IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 5, 2019, accepted December 3, 2019, date of publication December 5, 2019,
date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2957863

Parameter Estimation Processor for Chirp Signals Based on a Complex-Valued Deep Neural Network

HANNING SU[1], QINGLONG BAO[1], AND ZENGPING CHEN[2]
[1]College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China
[2]College of Electronics and Communication Engineering, Sun Yat-sen University, Guangzhou 510000, China

Corresponding author: Qinglong Bao (cbpest@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61401489.

ABSTRACT This paper addresses the problem of estimating the parameters of constant-amplitude chirp signals that have single or multiple components and are embedded in noise. Chirp signals are widely employed in applications such as radar and telecommunications, and it is a key task in countermeasure techniques to estimate their parameters without prior information. Hence, a parameter estimation processor based on a complex-valued deep neural network (CV DNN) is proposed to perform this task efficiently. The CV DNN, which is designed for regression, consists of a function fitter and a predictor. The function fitter acts like a eigenfunction mapping: it maps the one-dimensional input into a two-dimensional feature map suitable for subsequent network learning. As a special feature extraction tool, the predictor extracts local features from the feature map and estimates parameters. Simulation results indicate that the CV DNN outperforms conventional processors. Moreover, it is more accurate than the Wigner–Hough transform while being several orders of magnitude faster, which will enable real-time signal processing with fewer computational resources. Furthermore, we demonstrate that the CV DNN shows strong robustness to changes in modulation parameters and the number of components of a chirp signal. This study shows the advantages of deep learning systems for signal parameter estimation.

INDEX TERMS Chirp signals, complex valued deep neural network, deep learning, parameter estimation, sensitivity analysis, time–frequency analysis.

I. INTRODUCTION

Linear frequency-modulated signals, which are also called chirp signals, are widely employed in various applications such as radar [1], sonar [2], ultrasonics [3], and telecommunications [4]. Accurate estimation of the parameters of chirp signals, i.e., initial frequency f_0 and chirp rate μ, without any prior information is essential in electronic countermeasure technology. An example is in the area of source localization [5], where the estimated parameters can be used to construct a matched filter to match signals received at several locations to determine the time-difference-of-arrival of a target.

The main trends in research on chirp signal parameter estimation are improving estimation accuracy, increasing computational efficiency, and enhancing adaptation to low signal-to-noise ratios (SNRs).

For a long time, various methods based on maximum likelihood estimator (MLE) [6], [7] are the predominant solutions to this task, and they are reported to be asymptotically optimal. Most of these methods can be ascribed to a multivariable optimization algorithm, which requires a two-dimensional extremum search in practical application. Therefore, heavy computational complexity is generally needed for high estimation accuracy [8], which does not meet the requirements of real-time signal processing. The algorithms provided in refs. [9], [10] are much more efficient in computation, but not applicable to multi-component chirp signal.

A. TIME FREQUENCY ANALYSIS PROCESSORS

Time-frequency-based methods have been reported to be effective for detecting and estimating chirp signals. These techniques have attracted considerable attention and proved

The associate editor coordinating the review of this manuscript and approving it for publication was Zihuai Lin.

themselves to be effective [11]–[20]. The time–frequency transform (TFT), which is a method that extracts the characteristics of time–frequency joint distributions, is performed by transforming a chirp signal from a one-dimensional (1-d) time domain into a 2-d time–frequency domain through some linear or nonlinear transformations. TFTs are usually combined with some straight line detection (SLD) methods, e.g., the Hough [21] or Radon [22] transforms, which are a class of methods widely used in image processing for shape detection and feature extraction. Because chirp signals can be described as straight lines in the time–frequency domain, SLD, which integrates along all potential lines, is used to convert the task of tracking straight lines in the time–frequency domain into locating the maximum peak in the parameter (f_0 and μ) domain. This parameter estimation technique can be considered as a class of conventional processors whose input consists of chirp signals and output is the estimated parameters, as shown in Fig. 1.

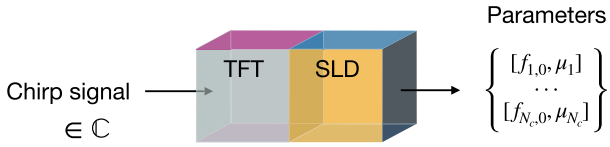


FIGURE 1. Conventional chirp-signal parameter estimator.

Therefore, for chirp signal estimation in conventional processors, a proper TFT is needed to obtain an appropriate line representation in the time–frequency domain. Early TFT methods employed the spectrogram, which is the square of the magnitude of the short-time Fourier transform (STFT) [11]. However, this technique suffers from a fixed time–frequency resolution due to the fixed window length used in the analysis, which limits its applications in practice. In [12] and [13], STFT is optimized from the perspective of a wavelet base and polynomial phase decomposition, respectively, and the derivative transformation of the STFT is obtained. Although the time–frequency resolution is improved, it is at the cost of computational efficiency. In contrast, the Wigner–Ville distribution (WVD) [14] is optimal in the sense of maximum energy concentration about the instantaneous frequency [23]. However, a bilinear transformation such as WVD generates redundant information such as cross terms for multi-component chirp signals. Although the Wigner–Hough transform (WHT) [24] can decrease the undesirable cross terms in the parameter domain, the integral operation on the cross terms also greatly increases the calculation cost. In addition, a series of Huang transforms, such as the Teager–Huang [15] and Hilbert–Huang [16] transforms, are noise-sensitive, and when they will be applied in a noisy environment, noise reduction processing should be carried out in advance [25].

It makes sense that chirp signals exhibit lines on the TFT image given that the TFT, which is an eigenfunction mapping, maps chirp signal from the 1-d time domain to the 2-d time–frequency domain to highlight the joint features.

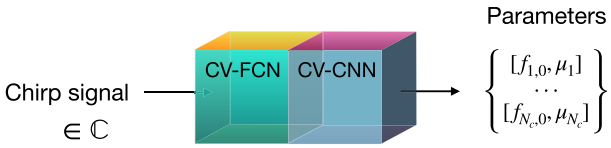


FIGURE 2. Proposed deep-learning-based chirp-signal parameter estimator.

Then, an SLD method, which is a special feature extraction tool, can be used to extract parameter features on such images. A common feature of the above-mentioned processors is that they are parametric and create forward mappings from signal sequences to parameter outputs. Each TFT has its individual advantages and disadvantages, and different matching criteria result in different processors.

B. PROPOSED PROCESSOR

The above analysis shows that TFT has its limitations as a eigenfunction mapping. An alternative could be provided by a deep neural network (DNN) [26]–[28], which is a machine learning-based method. The most striking fact about neural networks is that they are capable of fitting any function [29]. Therefore, it is theoretically feasible to replace a TFT with a neural network. Further, convolutional neural networks (CNNs) have been attracting attention due to their superior feature extraction capability [30]. Hence, for features that cannot be defined intuitively but are closely related to the parameters to be estimated, a CNN becomes a powerful tool for extracting them.

Although DNNs are generally used for classification [31]–[36], they have recently been used in regression problems, such as direction-of-arrival estimation [37] and the estimation of gravitational wave parameters [38], where DNNs are able to interpolate between waveform templates in a similar manner to that of Gaussian process regression $(GPR)^2$ [39]. Therefore, it makes sense to apply DNNs to parameter estimation problems.

In this paper, we propose a parameter estimation processor based on a DNN for chirp signals. The noisy time-series data is directly input to the DNN, and a fully connected neural network (FCN) is introduced as a function fitter to map the 1-d input to a 2-d feature map suitable for subsequent network learning. After that, a convolutional neural network (CNN), which is a special feature extraction tool for the feature map generated by the FCN, is used as a predictor to estimate parameters f_0 and μ. In contrast to the processor that combines TFT and SLD, in the processor based on DNNs, the TFT is replaced by an FCN and the SLD is replaced by a CNN.

