

Article

Efficient Speech Signal Dimensionality Reduction Using Complex-Valued Techniques

Sungkyun Ko ¹  and Minho Park ^{2,*} 

¹ Department of AI IT Convergence, Soongsil University, Seoul 06978, Republic of Korea; kosungkyun@naver.com

² School of Electronic Engineering, Soongsil University, Seoul 06978, Republic of Korea

* Correspondence: mhp@ssu.ac.kr

Abstract: In this study, we propose the CVMFCC-DR (Complex-Valued Mel-Frequency Cepstral Coefficients Dimensionality Reduction) algorithm as an efficient method for reducing the dimensionality of speech signals. By utilizing the complex-valued MFCC technique, which considers both real and imaginary components, our algorithm enables dimensionality reduction without information loss while decreasing computational costs. The efficacy of the proposed algorithm is validated through experiments which demonstrate its effectiveness in building a speech recognition model using a complex-valued neural network. Additionally, a complex-valued softmax interpretation method for complex numbers is introduced. The experimental results indicate that the approach yields enhanced performance compared to traditional MFCC-based techniques, thereby highlighting its potential in the field of speech recognition.

Keywords: mel-frequency cepstral coefficients; complex-valued neural network error correction; backpropagation; complex-valued softmax



Citation: Ko, S.; Park, M. Efficient Speech Signal Dimensionality Reduction Using Complex-Valued Techniques. *Electronics* **2024**, *13*, 3046. <https://doi.org/10.3390/electronics13153046>

Academic Editors: Xin Geng, Padma Iyengar and Elke Pulvermüller

Received: 3 June 2024

Revised: 31 July 2024

Accepted: 31 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Today, speech signal processing and recognition play crucial roles in various application domains. These technologies are utilized for diverse purposes, such as converting speech commands into text or identifying the owner of a voice. The advancement of speech recognition technology can improve human–machine interaction, thereby enhancing productivity and efficiency across various industries. However, speech signals possess highly diverse forms and high-dimensional characteristics, making effective processing and recognition a persistent challenge. The diversity and high dimensionality of speech signals make dimensionality reduction and pattern recognition difficult, leading to increased computational costs. Therefore, efficiently reducing the dimensionality of speech signals and improving the performance of neural network-based speech recognition models is a significant challenge.

Speech processing and recognition technologies are essential in various applications, such as voice interfaces, virtual assistants, and automatic translation. Achieving more accurate and efficient speech signal processing and recognition results can significantly enhance the user experience of related services. Consequently, continuous research is being conducted in the field of speech signal processing and recognition. Speech signals exhibit high-dimensional characteristics, making effective dimensionality reduction challenging. Moreover, accurate recognition of diverse speech patterns requires intricate neural network models, which, in turn, increases computational costs [1]. Therefore, effectively reducing the dimensionality of speech signals without information loss and simultaneously enhancing the accuracy and efficiency of neural network-based speech recognition models is a crucial task.

Despite the rich literature on complex-valued networks, there have been few examples of their application in audio or speech processing. Given that the human auditory system

processes sound in the frequency domain, there is also meaning in exploring complex-valued neural networks for audio processing rather than relying solely on real-valued neural networks in the time domain [2].

Recent research has made significant progress in speech recognition by combining traditional methods like mel-frequency cepstral coefficients (MFCCs) with neural networks [3]. While the MFCC technique has been widely used for speech feature extraction, it has the drawback of being unable to perform lossless dimensionality reduction, as it uses only real values. In other words, reducing the number of MFCCs leads to information loss, which decreases speech recognition accuracy. These methods still exhibit limited effectiveness, and improved approaches beyond traditional learning methods are necessary.

Utilizing the complex representation of MFCC allows for the consideration of both real and imaginary parts, enabling lossless dimensionality reduction. Additionally, employing complex-valued neural networks can reflect phase changes through complex multiplication operations [4], better representing the characteristics of speech signals. This approach is expected to enable more efficient dimensionality reduction and neural network-based speech recognition compared to existing methods.

In this study, we propose the CVMFCC-DR algorithm for efficient and lossless speech signal dimensionality reduction. By combining CVMFCC-DR with complex-valued neural networks, this technique can demonstrate superior speech recognition performance compared to existing methods.

The main contributions of this study are as follows:

- Proposing an efficient dimensionality reduction technique (CVMFCC-DR) utilizing the complex representation of MFCCs.
- Enhancing speech recognition performance by combining complex-valued MFCCs and complex-valued neural networks.
- Proposing a new interpretation method for a complex-valued softmax function with complex inputs.

This paper begins by introducing the existing MFCC technique and complex-valued neural networks. It then provides a detailed explanation of the proposed CVMFCC-DR algorithm and the learning method for complex-valued neural networks [5]. Subsequently, the effectiveness of the proposed method is experimentally validated, and the results are analyzed. Finally, the paper concludes with a summary of the study's findings and suggests directions for future research.

2. Related Works

In the field of speech recognition, various neural network-based approaches and techniques are utilized. Different types of neural networks, such as CNNs, RNNs, LSTM, and DNNs, are used, each with their own advantages and disadvantages [3]. Moreover, hybrid models that combine traditional methods with neural network structures are also being researched. Hybrid models can optimize performance by combining the strengths of each structure, but they also come with the drawback of increased complexity in implementation and interpretation [6].

Amidst these developments, traditional feature extraction techniques like the MFCC technique are still widely used as inputs for neural networks [3]. Additionally, exploring novel neural architectures such as complex-valued neural networks could uncover new perspectives and potential breakthroughs in speech recognition capabilities. Recently, there have been studies showing that FPGAs can be used to significantly improve complex number computation speed [7]. These research results empirically demonstrate that the hardware implementation of complex number operations can bring about practical performance improvements. Furthermore, a recent study implemented complex-valued neural networks on FPGAs for classification, demonstrating superior accuracy compared to real-valued neural networks and achieving significant speed improvements over CPU-based systems [8]. This gives rise to the expectation that in the future, CPUs that currently only support real number operations may one day support complex number operations as well.

