AM and in multiple-CVNN-AM are fully connected. With respect to the complex-valued layers in CLP-AM and in multiple-CVNN-AM, we investigated three types of architecture:

- (a) Fully connected
  All complex-valued layers are fully connected.
- (b) Separately connected
  All complex-valued layers are divided according to context size, and each part of the complex-valued layer is fully connected. The output from each part of a complex-valued layer is concatenated as input to the absolute layer.
- (c) Separately connected with weight sharing
  The architecture is the same as in (b), but the weight matrix of the complex-valued layer is shared as in Network-in-Network [21].

Fig. 5 shows details for the RVNN-AM architecture, three CLP-AM architectures, and three multiple-CVNN-AM. For CLP-AM architectures, the number of nodes in a complex-valued layer is set to 440 (complex-valued) in (a), and to 40 (complex-valued) in (b) and (c) as for RVNN-AM. For multiple-CVNN-AM, the number of nodes in a complex-valued layer is set so that the number of parameters for multiple-CVNN-AM is close to the number of parameters for CLP-AM.

B. Results

Table V shows WERs for the RVNN-AM, the three CLP-AM architectures, and the three multi-CVNN-AM architectures. In Table V, "+BAMN" indicates that BAMN is applied to each complex-valued layer. "BAMN→Activ" and "Activ→BAMN" indicate whether BAMN is applied before or after the activation function.

We compared three CLP-AM and multi-CVNN-AM architectures. Table V shows that separately connected structures (b) and (c) achieve superior speech recognition performance as compared to fully connected structure (a).

We analyzed the weight matrix of the first layer in multi-CVNN-AM. Fig. 6–8 show the logarithm of magnitude weights in the first complex-valued layer in multiple-CVNN-AMs (a), (b), and (c), respectively. In Fig. 7, all complex-valued weight matrices for the first complex-valued layer are horizontally concatenated. Fig. 6 shows that the set of narrowband bandpass filters does not appear when all complex-valued layers are fully connected. Fig. 7 shows that some weights of the first complex-valued layer have a set of narrowband bandpass filters, whereas other weights do not. On the other hand, Fig. 8 clearly shows that the weight of the first complex-valued layer has a set of narrowband bandpass filters. The complex-valued layers in multiple-CVNN-AM (c) have lower degrees of freedom than do multiple-CVNN-AMs (a) and (b) by sharing the weight matrix. Consequently, optimizing the complex-valued layers in multiple-CVNN-AM (c) is easier than in the multiple-CVNN-AMs (a) and (b). Therefore, multiple-CVNN-AM (c) achieves superior speech recognition performance as compared to multiple-CVNN-AMs (a) and (b).

We also compared multiple-CVNN-AM with RVNN-AM and CLP-AM. Table V shows that multiple-CVNN-AM (c) with BAMN (BAMN→Activ) obtains performance equivalent to RVNN-AM when the SNR is 15 or 20 dB, and outperforms RVNN-AM when the SNR is 0, 5, and 10 dB. The relative improvements as compared with RVNN-AM under conditions of SNR 0, 5, or 10 dB are 1.99%, 7.45%, and 4.97%, respectively. The table also shows that multiple-CVNN-AM (c) with BAMN (BAMN→Activ) outperforms CLP-AM under all SNR conditions. The relative improvements as compared to CLP-AM (b) under conditions of SNR 0, 5, 10, 15, and 20 dB are 6.78%, 6.95%, 11.90%, 4.05%, and 7.86%, respectively. Table V show that multiple-CVNN-AM outperforms RVNN-AM and CLP-AM under noise conditions.

V. Conclusions

This paper explored applying an acoustic model with multiple-CVNN-AM and spliced features to speech recognition. We empirically determined that having a phase–magnitude-type activation function, not having bias, and having two complex-valued layers is optimal for multiple-CVNN-AM. We also proposed BAMN to more quickly and stably train complex-valued layers. Our experimental results showed that BAMN is effective for multiple-CVNN-AM. Experimental results also showed that multiple-CVNN-AM outperforms RVNN-AM and CLP-AM under noise conditions.

References

[1] A. Hirose, "Complex Valued Neural Networks," *Studies in Computational Intelligence*, vol. 32, Springer-Verlag, 2006.

[2] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Trans. on Neural Networks and learning systems*, 23(4): 541-551, 2012.

[3] M. Arjovsky, A. Shah, and Y. Bengio, "Unitary evolution recurrent neural networks," *arXiv preprint* arXiv:1511.06464,