Further, a chirp signal is a type of non-stationary signal that contains not only the amplitude but also the phase information, which is hidden in the real and imaginary parts. It has been shown that the phase information in speech signals affects their intelligibility [40]. Thus, to make the network more robust, it makes sense to retain as much useful

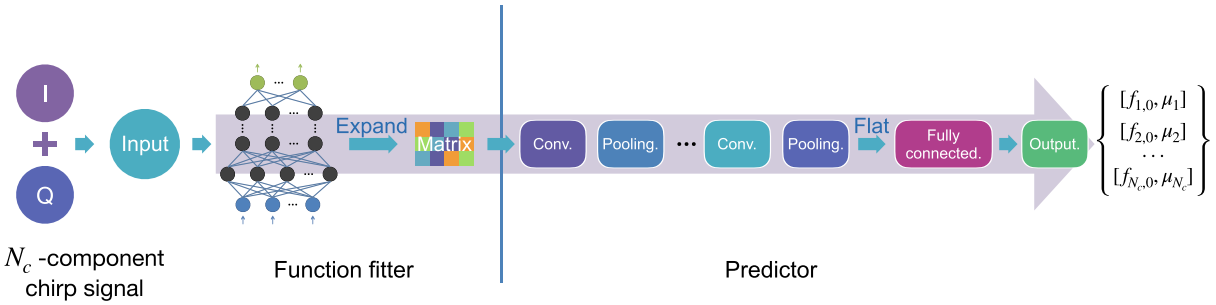


FIGURE 3. Framework of an N_c-component CV DNN, where *I* denotes the in-phase, i.e., real part of the signal, and *Q* is the quadrature, i.e., imaginary part of the signal.

information as possible from the original signal. Therefore, a noisy complex-valued time-series data is input to the DNN directly, and the complex-valued form of a DNN, the CV DNN [41]–[44], is introduced. Accordingly, in the proposed processor, the FCN is modified into a complex-valued FCN (CV FCN), and the CNN is modified into a CV CNN, as shown in Fig. 2.

To the best of our knowledge, this is the first time that a DNN has been utilized for the parameter estimation of chirp signals.

The rest of the paper consists of five parts. Section II introduces the framework of the CV DNN and details the forward propagation and backpropagation. Section III presents the details of CV DNN training. Section IV presents the results of simulations to evaluate the performance of the proposed processor with respect to that of the conventional one. Section V concludes the paper.

II. CV DNN FOR PARAMETER ESTIMATION

In this section, we present the design of the CV DNN framework for parameter estimation.

A. OVERALL FRAMEWORK

The overall framework of the proposed processor based on CV DNN is shown in Fig. 3, which is divided into two parts: the function fitter and the predictor. The two parts are composed of a CV FCN and a CV CNN, respectively.

The input of the whole framework is a complex-valued N_c-component ($N_c \geq 1$) chirp signal, which is a 1-d time-series data of length N. In the function fitter, the CV DNN denoises the input and acts as a eigenfunction mapping to fit a function. Then, the output is reshaped into a complex-valued 2-d matrix, which can be regarded as a feature map, for input to the predictor. In the predictor, a CV CNN, which acts as a special feature extraction tool, extracts features related to the parameters from the complex-valued matrix. After several convolution and pooling operations, all local features are finally combined by a fully connected layer to output the estimated parameters. The output of the whole framework is several sets of estimated parameters, $\{[f_{1,0}, \mu_1], \ldots, [f_{N_c,0}, \mu_{N_c}]\}$, i.e., neurons in the output layer are divided into two categories, which are output initial frequency f_0 and chirp rate μ. A set of neurons outputting $[f_{j,0}, \mu_j], j \in \{1, \ldots, N_c\}$ can be regarded as a channel that outputs the parameters for the jth component of a chirp signal according to the specified order.

Note that the network structure designed in this paper has a one-to-one correspondence with the number of signal components N_c, i.e., signals with different numbers of components need to be trained by networks with different structures, respectively. For instance, the N_c-component CV DNN framework shown in Fig. 3 can only be applied to N_c-component chirp signals. The only difference between these networks is the number of neurons 2 N_c of the output layer.

B. FUNCTION FITTER

The universality theorem [29] proves that artificial neural networks with just three layers (one hidden layer) can model any function up to any desired level of accuracy. Thus, it makes sense to use neural networks instead of TFTs to fit a function that reflects the parameter features of chirp signals. In this paper, the function fitter consists of a 3-layer CV FCN.

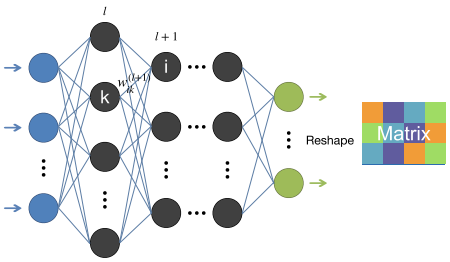


FIGURE 4. Function fitter.

Generally, a CV FCN has an input layer, one or more hidden layers, and an output layer, as shown in Fig. 4. A nonlinear activation function is applied to the output of each hidden layer. Without this nonlinearity, the use of multiple layers would become redundant, as the network would only be able to express linear combinations of the input. The most commonly used nonlinear activation functions are the logistic sigmoid, hyperbolic tan, and rectified linear unit (called ReLU) [45]. It has been empirically observed that the ReLU produces the best results for most applications [46].

This function is mathematically expressed as $\max(x, 0)$. In the CV DNN, we extend ReLU to its complex form (cReLU) [41] as follows:

$$f(\cdot) = \text{cReLU}(x) = \max(\Re(x), 0) + \text{j}\max(\Im(x), 0), \quad (1)$$

where $\Re$ and $\Im$ denote the real and imaginary parts of a complex value.

In the $l+1$th fully connected layer, the complex output $O_i^{(l+1)} \in \mathbb{C}^I$ is computed by the multiplication between all the previous layer's outputs $O_k^{(l)} \in \mathbb{C}^K$, and a bank of filters $\omega_{ik}^{(l+1)} \in \mathbb{C}^{K\times I}$. Then, a bias $b_i^{(l+1)} \in \mathbb{C}^I$ is added and the output $O_i^{(l+1)} \in \mathbb{C}^I$ can be obtained from the activated $V_i^{(l+1)}$, where $\mathbb{C}$ denotes the complex domain and the superscript is its dimension. Further, K and I respectively represent the number of neurons in layers l and $l+1$. The output can be written as

$$O_i^{(l+1)} = f\left(V_i^{(l+1)}\right), \quad (2)$$

where

$$V_i^{(l+1)} = \sum_{k=1}^{K} \omega_{ik}^{(l+1)} \cdot O_k^{(l)} + b_i^{(l+1)}. \quad (3)$$

C. PREDICTOR

The predictor consists of a 5-layer CV CNN that contains three convolutional layers, a fully connected layer, and an output layer.

1) BASIC NEURON STRUCTURE

The basic mechanism of a single neuron structure [42] in the convolutional layers of a CV CNN is illustrated in Fig. 5.

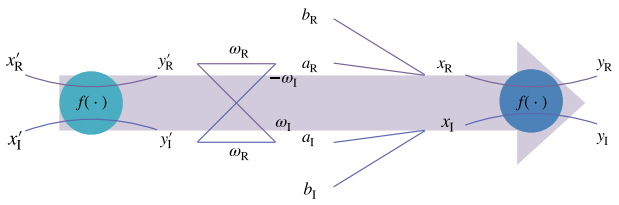


FIGURE 5. Basic neuron structure and connections of the convolutional layers in a CV CNN.