Based on these studies, utilizing complex number representations enables the encoding of real number pairs as single complex numbers, effectively reducing the input dimension by half. This reduction in dimensionality decreases the number of model parameters and enhances computational efficiency. Consequently, experimental evidence shows that complex-valued neural networks can achieve equal or superior performance with fewer parameters compared to conventional real-valued neural networks.

As research continues to push the boundaries, a synergistic combination of proven methods and innovative approaches may pave the way for the next generation of robust and high-performing speech recognition systems.

3. Background Knowledge

Various studies are being conducted in the field of speech recognition and classification, and in this section, we introduce MFCCs (mel-frequency cepstral coefficients) [3]. The MFCC technique is one of the feature extraction methods commonly used in the fields of speech signal processing and speech recognition. The MFCC technique represents the frequency characteristics of speech signals and is primarily utilized to effectively represent and recognize speech signals. The basic principle of the MFCC technique is illustrated in Figure 1.



Figure 1. An overview of the process of the MFCC technique.

Pre-emphasis: Pre-emphasis is essentially a high-pass filter. Due to the structure of the human body, high-frequency components tend to decrease in the actual emitted sound during speech production. Therefore, pre-emphasis is the process of applying a high-pass filter to amplify the attenuated high-frequency components first.

Framing and Windowing: Speech signals are divided into short segments called frames for analysis. This is because speech characteristics remain relatively stable over brief periods. These frames slightly overlap to capture speech changes more effectively. Each frame is processed using a window technique, which enhances analysis. Common window types include Hamming and Hanning windows. This framing method allows for more accurate analysis of complex speech data, which is crucial for developing speech recognition and synthesis technologies. While specific implementation details may vary, these fundamental principles are widely used in speech processing systems.

Discrete Fourier Transform: For calculating the power spectrum, DFT (Discrete Fourier Transform) is used to compute the power spectrum for each frame obtained earlier. This process enables the discernment of the frequency-specific energy distribution of the speech signal across each frame.

Mel-Frequency Filter Bank: The mel band-pass filter is a set of filters designed to extract speech signals in a nonlinear manner, mimicking the human auditory system. Typically constructed with 40 triangular filters, it analyzes the frequency characteristics of speech signals.

Discrete Cosine Transform: DCT (Discrete Cosine Transform) expresses a finite sequence of data points as a sum of cosine functions oscillating at different frequencies. In the MFCC process, DCT is applied to the mel filter bank to select acceleration coefficients or to separate the relationship in log-spectral magnitudes from the filter bank.

Speech recognition using MFCCs can be effectively employed in various environments [9], and this method can achieve high accuracy when combined with deep learning models. Therefore, MFCC serves as a crucial input for deep learning models for speech recognition [10].

4. Proposed Method

In this section, we propose a method for efficiently performing speech signal dimensionality reduction using complex-valued MFCCs and complex-valued neural networks. An overview of the proposed approach is presented in Figure 2. This figure illustrates the proposed approach for voice classification using complex-valued neural networks. The process commences with voice input, followed by feature extraction using mel-frequency cepstral coefficients (MFCCs). These features are then subjected to dimensionality reduction and transformed into complex-valued representations, resulting in complex-valued MFCCs. The processed features are subsequently inputted into a complex-valued neural network. The network's performance is enhanced through the selection of appropriate activation functions and learning algorithms. Ultimately, the neural network's output is utilized for classification purposes.

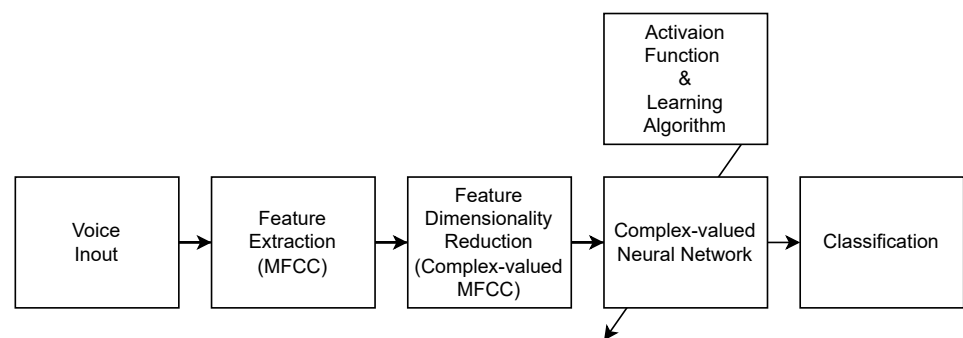


Figure 2. Overview of the proposed approach.

Complex-valued MFCCs: The number of MFCCs used in speech recognition can vary depending on specific applications and models. However, typically, MFCCs ranging from 13 to 40 are used. This range is known to effectively operate in various speech recognition systems. Generally, a higher number of MFCCs can allow the model to capture more frequency and spectral detail, but it also increases the computational cost of the model. Therefore, the selected number of MFCCs is determined considering a balance between computational efficiency and model performance.