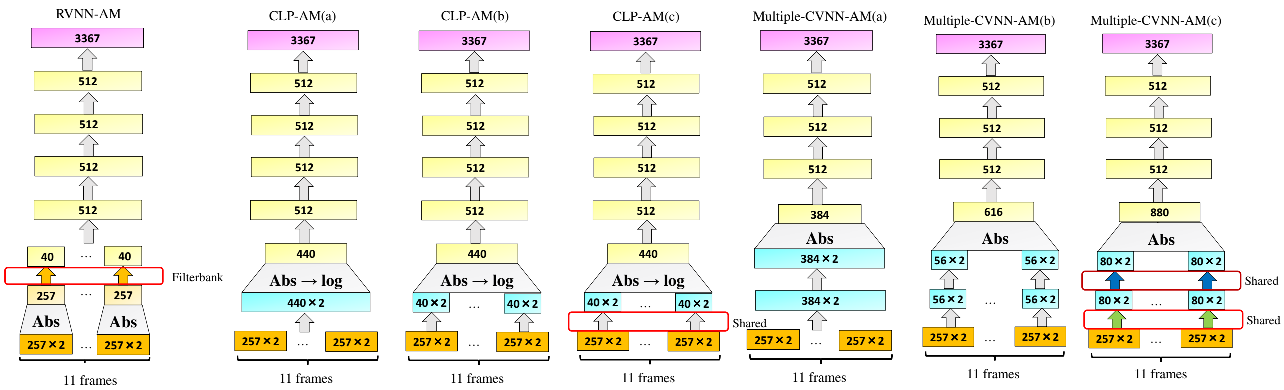


Fig. 5. RVNN-AM, CLP-AM, and multiple CVNN-AM architectures for acoustic models.

TABLE V
WERs for RVNN-AM, CLP-AM, and CVNN-AM

	# of Parameters	0 dB	5 dB	10 dB	15 dB	20 dB
RVNN-AM	2751311	30.73	13.15	8.24	**6.72**	6.73
CLP-AM (a)	5228791	38.31	16.41	10.19	8.91	8.17
CLP-AM (b)	2967191	32.31	13.08	8.91	7.41	7.25
CLP-AM (c)	2761591	34.75	15.13	9.85	8.35	7.85
Multi-CVNN-AM (a)	4915751	36.75	16.02	10.97	9.29	8.65
Multi-CVNN-AM (a) + BAMN(Activ→BAMN)	4916519	36.33	15.54	10.37	8.67	8.38
Multi-CVNN-AM (a) + BAMN(BAMN→Activ)	4916519	34.79	15.13	10.24	8.45	8.05
Multi-CVNN-AM (b)	2954103	36.24	14.92	9.73	8.47	7.66
Multi-CVNN-AM (b) + BAMN(Activ→BAMN)	2955335	36.20	14.37	9.09	7.99	7.62
Multi-CVNN-AM (b) + BAMN(BAMN→Activ)	2955335	34.20	13.99	8.81	8.06	7.27
Multi-CVNN-AM (c)	2757575	34.24	13.20	8.63	7.21	7.27
Multi-CVNN-AM (c) + BAMN(Activ→BAMN)	2759335	32.11	13.45	8.95	7.58	7.11
Multi-CVNN-AM (c) + BAMN(BAMN→Activ)	2759335	**30.12**	**12.17**	**7.85**	7.11	**6.68**

[4] I. Danihelka, G. Wayne, B. Uria, N. Kalchbrenner, and A. Graves, "Associative long short-term memory," *arXiv preprint* arXiv: 1602.03032, 2016.
[5] S. Wisdom, T. Powers, J. Hershey, J. L. Roux, and L. Atlas, "Full-capacity unitary recurrent neural networks," *Advances in Neural Information Processing Systems*, pp. 4880-4888, 2016.
[6] H. Tsuzuki, M. Kugler, S. Kuroyanagi, and A. Iwata, "An approach for sound source localization by complex-valued neural network," *IEICE Trans. Inf. & Syst.*, E96-D-10, pp. 2257-2265, 2013.
[7] C. Trabelsi, et al., "Deep complex networks," *arXiv preprint* arXiv:1705.09792, 2017.
[8] E. Variani, T. N. Sainath, I. Shafran, and M. Bacchiani, "Complex linear projection (CLP): A discriminative approach to joint feature extraction and acoustic modeling," *17th Proc. Interspeech*, San Francisco, USA, pp. 808-812, 2016.
[9] T. Nitta, "An extension of the back-propagation algorithm to complex numbers," *Neural Netw.*, vol. 10, no. 8, pp. 1391-1415, 1997.
[10] P. J. Werbos, "Backpropagation Through Time : What it Does and How to Do It," *Proc. of the IEEE*, vol. 78, no. 10, pp. 1550-1560, 1990.
[11] H. Leung and S. Haykin, "The complex backpropagation algorithm," *IEEE Trans. on Signal Processing*, vol. 39, pp. 2101-2104, 1991.
[12] A. Hirose, "Applications of complex-valued neural networks to coherent optical computing using phase-sensitive detection scheme," *Information Sciences-Applications*, vol. 2, pp. 103-117, 1994.
[13] A. J. Noest, "Associative memory in sparse phasor neural networks," *Europhysics Letters*, volume 6, pp. 469-474, 1988.
[14] G. M. Georgiou and C. Koutsougeras, "Complex domain backpropagation," *IEEE Trans. on Circuits and Systems-II*, vol. 39, pp. 330-334, 1992.
[15] S. Scardapane, S. V. Vaerenbergh, A. Hussain, and A. Uncini, "Complex-valued Neural Networks with Non-parametric Activation Functions," *arXiv preprint* arXiv: 1802.08026, 2018.
[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint* arXiv:1502.03167, 2015.
[17] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," *Proceedings of the workshop on Speech and Natural Language. Association for Computational Linguistics*, pp. 357-362, 1992.
[18] D. Povey, et al., "The kaldi speech recognition toolkit," *Proc. IEEE 2011 Workshop on Automatic Speech Recognition and Understanding (ASRU2011)*. IEEE, 2011.
[19] K. Vesely, A. Ghoshal, L. Burget, and D. Povey. "Sequence-discriminative training of deep neural networks," *Interspeech*, 2013.
[20] J. Barker, R. Marxer, E. Vincent, and S.Watanabe, "The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," *Proc. IEEE 2015 Workshop on Automatic Speech Recognition and Understanding (ASRU2015)*. IEEE, 2015.
[21] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint* arXiv:1312.4400, 2013.