Here, x' and x represent the input of the previous neuron and the next neuron, respectively. Similarly, y' and y represent their output, ω represents the weight between neurons, a represents the results of the input sent to the next layer through the weight, and b represents the bias, where x', x, y', y, ω a, $b \in \mathbb{C}$. Subscripts R and I represent real and imaginary components, respectively.

Note that the cReLU is used in this paper and is obtained as follows:

$$f(\cdot) = \text{cReLU}\left(x'\right) = \max\left(x'_{\text{R}}, 0\right) + \text{j}\max\left(x'_I, 0\right). \quad (4)$$

We can then obtain

$$\partial y'_{\text{R}}/\partial x'_{\text{R}} = \begin{cases} 1, x'_{\text{R}} > 0 \\ 0, x'_{\text{R}} \le 0 \end{cases}, \quad \partial y'_I/\partial x'_{\text{I}} = \begin{cases} 1, x'_I > 0 \\ 0, x'_I \le 0 \end{cases} \quad (5)$$

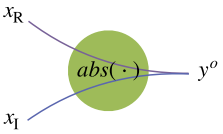


FIGURE 6. Basic neuron structure of the output layer in the CV-CNN.

In this paper, the expected output of the overall framework is the real value of the estimated parameters; therefore, the output layer's neurons need to be activated by the absolute value function $abs(\cdot)$, as follows (Fig. 6):

$$y^o = |x_{\text{R}} + \text{j}x_{\text{I}}| = \sqrt{(x_{\text{R}})^2 + (x_{\text{I}})^2}. \quad (6)$$

2) COMPLEX-VALUED CNN

Each hidden unit, which is complex valued, in the convolutional layers is connected to local patches in the feature maps, also complex-valued, of the previous layer through a set of complex weight matrices known as kernels or filter banks. The units in a local patch are convolved by the weight matrix and then passed through the nonlinear activation function. Convolution means that the units in a feature map are forced to share the same filter bank. Different feature maps in a layer use different filter banks, which correspond to different channels. Each filter bank detects specific regional features from the input matrices, so each feature map represents a unique feature at different positions in the previous layer.

1) In the convolutional layer, the complex-valued output feature maps $O_i^{(l+1)} \in \mathbb{C}^{W_2\times H_2\times I}$ are computed by the convolution between all the previous layer's output feature maps $O_k^{(l)} \in \mathbb{C}^{W_1\times H_1\times K}$ and a bank of filters $\omega_{ik}^{(l+1)} \in \mathbb{C}^{F\times F\times K\times I}$. Then, a bias $b_i^{(l+1)} \in \mathbb{C}^I$ is added, where $\mathbb{C}$ denotes the complex domain and the superscript is its dimension. This convolution is calculated by

$$\begin{aligned} O_i^{(l+1)} &= f\left(V_i^{(l+1)}\right) \\ &= max\left(\Re\left(V_i^{(l+1)}\right), 0\right) + jmax\left(\Im(V_i^{(l+1)}), 0\right), \quad (7) \\ V_i^{(l+1)} &= \sum_{k=1}^{K} \omega_{ik}^{(l+1)} * O_k^{(l)} + b_i^{(l+1)} \\ &= \sum_{k=1}^{K} \left(\Re\left(\omega_{ik}^{(l+1)}\right) \cdot \Re\left(O_k^{(l)}\right) - \Im(\omega_{ik}^{(l+1)}) \cdot \Im(O_k^{(l)})\right) \\ &\quad + j\sum_{k=1}^{K} \left(\Re\left(\omega_{ik}^{(l+1)}\right) \cdot \Im\left(O_k^{(l)}\right)\right. \\ &\quad \left. + \Im\left(\omega_{ik}^{(l+1)}\right) \cdot \Re\left(O_k^{(l)}\right)\right) + b_i^{(l+1)}, \quad (8) \end{aligned}$$

where $j = \sqrt{-1}$ is the imaginary unit. Symbol "$*$" denotes the convolution operation, $O_k^{(l)}$ is the unit of the kth output feature map in layer l, and $V_i^{(l+1)}$ denotes the weighted sum of inputs to the ith input feature map in layer $l+1$. Finally, $f(\cdot)$ denotes cReLU.

2) The pooling layer is used to merge semantically similar features that are detected by the convolutional layer. The pooling function calculates a summary statistic over a local patch independently for each feature map in the convolutional layer. Therefore, the numbers of feature maps in the pooling and convolutional layers are equal. The maximum and average of a rectangular neighborhood are the two most commonly used pooling functions.

A straightforward extension of average pooling from real to complex can be defined as

$$O_i^{(l+1)}(x,y) = \underset{u,v=0,\ldots,g-1}{\text{ave}} O_i^{(l)}(x\cdot s+u, y\cdot s+v), \quad (9)$$

where g denotes the pooling size, s is the stride, and $O_i^{(l+1)}(x,y)$ is the unit of the ith input feature map at position (x,y).

3) In the top layers, usually one or more fully connected layers are used in our CV CNN. That is, each neuron in the fully connected layers is connected to all neurons in the preceding layer, which can be seen as a special case of a convolutional layer. The output is written the same as it is in (2) and (3), while $f(\cdot)$ is associated with $abs(\cdot)$.

D. COMPLEX-VALUED BACKPROPAGATION

Supervised training of a DNN optimizes the weights/biases so that the network output matches the desired output or the given labels of the training data. After the multiple feature extraction stages, there still exists an error between the output and the target. The error can be described as a loss function E. Therefore, the network parameters are trained by stochastic gradient descent and the loss function is minimized during backpropagation [47]. Stochastic gradient descent refers to the process of estimating the gradient using only a subset of the training samples, i.e., an N_b-sample minibatch, where N_b is the batch size. Therefore, the estimated gradient is actually sampled from a small number of cases, which is in fact a stochastic sampling process. By computing the error gradient with respect to the parameters ($\partial E/\partial w$), the updating rule is $\omega \leftarrow \omega - \eta(\partial E/\partial \omega)$, where η is the learning rate. The training sample of each iteration can be expressed as $\{X(n), T(n)\}_{n=1}^{N_b}$, where $X(n)$ and $T(n)$ denote the nth input data and the label, respectively.

Considering that parameter estimation is a kind of regression-type problem that is different from the classification problem, the loss function, which can directly reflect the difference in value between output and label, should be selected. Mean squared error, root mean squared error (RMSE), and mean absolute deviation are all options for the loss functions for solving regression-type problems. Furthermore, considering that the two parameters f_0 and μ to be estimated in this paper have different lower bounds for their estimated errors, an equal weight is clearly not appropriate. Hence, a weighted RMSE (WRMSE) based loss function is introduced to the network in the proposed system and is expressed as follows:

$$\begin{aligned} E &= \frac{1}{N_b}\cdot\sum_{n=1}^{N_b}\Big[\varepsilon_{f_0}\cdot RMSE\left(T^{[f_0]}(n), O^{[f_0]}(n)\right) \\ &\quad + \varepsilon_\mu \cdot RMSE\left(T^{[\mu]}(n), O^{[\mu]}(n)\right)\Big] \\ &= \frac{1}{N_b}\cdot\sum_{n=1}^{N_b}\left[\frac{\varepsilon_{f_0}}{N_c}\cdot\left(\sum_{k=1}^{N_c}\left|T_k^{[f_0]}(n)-O_k^{[f_0]}(n)\right|^2\right)^{\frac{1}{2}}\right. \\ &\quad \left. + \frac{\varepsilon_\mu}{N_c}\cdot\left(\sum_{k=1}^{N_c}\left|T_k^{[\mu]}(n)-O_k^{[\mu]}(n)\right|^2\right)^{\frac{1}{2}}\right], \end{aligned} \quad (10)$$

where $T_k^{[f_0]}(n)$ and $T_k^{[\mu]}(n)$ respectively mean the f_0 and μ (label) of the kth component of the nth data, whereas $O_k^{[f_0]}(n)$ and $O_k^{[\mu]}(n)$ are those of the output. The loss is the weighted sum of the errors estimated by the two parameters $\varepsilon_{f_0}+\varepsilon_\mu = 1$ and ε_{f_0}. In addition, $\varepsilon \in \mathbb{R}$ are weight coefficients, which are related to the normalized minimum of the RMSE. Note that ε_{f_0} and ε_μ are defined in Section III-C.2.