When representing physical problems using complex numbers, there are two main methods. One is a simple substitution method, and the other is a more elegant approach known as mathematical equivalence, like the Fourier Transform. Substitution takes two real physical parameters and places one in the real part of the complex number and one in the imaginary part. This allows the two values to be manipulated as a single entity, i.e., a single complex number [11]. The method proposed for dimensionality reduction in this paper involves converting n MFCC values into complex numbers. The first MFCC is used as the real part, and the second MFCC is used as the imaginary part of the complex number. Subsequently, each MFCC alternates between being the real and imaginary parts within the complex number. This pattern continues up to the n th MFCC, resulting in $n/2$ complex numbers. For example, when we have n MFCC values, they could be as follows:

$$[x_1, x_2, x_3, x_4, \dots, x_{n-1}, x_n]$$

In this case, the first MFCC x_1 is used as the real part of the first complex number. The second MFCC x_2 is used as the imaginary part of the first complex number. The third MFCC x_3 is used as the real part of the second complex number. The fourth MFCC x_4 is used as the imaginary part of the second complex number. This pattern continues alternating between real and imaginary parts up to the n th MFCC x_n , which becomes the imaginary part of the $(n/2)$ -th complex number. Therefore, with n MFCC values, $n/2$ complex numbers are generated. The first complex number is $x_1 + jx_2$, the second complex

number is $x_3 + jx_4$, and so on until the last complex number, which is $x_{n-1} + jx_n$. We have illustrated this explanation in Figure 3.

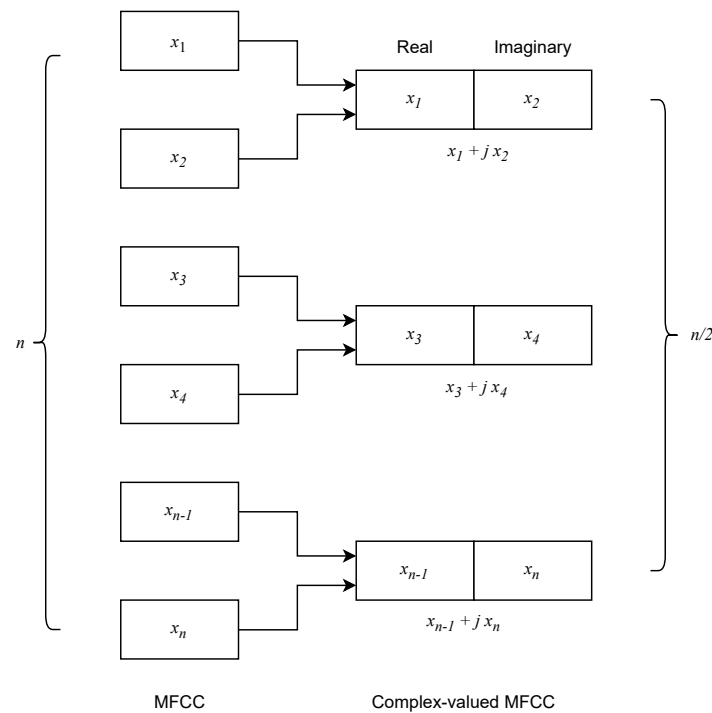


Figure 3. Process of generating complex-valued MFCCs.

Complex-Valued Neural Network: Complex-valued neural networks are neural network models that use complex numbers as input values or weights, and they have the advantage of handling richer information compared to real-valued neural networks. Furthermore, various research studies have been conducted in various fields, and complex-valued dimensionality reduction methods or classification and detection using complex numbers are being effectively utilized. For this purpose, the multiplication of complex inputs and complex weights is performed, which implies rotation in the complex plane and thus introduces changes in phase [5]. The sum of the multiplication of input X and weight W can be represented as Equations (1) and (2).

$$Z = W_0 + W_1X_1 + W_2X_2 + \dots + W_nX_n \quad (1)$$

$$Z = \sum_n W_n \cdot X_n \quad (2)$$

Activation Function and Learning Algorithm: In complex-valued neural networks, a phase-dependent activation function is used, forcing the magnitude information to be normalized to a unit circle of 1 and only the extraction of the phase information, as shown in Equation (3).

$$P(Z) = \frac{Z}{|Z|} = e^{j \cdot \arg(Z)} \quad (3)$$

The learning algorithm of complex-valued neural networks is as follows. When there is a desired target t and an output value z , as shown in Figure 4, the weights are adjusted according to Equation (4) to gradually move in the direction of the desired target [12].

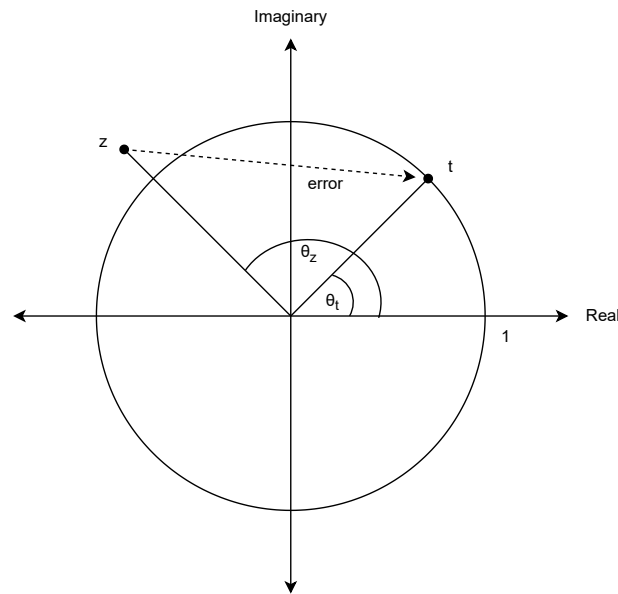


Figure 4. Learning algorithm for CVNN.

$$t = \sum_n (W_n + \Delta W_n) \cdot X_n \quad (4)$$

The error with respect to the training target t is $e = t - z$, as shown in Equation (5).

$$e = \sum_n (W_n + \Delta W_n) \cdot X_n - \sum_n (W_n) \cdot X_n = \sum_n (\Delta W_n) \cdot X_n \quad (5)$$