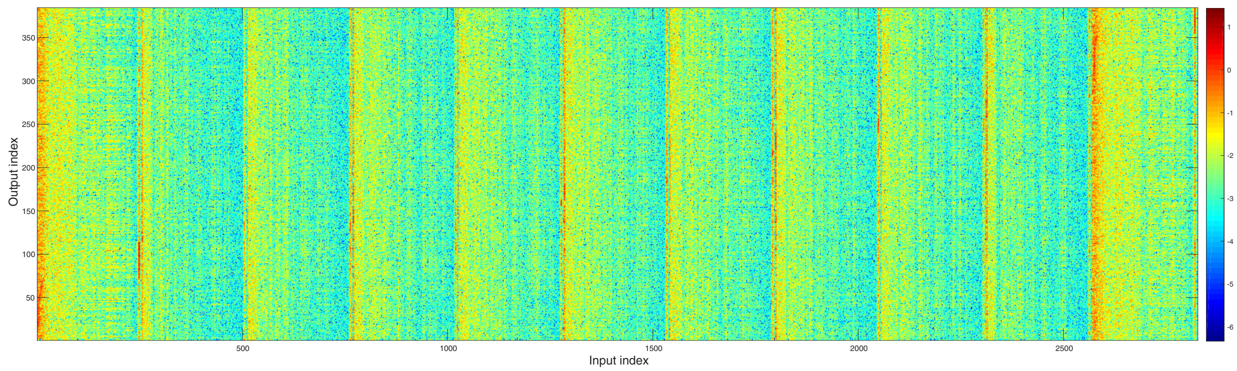


Fig. 6. Logarithm of magnitude weights in the first complex-valued layer in multiple-CVNN-AM (a).

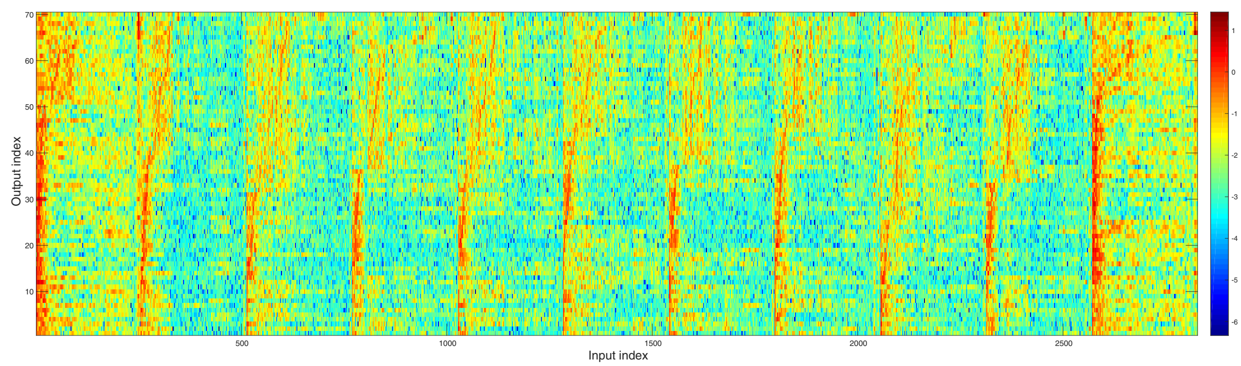


Fig. 7. Logarithm of magnitude weights of the first complex-valued layer in multiple-CVNN-AM (b). The first complex-valued layer has eleven complex-valued weight matrices. All complex-valued weight matrices are horizontally concatenated.

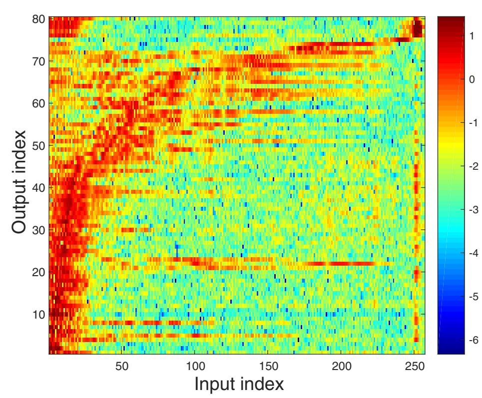


Fig. 8. Logarithm of magnitude weights in the first complex-valued layer in multiple-CVNN-AM (c).