The minimum of the above loss function is searched for by iteratively adjusting the weights according to

$$\begin{aligned} w_{ik}^{(l+1)}[t+1] &= w_{ik}^{(l+1)}[t] + \Delta w_{ik}^{(l+1)}[t] \\ &= w_{ik}^{(l+1)}[t] - \eta\frac{\partial E[t]}{\partial w_{ik}^{(l+1)}[t]}, \\ b_i^{(l+1)}[t+1] &= b_i^{(l+1)}[t] + \Delta b_i^{(l+1)}[t] \\ &= b_i^{(l+1)}[t] - \eta\frac{\partial E[t]}{\partial b_i^{(l+1)}[t]}. \end{aligned} \quad (11)$$

The derivatives of complex functions are obtained according to the complex chain rule [48]. The key process is computing the error gradient of the weights as

$$\begin{aligned} &\frac{\partial E}{\partial w_{ik}^{(l+1)}} \\ &= \frac{\partial E}{\partial \Re\left(w_{ik}^{(l+1)}\right)} + j\frac{\partial E}{\partial \Im\left(w_{ik}^{(l+1)}\right)} \\ &= \left(\frac{\partial E}{\partial \Re\left(V_i^{(l+1)}\right)}\frac{\partial \Re\left(V_i^{(l+1)}\right)}{\partial \Re\left(w_i^{(l+1)}\right)} + \frac{\partial E}{\partial \Im\left(V_i^{(l+1)}\right)}\frac{\partial \Im\left(V_i^{(l+1)}\right)}{\partial \Re\left(w_i^{(l+1)}\right)}\right) \\ &\quad + j\left(\frac{\partial E}{\partial \Re\left(V_i^{(l+1)}\right)}\frac{\partial \Re\left(V_i^{(l+1)}\right)}{\partial \Im\left(w_{ik}^{(l+1)}\right)}\right. \\ &\quad \left. + \frac{\partial E}{\partial \Im\left(V_i^{(l+1)}\right)}\frac{\partial \Im\left(V_i^{(l+1)}\right)}{\partial \Im\left(w_{ik}^{(l+1)}\right)}\right). \end{aligned} \quad (12)$$

To simplify the expression, an intermediate quantity called the error term is defined as

$$\delta_i^{(l+1)} = -\frac{\partial E}{\partial \Re\left(V_j^{(l+1)}\right)} - j\frac{\partial E}{\partial \Im\left(V_i^{(l+1)}\right)}, \quad \delta_i^{(l+1)} \in \mathbb{C}. \quad (13)$$

With (7), (8), and (13), (12) can be simplified as

$$\frac{\partial E}{\partial w_{ik}^{(l+1)}} = -\delta_i^{(l+1)}\overline{O_i^{(l)}}, \tag{14}$$

where $\overline{(\cdot)}$ denotes the complex conjugate.

The specific derivation of error terms in each layer is given in [43], and several final generalized formulae for parameter updating are obtained.

Neurons in the fully connected layers are formed in a 1-d array, and their parameters are updated by

$$w_{ik}^{(l+1)}[t+1] = w_{ik}^{(l+1)}[t] + \eta\delta_i^{(l+1)}\overline{O_i^{(l)}}, \tag{15}$$

$$b_i^{(l+1)}[t+1] = b_i^{(l+1)}[t] + \eta\delta_i^{(l+1)}. \tag{16}$$

In other layers, neurons are formed in a 2-d array as

$$\begin{aligned} &w_{ik}^{(l+1)}[t+1] \\ &\quad = w_{ik}^{(l+1)}[t] + \eta\delta_k^{(l+1)} * \overline{O_k^{(l-1)}} \\ &\quad = w_{ik}^{(l+1)}[t] + \eta\sum_{x,y}\delta_k^{(l)}(x,y)\cdot\overline{O_k^{(l-1)}}(x-u,y-v), \end{aligned} \tag{17}$$

$$\begin{aligned} &b_i^{(l+1)}[t+1] \\ &\quad = b_i^{(l+1)}[t] + \eta\delta_i^{(l+1)} = b_i^{(l+1)}[t] + \eta\sum_{x,y}\delta_k^{(l+1)}(x,y). \end{aligned} \tag{18}$$

The parameters are adjusted iteratively until the error between the target and the network output is small enough to meet the requirement.

III. APPLICATION TO THE PARAMETER ESTIMATION OF CHIRP SIGNALS

In this section, we present the detailed implementation of the proposed CV DNN for estimating the parameters of chirp signals. First, some assumptions are made; next, the data set preparation is described, followed by the detailed configuration of the CV DNN. Finally, the training strategy of the network is introduced.

A. ASSUMPTIONS

The study in this paper is based on the following assumptions:

1) The sampling point of a time-series data is fixed $N = 256$ and the number of components N_c is known.
2) The pulse width T of a chirp signal is equal to the length of the observation time.
3) There is only the chirp signal and additive white Gaussian noise in the channel without other interference or noise.

B. TRAINING SET AND TEST SET

For the datasets for an N_c-component CV DNN, the data are vectors of size $1 \times N$ generated by the computer. They consist of N_c-component chirp signals with the same number of components and different modulation parameters. The N_c-component simulated chirp signals in the data all have the following form:

$$z_{N_c}(t) = \sum_{i=1}^{N_c} s_i(t) + n(t), \tag{19}$$

$$s_i(t) = e^{2\pi j\left(f_{i,0}t + \frac{1}{2}\mu_i t^2,\right)} \tag{20}$$

where $n(t)$ is additive white Gaussian noise, sampling frequency $F_s = 1.5\ MHz$, and length $N = 256$.

For the training data, chirp rate μ is varied from 0.04 to 4.00 GHz/s in steps of $p_1 = 0.04\ GHz/s$, and the initial frequency f_0 is varied from 7.5 KHz to 750 KHz in steps of $p_2 = 7.5\ KHz$. The testing data have intermediate parameter values, i.e., parameters separated from values in the training data by $0.5 \cdot p$. By not having overlapping values in the training and testing data, one can ensure that overfitting can be avoided to a certain extent, i.e., the network memorizes only the inputs shown to it without learning to generalize to new inputs. The distribution of the parameters and 5,050 templates for generating the training and testing data, are shown in Fig. 7.

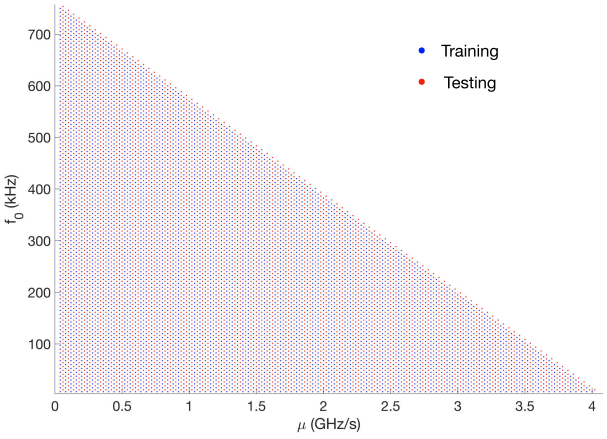


FIGURE 7. Distribution of the data.