To simplify and expedite the calculation of the error with respect to weight changes, assuming that each node contributes equally to the error, the summation symbol σ is removed from the equation, and both sides are divided by the number of contributing nodes N in the complex-valued neural network for computational convenience. The resulting equation is Equation (6).

$$\Delta W_n \cdot X_n = \frac{e}{N} \quad (6)$$

To obtain ΔW_n in order to change the weights, multiply both sides of Equation (6) by X_n^{-1} . X_n^{-1} becomes $\overline{X_n}$ when multiplied by the complex conjugate $\overline{X_n}$, with n being both the numerator and denominator; thus, $X_n^{-1} = \frac{\overline{X_n}}{|X_n|^2}$, where $|X_n| = 1$ for X_n within the unit circle. Therefore, $X_n^{-1} = \overline{X_n}$, and Equation (6) is derived again as Equations (7) and (8).

$$\Delta W_n \cdot X_n \cdot X_n^{-1} = \frac{e}{N} \cdot X_n^{-1} = \frac{e}{N} \cdot \overline{X_n} \quad (7)$$

$$\Delta W_n = \frac{e}{N} \cdot \overline{X_n} \quad (8)$$

In this way, the obtained change in weights can be observed in Equation (8). This change represents the adjustment of complex weights for each node contributing to the error with respect to the training target and is uniformly applied across the entire network. Through this process, complex-valued neural networks adjust closer to the target utilizing phase-dependent activation functions and efficient learning algorithms while handling rich information through complex multiplication operations between inputs and weights [13].

Complex Domain Backpropagation: When using hidden layers, the error from the output layer is backpropagated to the hidden layers to calculate how much each node

in the hidden layers contributed to that error. Based on this, the weights between the hidden and output layers need to be adjusted appropriately. This process is called error backpropagation, through which the neural network can adjust the weights to minimize the error in the output values during learning. The structure of a multi-layer complex-valued neural network is illustrated in Figure 5.

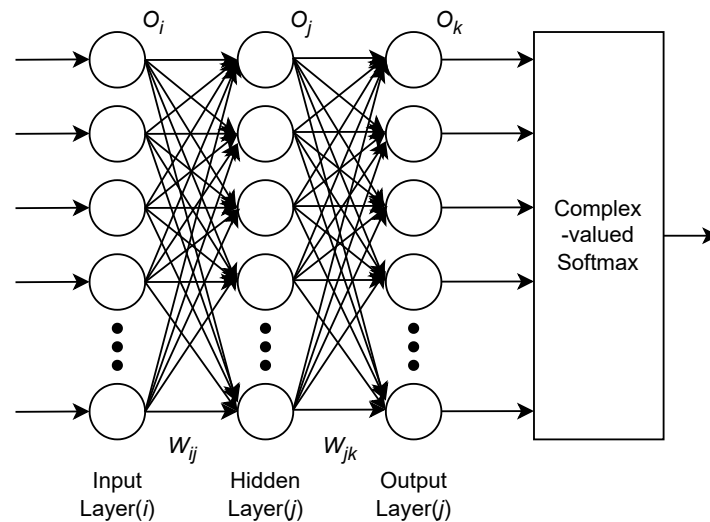


Figure 5. Multi-layer CVNN.

In this paper, an error correction method is used for the output layer k to update the weights, while an error backpropagation method is employed for the hidden layer j to examine its influence on the output. In other words, the output layer k utilizes the error correction method to avoid falling into local minima, while the error backpropagation method is applied to the hidden layer j to ensure better distribution of errors from the output layer to the hidden layer. This combination of error backpropagation in the hidden layer and error correction in the output layer can be expected to improve the problem of the entire neural network getting stuck in local minima while enabling faster convergence. The error backpropagation method for complex-valued neural networks is as follows.

When using gradient descent for weight updates in complex-valued neural networks, Wirtinger calculus is employed for differentiating complex functions. As complex functions comprise real and imaginary parts, Wirtinger calculus conducts differentiation separately for the real and imaginary parts. For a complex number $z = x + jy$ defined with real and imaginary parts x and y , respectively, and a complex function $f(z) = u(x, y) + jv(x, y)$, the Wirtinger derivatives are defined as follows [14].

$$\frac{\partial f}{\partial z} = \frac{1}{2} \left(\frac{\partial f}{\partial x} - j \frac{\partial f}{\partial y} \right) = \frac{1}{2} \left(\frac{\partial u}{\partial x} + j \frac{\partial v}{\partial x} - j \frac{\partial u}{\partial y} + \frac{\partial v}{\partial y} \right) \quad (9)$$

$$\frac{\partial f}{\partial \bar{z}} = \frac{1}{2} \left(\frac{\partial f}{\partial x} + j \frac{\partial f}{\partial y} \right) = \frac{1}{2} \left(\frac{\partial u}{\partial x} + j \frac{\partial v}{\partial x} + j \frac{\partial u}{\partial y} - \frac{\partial v}{\partial y} \right) \quad (10)$$

$$df = \frac{\partial f}{\partial z} dz + \frac{\partial f}{\partial \bar{z}} d\bar{z} \quad (11)$$

In this context, $\bar{z}$ denotes the complex conjugate of z . In the Wirtinger derivative method, complex functions can be treated as extensions of real functions by dividing them into real and imaginary parts for differentiation. The following describes the updating process of complex variables using Step Size $\alpha/2$ and Loss Function L .

$$x_{n+1} = x_n - \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial x} \quad (12)$$

$$y_{n+1} = y_n - \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial y} \quad (13)$$

By substituting $z_{n+1} = x_{n+1} + j y_{n+1}$ into Equations (12) and (13) and then substituting the Wirtinger derivative at the end of the derivation process, it becomes as follows.