Subsequently, different SNRs are superimposed on top of the signals, thus augmenting the size of the data sets. The SNR was adjusted according to the aims of each training session. As usual, the inputs are normalized to make the training process easier.

The final training sets at each given SNR^i consist of 1,500,000 data, where each instance of training data is produced from 5,050 templates by adding noise within the range of $\left(SNR^i - 5\ dB, SNR^i + 5\ dB\right]$, $i = 1, \ldots, 13$. It is also standard practice to use a validation set to monitor the performance of the network on unseen data during training to prevent overfitting. The validation and testing sets at SNR^i, which consist of 500,000 data each, were generated from a different set of 5,050 templates by superimposing random noise.

Because of the randomness of parameter selection, for multi-component chirp signals, it will be difficult for the loss

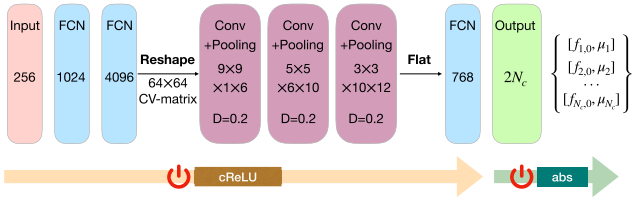


FIGURE 8. Node details of an N_c-component CV DNN.

function to converge if output channels of the parameters of each chirp component are not specified in order. Therefore, during the preparation phase, the parameter labels of the data sets need to be arranged in the order specified. Therefore, for each element $(u_p, f_{0,q})$, $p, q = 1, ...100$ in the evenly distributed parameter templates, we encode each element using the following rule:

$$order = \sum_{i=0}^{p+q} i + q, \tag{21}$$

Then, we can obtain

$$\begin{array}{llll} order & p, q, & \sum_{i=0}^{p+q} i, & q \\ 1 & 1, 0, & 1, & 0 \\ 2 & 0, 1, & 1, & 1 \\ 3 & 2, 0, & 3, & 0 \\ 4 & 1, 1, & 3, & 1 \\ 5 & 0, 2, & 3, & 2 \\ \cdots & \cdots & \cdots & \cdots \end{array} . \tag{22}$$

With the help of this constraint, parameter labels in the data sets are arranged in order from maximum to minimum, so that neurons outputting μ and f_0 are aligned, and the channels for outputting each component's parameters are formed.

C. NETWORK STRUCTURE AND IMPLEMENTATION

This section first presents the process of building the whole network structure; then, ε_{f_0} and ε_μ in the loss function are defined. Finally, the training strategy of the network is introduced in detail.

1) NETWORK STRUCTURE

Each instance of input training data is a sequence of size 1×256. Accordingly, the input layer contains 256 neurons. After the data flow passes through the filters of the fully connected layers, the 1-d vector is reshaped into a complex-valued matrix with size of $64 \times 64 \times 1$, which means the size of a local patch is 64×64 and the number of channels is 1. The complex-valued matrix can be regarded as a feature map fitted by the CV FCN based on the characteristics of the training examples, which then is input to the CV CNN. The input complex-valued matrix is filtered by a bank of six convolution filters of size $9 \times 9 \times 1$ with a stride of one in the first convolutional layer.

Meanwhile, to retain more edge information, each convolution operation is accompanied by a corresponding padding, resulting in six feature maps of size 64×64. This is followed by an average pooling layer, with a pooling size of 2×2 and stride of one: its size then becomes 32×32. The filter size of the second convolutional layer is $5 \times 5 \times 6 \times 10$, producing 10 feature maps of size 32×32. The output of the same pooling layer is feature maps of size $16 \times 16 \times 10$. Similarly, the filter size of the third convolutional layer is $3 \times 3 \times 10 \times 12$. After convolution and pooling, its output is 12 feature maps of size 8×8. Then, the 3-d feature maps are reshaped into a 1-d vector containing 768 neurons for the fully connected layer. This is followed by the output layer containing $2N_c$ neurons, whose output is activated by $abs(\cdot)$. Here, $2N_c$ is equal to the number of parameters and N_c is the number of signal components.

2) LOSS FUNCTION

As mentioned in Section II-*D*, because the estimated error of f_0 and μ have different lower bounds, we introduce the following WRMSE as the loss function:

$$\begin{aligned} E &= \frac{1}{N_b} \cdot \sum_{n=1}^{N_b} \left[\varepsilon_{f_0} \cdot RMSE\left(T^{[f_0]}(n), O^{[f_0]}(n)\right) \right. \\ &\quad \left. + \varepsilon_\mu \cdot RMSE\left(T^{[\mu]}(n), O^{[\mu]}(n)\right) \right] \\ &= \frac{1}{N_b} \cdot \sum_{n=1}^{N_b} \left[\frac{\varepsilon_{f_0}}{N_c} \cdot \left(\sum_{k=1}^{N_c} \left| T_k^{[f_0]}(n) - O_k^{[f_0]}(n) \right|^2 \right)^{\frac{1}{2}} \right. \\ &\quad \left. + \frac{\varepsilon_\mu}{N_c} \cdot \left(\sum_{k=1}^{N_c} \left| T_k^{[\mu]}(n) - O_k^{[\mu]}(n) \right|^2 \right)^{\frac{1}{2}} \right], \end{aligned} \tag{23}$$

where $\varepsilon_{f_0} + \varepsilon_\mu = 1$, ε_{f_0}, and $\varepsilon_\mu \in \mathbb{R}$ are weight coefficients that are related to the normalized minimum of the RMSE. One optional reference is the Cramér–Rao lower bound (CRLB) [10], which represents the best estimation accuracy that can be obtained in an unbiased estimation, i.e., the minimum estimated error of the mean squared error. Thus, $CRLB^{\frac{1}{2}}$ for RMSE.

$$CRLB^{\frac{1}{2}}(f_0) \approx \sqrt{\frac{96}{4\pi^2 N^3 \Delta^2 10^{\frac{SNR}{10}}}} = \sqrt{\frac{96}{4\pi^2 N T^2 10^{\frac{SNR}{10}}}} \tag{24}$$

$$CRLB^{\frac{1}{2}}(\mu) \approx \sqrt{\frac{90}{\pi^2 N^5 \Delta^4 10^{\frac{SNR}{10}}}} = \sqrt{\frac{90}{\pi^2 N T^4 10^{\frac{SNR}{10}}}} \tag{25}$$

The bounds $CRLB^{\frac{1}{2}}(f_0)$ and $CRLB^{\frac{1}{2}}(\mu)$ are different in this paper; for instance, when $SNR = 10\ dB$, $CRLB^{\frac{1}{2}}(f_0) = 180.5872$ and $CRLB^{\frac{1}{2}}(\mu) = 2.0491 \times 10^6$. Further, because of the difference in the normalized dimensions, the RMSE of f_0 and μ converge at different bounds. Data sets are normalized according to maxmin criterion as follows:

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}, \tag{26}$$

$$\hat{X}_{\text{norm}} = \frac{\hat{X} - X_{\min}}{X_{\max} - X_{\min}}, \tag{27}$$

where $X \in \{f_0, \mu\}, \hat{X} \in \left\{\hat{f}_0, \hat{\mu}\right\}$. Then,

$$RMSE\left(T_{X_{norm}}, O_{X_{norm}}\right) = \left(\frac{\sum_{n=1}^{N_c}\left|T_{X_{norm}}(n) - O_{X_{norm}}(n)\right|^2}{N_c}\right)^{\frac{1}{2}}$$
$$= \left(\frac{\sum_{n=1}^{N_c}\left|X_{norm} - \hat{X}_{norm}\right|^2}{N_c}\right)^{\frac{1}{2}}$$
$$= \left(\frac{\sum_{n=1}^{N_c}\left|X - \hat{X}\right|^2}{N_c \cdot |X_{max} - X_{min}|^2}\right)^{\frac{1}{2}}$$
$$= \frac{RMSE\left(T_X, O_X\right)}{|X_{max} - X_{min}|}. \tag{28}$$

Therefore, we can obtain the normalized form of $CRLB^{\frac{1}{2}}$. Again, we take $SNR = 10\ dB$. This yields

$$CRLB^{\frac{1}{2}}(f_0)_{norm} = \frac{CRLB^{\frac{1}{2}}(f_0)}{|max(f_0) - min(f_0)|}$$
$$[-1pt] = \frac{180.5872}{\left|7.5 \times 10^5 - 7.5 \times 10^3\right|}$$
$$[-1pt] = 2.4322 \times 10^{-4}, \tag{29}$$

$$[-1pt]CRLB^{\frac{1}{2}}(\mu)_{norm} = \frac{CRLB^{\frac{1}{2}}(\mu)}{|max(\mu) - min(\mu)|}$$
$$[-1pt] = \frac{2.0491 \times 10^6}{\left|4 \times 10^9 - 4 \times 10^7\right|}$$
$$[-1pt] = 5.1745 \times 10^{-4}, \tag{30}$$

and we can obtain

$$\frac{CRLB(f_0)_{norm}}{CRLB(\mu)_{norm}} = \frac{0.2837}{0.7163}, \tag{31}$$

which means that at the same loss, μ converges more easily than f_0. Moreover, it can be found from (24) and (25) that this ratio is independent of the SNR. Therefore, it is an effective method to weight the RMSEs of the parameters according to this ratio so that the RMSEs of both can converge to their minimum values synchronously. The weights are hence

$$\varepsilon_{f_0} : \varepsilon_\mu = 0.7163 : 0.2837 \quad \left(\varepsilon_{f_0} + \varepsilon_\mu = 1, \varepsilon_{f_0}, \varepsilon_\mu \in \mathbb{R}\right). \tag{32}$$

D. TRAINING STRATEGY

The hyperparameters were optimized by trial and error to obtain the architectures of the DNNs that achieve the best performance in terms of speed and accuracy. First, we used a mono-component chirp signal without Gaussian white noise to determine the optimal architectures of the mono-component CV DNN. The structure of the hidden layers in the mono-component CV DNN was also found to be optimal for multi-component CV DNNs. The CV DNNs were designed and trained using Python's TensorFlow.

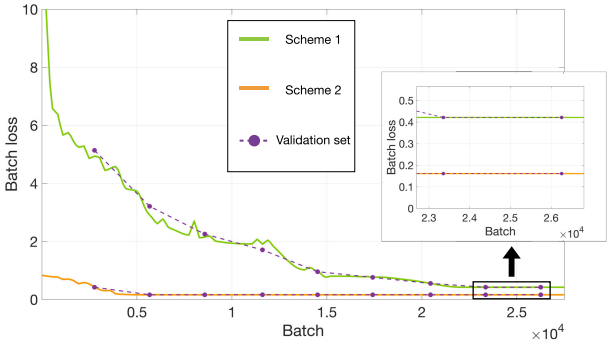


FIGURE 9. Training process of mono-component CV DNN under $SNR^{10} = 10\ dB$, where batchsize = 512, number of iterations = 2,930, and rounds = 80. The solid line represents the batch loss of the training set, $loss = 512 \cdot E$, whereas the dots represent the batch loss of the validation set.

A new strategy was devised to reduce the training time of the CV DNNs while ensuring optimal performance. In this strategy, training is started by training the CV DNN on inputs with high SNR^1 (100 dB) and then gradually decreasing the SNR^i in steps of 10 dB in each subsequent training session until a final value of SNR^{13} ($-20\ dB$). This process ensures that the performance can be quickly maximized for low SNR while remaining accurate for signals with high SNR. For instance, Fig. 9 shows the training process of a mono-component CV DNN at $SNR^{10} = 10\ dB$, where Scheme 2 is the loss curve obtained when trained using this scheme with gradually decreasing SNR and Scheme 1 is obtained without decreasing the SNR. Scheme 2 clearly enables the network to learn the training features under higher SNR^i. Moreover, it makes the training loss under lower SNR^i converge better and more quickly: Scheme 2 only needs two epochs to converge the loss, whereas Scheme 1 needs eight epochs. Meanwhile, the minimum value of the convergence using Scheme 2 is nearly 50% lower than that of Scheme 1.

A similar approach can be applied to multi-component CV DNN. Fortunately, with a trained mono-component CV DNN, one strategy to reduce the training time of multi-component CV DNNs is transfer learning [49]. The basic idea of transfer learning is to make full use of a pre-trained model, that is, a model that has been trained on a ready-made data set that can correspond to completely different problems to be solved. It is necessary to find a layer that can output reusable features in the pre-trained model, and then use the output of this layer as the input feature for training the smaller neural networks that need fewer parameters. Because pre-trained models have already learned patterns of data, this smaller network only needs to learn about the specific connections of the data to a specific problem. As mentioned earlier, the only difference between these networks is the number of neurons 2 N_c of the output layer. Therefore, we treat the parameters of the hidden layers (excluding the input and output layers) as the transfer object by starting off training the mono-component CV DNN and then transferring the parameters to a multi-component CV DNN.

IV. SIMULATION AND RESULTS

In this section, we first introduce a conventional processor, WHT, as the comparison method, then, several specific targeting experiments are conducted to observe the estimated performance of the proposed processor CV DNN.

A. PREPARATION

WHT [24], which is a representative processor combining TFT and SLD, is an effective method for analyzing chirp signals and is expressed by

$$W_s(t,f) = \int_{-\infty}^{\infty} s\left(t+\frac{\tau}{2}\right)\overline{s\left(t-\frac{\tau}{2}\right)}e^{-j2\pi tf}\,d\tau, \quad (33)$$

where $\overline{(\cdot)}$ denotes the complex conjugate and $W_s(t,f)$ depicts the energy distribution of $s(t)$ in frequency f at time t.

The WHT is defined as a mapping from the time domain to the parameter domain (f_0, μ) and expressed by the following integral transformation:

$$WH_s[s(t;f,\mu)] = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} s\left(t+\frac{\tau}{2}\right)s^*\left(t-\frac{\tau}{2}\right) \cdot e^{-j2\pi(f_0+\mu t)\tau}\,dt\,d\tau. \quad (34)$$

In polar coordinates, WHT is defined as

$$\begin{aligned} WH_s[s(t;\rho,\theta)] &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} s\left(t+\frac{\tau}{2}\right)s^*\left(t-\frac{\tau}{2}\right) \\ &\quad \cdot e^{-j2\pi\frac{1}{\sin\theta}(\rho+\cos\theta\cdot t)\tau}\,dt\,d\tau \\ &= \int_{-\infty}^{+\infty} W_s\left(t, \frac{1}{\sin\theta}(\rho+\cos\theta\cdot t)\right)dt, \end{aligned} \quad (35)$$

where $\theta \in (0, 2\pi)$.

Because of the oscillating properties [50] of the cross terms in WVD, the cross terms can be effectively suppressed on the parameter plane by the Hough transform, which enables the WHT to achieve high estimation accuracy. Therefore, it makes sense to take WHT as the comparison object of the proposed processor.

The main issues addressed in the experiments are as follows:

- In Experiment 1, the sensitivities of WHT and CV DNN to chirp signal parameters are compared.
- In Experiment 2, the estimated accuracy and computational efficiency of WHT and CV DNN are studied using a set of samples.
- In Experiment 3, the sensitivities of WHT and CV DNN to the number of components N_c ($N_c > 1$) of a multi-component chirp signal are studied.