$$\begin{aligned} z_{n+1} &= x_n - \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial x} + j \left(y_n - \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial y} \right) \\ &= x_n - \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial x} + j y_n - j \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial y} \\ &= x_n + j y_n - \left(\frac{\alpha}{2}\right) \left(\frac{\partial L}{\partial x} + j \frac{\partial L}{\partial y} \right) \\ &= z_n - \alpha \frac{1}{2} \left(\frac{\partial L}{\partial x} + j \frac{\partial L}{\partial y} \right) \\ &= z_n - \alpha \frac{\partial L}{\partial \bar{z}} \end{aligned} \quad (14)$$

The partial derivative of L with respect to the complex conjugate $\frac{\partial L}{\partial \bar{z}}$ is referred to as the Conjugate Wirtinger Derivative. This demonstrates how the Conjugate Wirtinger Derivative can be utilized to simplify the update formula for complex variables. In terms of using the above method, rewriting Equation (5) as an output layer error yields Equation (15), where t_k represents the target value, and o_k represents the actual output value of output layer k .

$$e_k = t_k - o_k \quad (15)$$

And the Loss Function is defined as Equation (16).

$$L = e_k^2 = |t_k - o_k| \overline{|t_k - o_k|} \quad (16)$$

If the weights are large, the error is more likely to propagate back to the hidden layers. The hidden layer error e_j can be expressed by the following equation.

$$e_j = W_{jk}^T \cdot e_k \quad (17)$$

Using the result of Equation (14), the gradient of the error function with respect to the weight $\bar{W}_{ij}$ between input layer i and hidden layer j can be expressed as follows. Here, o_i is the output value of input layer i , and o_j is the output value of hidden layer j .

$$\frac{\partial L}{\partial \bar{W}_{ij}} = \frac{\partial L}{\partial o_j} \cdot \frac{\partial o_j}{\partial \bar{W}_{ij}} \quad (18)$$

$$\frac{\partial L}{\partial \bar{W}_{ij}} = -e_j \cdot \frac{\partial}{\partial \bar{W}_{ij}} P \left(\sum \bar{W}_{ij} \cdot \bar{o}_i \right) \quad (19)$$

$$\frac{\partial L}{\partial \bar{W}_{ij}} = -e_j \cdot P' \cdot \bar{o}_i \quad (20)$$

In Equations (19) and (20), P represents the phase-dependent activation function used in the hidden layer output, and P' is the derivative of the dependent activation function. The updating of hidden layer weights can be expressed as follows.

$$new\ W_{ij} = old\ W_{ij} - lr \cdot \frac{\partial L}{\partial \bar{W}_{ij}} \quad (21)$$

When dealing with complex variables, it is necessary to use $\frac{\partial L}{\partial \bar{W}_{ij}}$ instead of $\frac{\partial L}{\partial W_{ij}}$ for updating. Here, lr plays a role in adjusting the intensity of the change to prevent overshooting, and it is called the learning rate. Additionally, the derivative of the activation function P' is a complex function, and to find the derivative of a complex function, the Wirtinger derivative is used [15].

Complex-Valued Softmax: The softmax function, which is used as the activation function in the output layer, takes input values and normalizes each value to be between 0 and 1. It ensures that the sum of the normalized values is always 1. This function allows for the interpretation of results as probabilities. It can be expressed as Equation (22).

$$Softmax(Z)_i = \frac{e^{Z_i}}{\sum_{i=1}^n e^{Z_i}} \quad (22)$$

However, the softmax function can only interpret results as probabilities for real number inputs, and for complex number inputs, a different interpretation method is needed. In this paper, we propose a complex-valued softmax interpretation method for complex number inputs. The result of the softmax function for complex number inputs yields complex output values, and since the sum of all output values is $1 + 0j$, we propose an algorithm that calculates the distance between the output values and $1 + 0j$ in the complex plane and selects the index of the closest value. This can be expressed as Equation (24).

$$distance_i = |Softmax(Z)_i - \omega| \quad (23)$$

$$closest_index = argmin(distance_i) \quad (24)$$

Here, $\omega = 1 + 0j$ is defined. Consequently, we decide to classify the complex number corresponding to $closest_index$ as class 1.

5. Performance Analysis

In our experiments of speech signal recognition, a detailed comparison between the results of real-valued neural networks and complex-valued neural networks was conducted. Performance analysis was carried out alongside the pre-processing of complex-valued mel-frequency cepstral coefficients (MFCCs) for dimensionality reduction. Additionally, the impact of complex-valued MFCCs on speech recognition results was observed. Specifically, in recognizing given speech signals using neural networks, the influence of MFCC values and their complexification effect on model performance was evaluated and compared to determine whether it is possible to enhance performance while reducing the number of parameters.

Experimental Method: Figure 6 depicts the architecture of a multi-layer complex-valued neural network used for training and testing with a dataset consisting of 8000 short speech samples, each containing eight different words and lasting less than one second. The speech data used in this study are part of the speech command dataset collected by Google and distributed under the CC BY license.

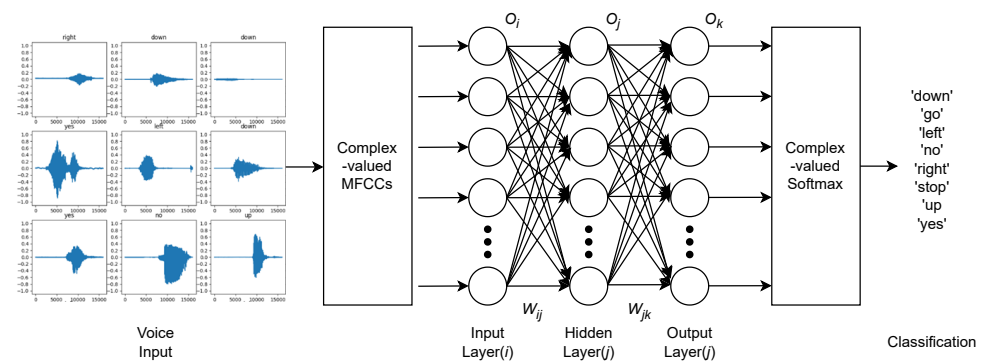


Figure 6. Experimental setup of multi-layer CVNN.

Complex-valued neural networks are structurally similar to real-valued neural networks, but the difference lies in the use of complex numbers, which introduces phase changes as the sum of products between inputs and weights rotates in the complex plane. These phase changes play an important role in complex-valued neural networks. The complex-valued neural network consists of an input layer with dimension-reduced complex-valued MFCCs obtained from the speech data through the complexification process shown in Figure 3, a hidden layer with 500 nodes, and an output layer with 8 nodes for classification. All nodes and weights use complex numbers. The final results were classified at the output layer. The dataset comprised 7000 samples for training and 1000 samples for testing. Eight-word audio clips were used as data, and experiments were conducted by dividing the input data based on whether complex-valued MFCC transformation was applied or not. In the case of using real-valued MFCCs, a real-valued neural network was employed, while a complex-valued neural network was used with complex-valued MFCCs. Both the real-valued and complex-valued neural networks were trained under the same conditions: $\text{lr} = 0.01$, batch size = 50, and number of epochs = 50. For the input data, the results of three different dimensionality reduction pre-processing steps were compared with the real-valued neural network. These pre-processing steps included MFCC dimensions of 20, 10, and 5. The length of the input signal for the neural network can be determined as shown in Equation (25). Specifically, the input speech signal was divided into 32 frames, and MFCCs were generated for each frame.

$$\text{The length of input signal data} = \text{MFCCs} \times \text{frames} \quad (25)$$

For instance, in the case of complex-valued MFCC 20, real-valued MFCC 40 is computed first. Using Equation (25), the length of the input signal for real-valued MFCC 40 becomes $40 \times 32 = 1280$. If we then convert these 40 MFCCs into complex numbers, as illustrated in Figure 3, dimensionality reduction is achieved to half the length, resulting in 16 frames. Consequently, the input signal length becomes $40 \times 16 = 640$, which is equivalent to the input signal length of real-valued MFCC 20. Similar dimensionality reduction can be performed for complex-valued MFCC 10 and complex-valued MFCC 5 using the same method. The total number of parameters for the entire neural network obtained through this process is presented in Table 1. In this table, $I(n)$ represents the number of neurons in the input layer, $H(n)$ represents the number of neurons in the hidden layer, and $O(n)$ represents the number of neurons in the output layer.

Table 1. Total number of parameters in neural network.

MFCC	Real-Valued MFCC	Complex-Valued MFCC
20	$I(20 \times 32 = 640) \times H(500) + H(500) \times O(8) = 324,000$	$I(40 \times 16 = 640) \times H(500) + H(500) \times O(8) = 324,000$
10	$I(10 \times 32 = 320) \times H(500) + H(500) \times O(8) = 164,000$	$I(20 \times 16 = 320) \times H(500) + H(500) \times O(8) = 164,000$
5	$I(5 \times 32 = 160) \times H(500) + H(500) \times O(8) = 84,000$	$I(10 \times 16 = 160) \times H(500) + H(500) \times O(8) = 84,000$

Complex-valued MFCC 20 vs. Real-valued MFCC 20: As shown in Figure 7, complex-valued MFCC 20 achieved a maximum training accuracy of 99.40% and a testing accuracy of 84.90% despite having the same input data length as real-valued MFCC 20, which achieved a maximum training accuracy of 98.13% and a testing accuracy of 83.10%. The results obtained using complex-valued MFCCs outperformed those obtained using real-valued MFCCs in terms of both training and testing accuracy.

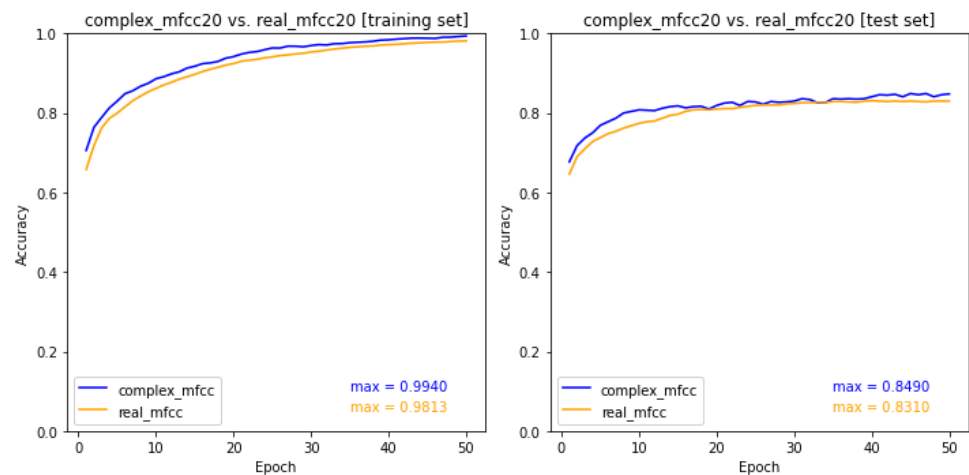


Figure 7. Comparison of complex-valued MFCC 20 and real-valued MFCC 20 results.

Complex-valued MFCC 10 vs. Real-valued MFCC 10: As shown in Figure 8, complex-valued MFCC 10 achieved a maximum training accuracy of 96.90% and a testing accuracy of 83.80% despite having the same input data length as real-valued MFCC 10, which achieved a maximum training accuracy of 94.24% and a testing accuracy of 81.20%. The results obtained using complex-valued MFCCs outperformed those obtained using real-valued MFCCs in terms of both training and testing accuracy.