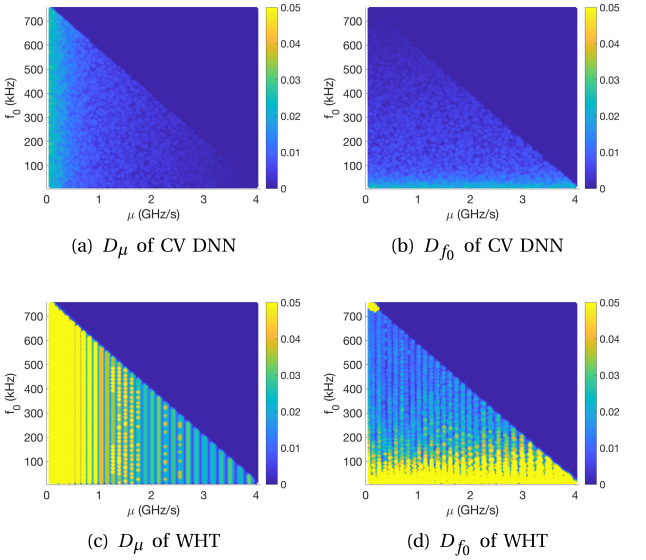
B. EXPERIMENT 1

To study the sensitivity of the processors to chirp signals with different parameters, under each test template (the red dot shown in Fig. 7), a mono-component chirp signal with $SNR = 10\ dB$ was generated as a sample. All samples make up the sample set in Experiment 1. Then, each sample was processed in a Monte Carlo simulation (MCS) with 200 trials using WHT and a mono-component CV DNN, respectively.

The degree of deviation under each parameter template is expressed as relative error as follows:

$$D_P = \frac{1}{M}\sum_{1}^{M}\frac{\left|P_{estimation} - P_{accuracy}\right|}{P_{accuracy}}, \quad (36)$$

where M is the number of MCS trials, $P_{estimation}$ is the parameter value estimated by the processors, and $P_{accuracy}$ is the accurate value of the parameter, where $P \in \{f_0, \mu\}$.



(a) D_μ of CV DNN (b) D_{f_0} of CV DNN

(c) D_μ of WHT (d) D_{f_0} of WHT

FIGURE 10. Sensitivity of CV DNN and WHT to chirp signal parameters.

Figure 10 shows the error matrix obtained by CV DNN and WHT. The triangular region represents the parameter distribution of the sample set, which is same as the testing set's parameter distribution in Fig. 7. Different colors represent the magnitude of the relative error D_P, whose upper bound shown in the figure is 0.05, i.e., for each point $D_P = 0.05$ in the figure, its true $D_P \geqslant 0.05$. To make the error matrix more general, we performed a cubic spline interpolation on the error matrix and extended its size from 100×100 to $1,024 \times 1,024$. Figures 10(a) and 10(c) respectively show the error matrix amplitude diagram of CV DNN and WHT estimating μ on the sample set, whereas Fig. 10(b) and 10(d) respectively show the results for f_0.

The simulation shows that CV DNN is more robust to the distribution of chirp signal parameters than WHT.

These results show that in the sample set, WHT has a large estimation error for the samples associated with a lower μ or lower f_0, especially when $\mu, f_0 \to 0$. For instance, when the μ of a sample is less than $0.5\ GHz/s$, D_μ is always $\geqslant 0.05$. Similarly, $D_{f_0} \geqslant 0.05$ for samples with f_0 less than $50\ KHz$. Furthermore, as μ and f_0 increase, the relative error gradually decreases, but oscillates as μ varies.

For CV DNN, although the region with lower parameter values also shows higher errors, the sensitive region is

relatively smaller, the relative errors are smaller, and the stability in the error variation is much better than that of WHT. For instance, in the D_μ matrix, D_μ is always $\leqslant 0.05$, D_μ is $\geqslant 0.04$ only when $\mu \leqslant 0.1\ GHz/s$, and $D_\mu \leqslant 0.02$ invariably when $\mu \geqslant 1\ GHz/s$. Similarly, the maximum of matrix D_{f_0} is also $\leqslant 0.05$, and $D_{f_0} \leqslant 0.04$ when $f_0 \leqslant 30\ KHz$.

C. EXPERIMENT 2

1) OBTAINING EXPERIMENT SAMPLES

Three mono-component chirp signals with the same amplitude were generated for the experiments as samples of Experiment 2 and Experiment 3, whose modulation parameters are different from each element in Fig. 7 but within the template's scope.

- The parameters of the samples $s_i\{f_{i,0}, \mu_i\}$ are
  $s_1\{300\ KHz, 2.63\ GHz/s\}$,
  $s_2\{400\ KHz, 0.88\ GHz/s\}$,
  $s_3\{200\ KHz, 0.44\ GHz/s\}$.

For a clearer representation, Fig. 11(a) shows the sum of WVD for each mono-component chirp signal, i.e., $\mathbf{WVD}=\left|\sum_{i=1}^{3} \boldsymbol{W}_{\boldsymbol{s_i}}\right|$, and 11(c) shows its WHT, $\sum_{i=1}^{3} \boldsymbol{WH}_{\boldsymbol{s_i}}$. Moreover, Fig. 11(b) and 11(d) show the $\mathbf{WVD} = \left|\boldsymbol{W}_{\sum_{i=1}^{N_c} s_i}\right|$ and $\boldsymbol{WH}_{\sum_{i=1}^{3} s_i}$ of a 3-component chirp signal composed of three mono-component chirp signals.

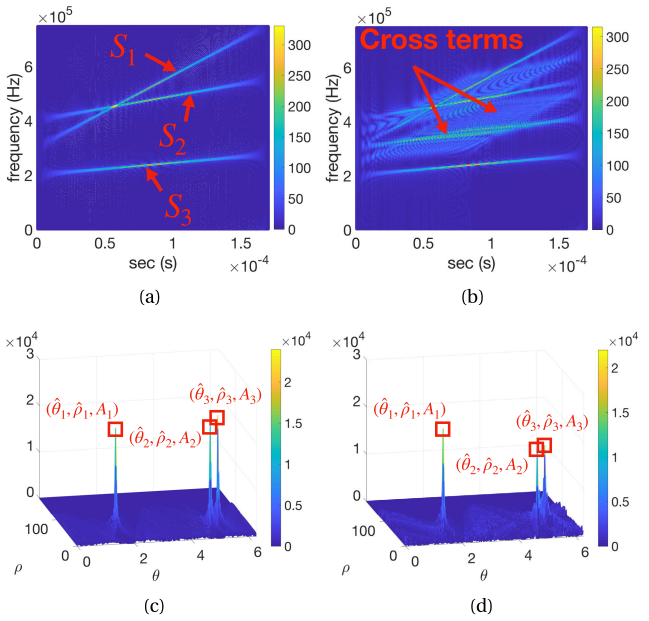


FIGURE 11. (a) $\mathbf{WVD}=\left|\sum_{i=1}^{3} W_{s_i}\right|$, (b) $\mathbf{WVD} = \left|W_{\sum_{i=1}^{N_c} s_i}\right|$, (c) $\sum_{i=1}^{3} WH_{s_i}$, and (d) $WH_{\sum_{i=1}^{3} s_i}$.

2) ESTIMATION PERFORMANCE

A mixed 2-component chirp signal was composed of mono-components s_1 and s_3, which have different initial frequencies and chirp rates, as $s = s_1 + s_3$, as shown in Fig. 11(a).