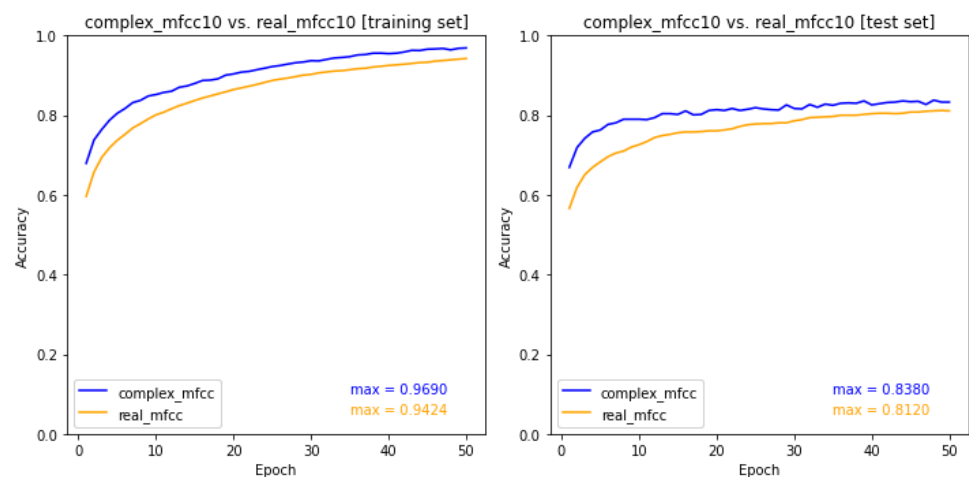


Figure 8. Comparison of complex-valued MFCC 10 and real-valued MFCC 10 results.

Complex-valued MFCC5 vs. Real-valued MFCC5: As shown in Figure 9, complex-valued MFCC 5 achieved a maximum training accuracy of 92.33% and a testing accuracy of 81.40% despite having the same input data length as real-valued MFCC 5, which achieved a maximum training accuracy of 86.34% and a testing accuracy of 74.90%. The results obtained using complex-valued MFCCs outperformed those obtained using real-valued MFCCs in terms of both training and testing accuracy.

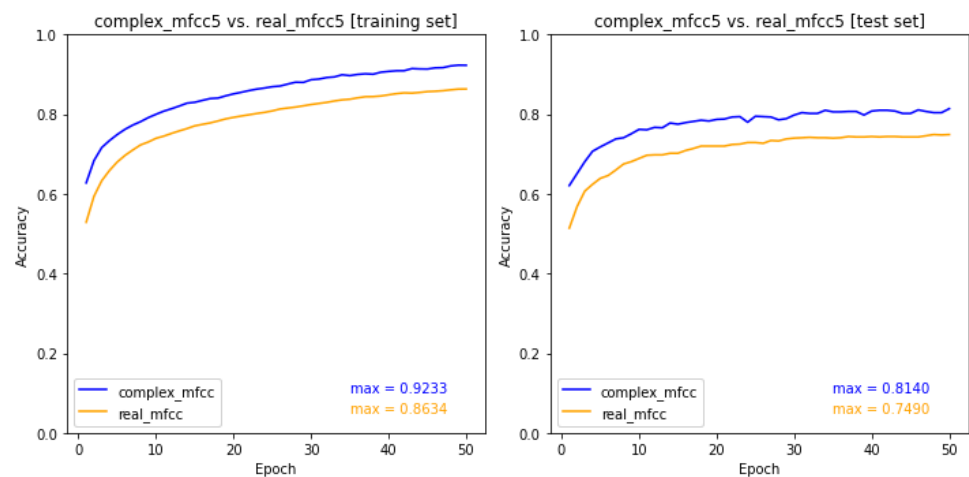


Figure 9. Comparison of complex-valued MFCC 5 and real-valued MFCC 5 results.

Overall Analysis Results: Upon reviewing the experimental results, it was observed that using complex-valued MFCCs yields higher performance compared to using real-valued MFCCs, even with the same input data length. Therefore, it was confirmed that employing complex-valued MFCCs and complex-valued neural networks leads to better results. The variation in accuracy according to the number of MFCC features is illustrated in Figure 10. Although not shown here, the same results were obtained using the F1 score.

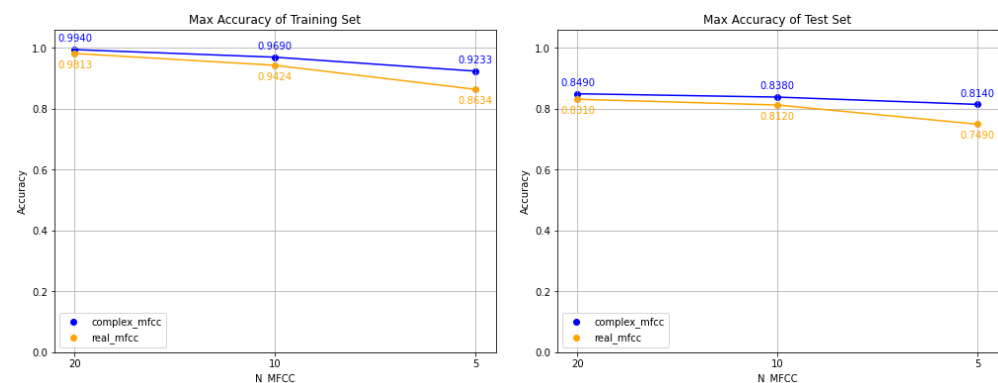


Figure 10. Accuracy variation with No. of MFCCs.

If there is a sufficient number of MFCCs, with increasing epochs, it appears that the performance using real numbers begins to converge towards that using complex numbers. However, Figure 9 shows a different outcome. Complex-valued MFCC 5 has 5 dimensionally reduced complex MFCCs obtained from 10 MFCCs through the complexification process shown in Figure 3. A real-valued MFCC 5 with only 5 MFCCs cannot catch up with complex-valued MFCC 5 even as the epochs increase. In other words, when the number of MFCCs is low, even as the epochs increase, the results of real-valued MFCCs do not match the performance of complex-valued MFCCs. As shown in Figure 10, despite having the same number of parameters, the results of complex-valued MFCC 5 are superior to those of real-valued MFCC 5. Therefore, it was observed that when using complex-valued MFCCs, the change in accuracy as the number of MFCC features decreases is smoother compared to when using real-valued MFCCs.

6. Conclusions

In this paper, we introduced complex-valued neural networks to address the problem of speech recognition. Our experiments utilized complex-valued MFCCs as inputs for a

complex-valued neural network. The results demonstrated superior performance when employing complex-valued MFCCs, with a more gradual change in accuracy as the number of MFCC features decreased, compared to using real-valued MFCCs. This suggests that complex-valued MFCCs can help to maintain performance with fewer features. These findings provide valuable insights into the potential of complex-valued neural networks and demonstrate the effectiveness of combining complex-valued MFCCs with complex-valued neural networks in speech recognition.

Despite these promising results, we acknowledge the certain limitations of our approach. This study was conducted on a limited dataset, which may not fully represent the diversity of real-world speech patterns. Moreover, the complex-valued neural network architecture described in this study is relatively simple and may not capture all the intricacies of speech signals.

Looking ahead, we aim to build upon this work to further enhance the accuracy and performance of speech signal processing and recognition models. Our future research will focus on designing more sophisticated complex-valued neural network models and conducting experiments on diverse datasets to improve performance. Additionally, we plan to explore the application of complex-valued neural networks to other fields beyond speech recognition, potentially expanding their use in various signal processing domains [16].

Author Contributions: Conceptualization, S.K.; project administration, M.P.; supervision, M.P.; writing—original draft, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was jointly supported by the National Research Foundation of Korea (NRF) via a grant provided by the Korea government (MSIT) (grant no. NRF-2023R1A2C1005461), and by the MSIT (Ministry of Science and ICT), Korea, under the Convergence Security Core Talent Training Business Support Program (IITP-2024-RS-2024-00426853) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation).

Data Availability Statement: The original contributions presented in the study are included in the article. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tebelskis, J. Speech Recognition Using Neural Networks. Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 1995.
2. Sarroff, A.M. Complex Neural Networks for Audio. Ph.D. Thesis, Dartmouth College, Hanover, NH, USA, 2018.
3. Abdul, Z.K.; Al-Talabani, A.K. Mel Frequency Cepstral Coefficient and Its Applications: A Review. *IEEE Access* **2022**, *10*, 122136–122158. [\[CrossRef\]](#)
4. Barrachina, J.A.; Ren, C.; Vieillard, G.; Morisseau, C.; Ovarlez, J.P. Theory and Implementation of Complex-Valued Neural Networks. *arXiv* **2023**, arXiv:2302.08286.
5. Aizenberg, I. *Complex-Valued Neural Networks with Multi-Valued Neurons*; Springer: Berlin/Heidelberg, Germany, 2016.
6. Graves, A.; Jaitly, N.; Mohamed, A.R. Hybrid speech recognition with Deep Bidirectional LSTM. In Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, 8–12 December 2013; pp. 273–278. [\[CrossRef\]](#)
7. Paz, P.; Garrido, M. Efficient Implementation of Complex Multipliers on FPGAs Using DSP Slices. *J. Signal Process. Syst.* **2023**, *95*, 543–550. [\[CrossRef\]](#)
8. Ahmad, M.; Zhang, L.; Chowdhury, M.E.H. FPGA Implementation of Complex-Valued Neural Network for Polar-Represented Image Classification. *Sensors* **2024**, *23*, 2627. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Anggraeni, D.; Sanjaya, W.S.; Nurasyidiek, M.Y.; Munawwaroh, M. The Implementation of Speech Recognition using Mel-Frequency Cepstrum Coefficients (MFCC) and Support Vector Machine (SVM) Method Based on Python to Control Robot Arm. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *288*, 012042. [\[CrossRef\]](#)
10. Dhanjal, A.S.; Singh, W. *A Comprehensive Survey on Automatic Speech Recognition Using Neural Networks*; Springer Nature: Berlin/Heidelberg, Germany, 2023.
11. Smith, S.W. *The Scientist and Engineer's Guide to Digital Signal Processing*, 2nd ed.; California Technical Publishing: San Diego, CA, USA, 1999; pp. 557–558.
12. Bassey, J.; Li, X.; Qian, L. A Survey of Complex-Valued Neural Networks. *arXiv* **2021**, arXiv:2101.12249v1.
13. MYONeuralNet. Complex-Valued Neural Networks—Experiments. 2016. Available online: <http://makeyourownneuralnetwork.blogspot.com/2016/05/complex-valued-neural-networks.html> (accessed on 19 February 2021).

14. Fischer, R. Wirtinger Calculus. In *Precoding and Signal Shaping for Digital Transmission*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2002; pp. 405–413.
15. PyTorch. Automatic Differentiation Package—torch.autograd, n.d. Available online: <https://pytorch.org/docs/stable/notes/autograd.html> (accessed on 3 November 2023).
16. Lee, C.; Hasegawa, H.; Gao, S. Complex-Valued Neural Networks: A Comprehensive Survey. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 1433–1454. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.