Further, to strengthen the comparison, we introduce a part-search maximum likelihood estimation (PSMLE) [7] on the basis of WHT, i.e., MLE is carried out in the region of the estimated parameters obtained by WHT to obtain a better estimation, as follows:

$$\{\hat{f}_0, \hat{\mu}\} = \underset{\{(f_0,\mu)\}\in\Omega}{\arg\max} \left|\sum_{n=0}^{N-1} x(n) \mathrm{e}^{-\mathrm{j}2\pi\left(f_0 n + \frac{1}{2}\mu n^2\right)}\right|, \tag{37}$$

where

$$\left\{(f_0, \mu) \,\middle|\, \left\|f_0 - \hat{f}_0^{WH_s}\right\|_2 < \varsigma_{f_0}, \left\|\mu - \hat{\mu}^{WH_s}\right\|_2 < \varsigma_\mu\right\} \subseteq \Omega, \tag{38}$$

where ς is search radius, $\hat{f}_0^{WH_s}$ and $\hat{\mu}^{WH_s}$ are parameters estimated by WHT, and the search step is $\frac{2\varsigma}{100}$. PSMLE serves as a reference for asymptotically optimal estimation.

An MCS with 200 trials was run with different SNRs (-15 dB to 10 dB in 1 dB intervals). The RMSE in the form of dB is used, as the evaluation criteria of Experiment 1. It is defined as

$$RMSE(dB) = 10 * \log\left[\frac{1}{M}\sum_{1}^{M}\left(P_{estimation} - P_{accuracy}\right)^2\right]^{\frac{1}{2}}. \tag{39}$$

The RMSE (dB) of the initial frequencies and chirp rates obtained by a trained 2-component CV DNN, WHT, and PSMLE are shown in Fig. 12. Here, the CRLB [10] serves as a reference. TABLE 1 lists the average runtime of MCS conducted by these processors under different SNRs.

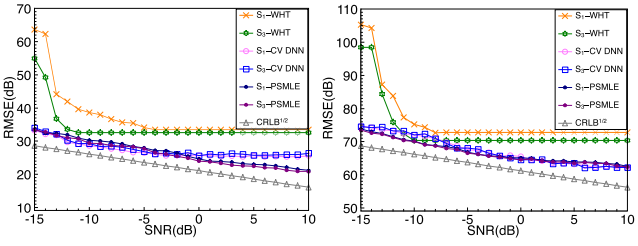


FIGURE 12. RMSE (dB) results for (a) initial frequency f_0 estimation and (b) chirp rate μ estimation.

TABLE 1. Average runtime of 200 MCSs.

Method	Time Consumed
WHT	32.176 s
PSMLE	437.917 s (+32.176 s)
CV DNN	0.1701 s

These results allow us to draw some conclusions about the methods:

1) CV DNN has a lower SNR threshold, i.e. when SNR < -12 dB, the estimation accuracy of WHT drops significantly, whereas that of the CV DNN is still robust. Moreover, the error curve of the CV DNN is

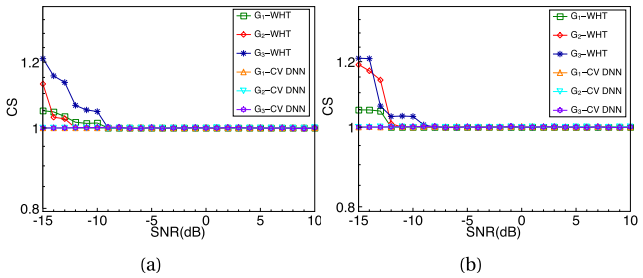


FIGURE 13. CS estimation results: (a) initial frequency f_0 estimation and (b) chirp rate μ estimation.

closer to that of PSMLE and CRLB$^{1/2}$, which indicates that the CV DNN is close to the optimal estimation and much better than WHT in terms of estimation accuracy.

2) CV DNN is insensitive to the change in signal parameters. The error curves of s_1 and s_3, which have different parameters under the CV DNN, are similar whereas the error curves under WHT have different SNR threshold results, which supports the conclusion of Experiment 1.
3) Although the neural network-based approaches require a substantial amount of time to train the network in the early stages, the time spent on subsequent calculations for the trained network is greatly reduced. A trained CV DNN improves the computational efficiency by nearly 200 times compared with that of WHT and by 2,700 times compared with that of PSMLE.

D. EXPERIMENT 3

For multi-component chirp signals, mutual interference caused by the aliasing between components may influence the parameter estimation accuracy. Therefore, we designed an experiment to study the sensitivity of the proposed processor to the number of chirp signal components by observing the estimation accuracy of a specified component in different multi-component chirp signal combinations.

Three mono-component chirp signals shown in Fig. 11(a) were combined into different groups, where the multi-component signals are composed of different mono-component signals, $G = \{s_1 + s_2, s_1 + s_3, s_1 + s_2 + s_3\}$. By studying the performance of the s_1 estimation results under different groups, the impact of changes in the number of components on the results can be analyzed. The evaluation criterion is component sensitivity (CS), which is defined as the ratio of the s_1 estimation results in the multi-component signal to that of s_1 in the mono-component signal, i.e.,

$$CS(i) = \frac{1}{M}\sum_{1}^{M}\frac{RMSE(s_1 \in G_i, i = 1, 2, 3)}{RMSE(s_1)}. \quad (40)$$

CV DNN and WHT were used for an MCS with 200 trials in each group. Note that, in the CV DNN processor, s_1 in the denominator of (40) is the input of a trained mono-component network and G_1 and G_2 are the inputs of the same network structure, which is a trained 2-component network, whereas G_3 relies on another network structure, a trained 3-component network. Figure 13 shows the CS results of these processors.

The results indicate that both CV DNN and WHT are insensitive to changes in N_c. The mutation of WHT's CS is caused by the SNR threshold effect. A reasonable guess about the stability of the CS of CV DNN is that transfer learning enables multi-component CV DNN to inherit and learn features that are common to mono-component CV DNNs, which makes the network robust to the number of components or the number of neurons in the output layer.

V. CONCLUSION AND FUTURE WORK

There are many parameter estimation processors of chirp signals. Most of them are composed of parts with clear functions. From this perspective, because of the strong advantages of neural networks in function fitting and feature extraction, we replace each part in the conventional processor with a neural network to achieve comparable or better performance. In this paper, a CV DNN processor was proposed to replace a conventional time–frequency analysis method processor. The results of simulation experiments show that compared with the conventional processor, the proposed processor has the following advantages:

1) It is robust to the parameters of a chirp signal.
2) The estimation accuracy and computational efficiency are greatly improved.
3) The estimation accuracy of each component in a multi-component chirp signal is not affected by other components, i.e., the proposed processor is robust to the number of components of the chirp signal.

We revisit here the assumptions made in Section III-A. The proposed system can be thought of as a processor to estimate the parameters of a chirp signal with a known number of components. In a channel that contains complex signals with different numbers of components, if the number of components can be determined using classification, the parameter estimation of a complex signal could be realized by the parallel operation of the processor proposed in this paper. Besides, it will also be necessary to add other forms of interference to the training data so that the network can accurately distinguish chirp signals from interference. Therefore, in future work, we will discuss the application of the proposed method in complex electromagnetic environments. Further, signal length, network structure, and data set size are common factors affecting network learning ability in deep learning, and these topics are also worthy of further discussion and research.

REFERENCES

[1] A. G. Stove, "Linear FMCW radar techniques," *IEE Proc. F Radar Signal Process.*, vol. 139, no. 5, pp. 343–350, Oct. 1992.
[2] P. R. Atkins, T. Collins, and K. G. Foote, "Transmit-signal design and processing strategies for sonar target phase measurement," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 1, pp. 91–104, Jun. 2007.

[3] T. Misaridis and J. A. Jensen, "Use of modulated excitation signals in medical ultrasound. Part I: Basic concepts and expected benefits," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 52, no. 2, pp. 177–191, Feb. 2005.

[4] M. Martone, "A multicarrier system based on the fractional Fourier transform for time-frequency-selective channels," *IEEE Trans. Commun.*, vol. 49, no. 6, pp. 1011–1020, Jun. 2001.

[5] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, Jul. 2005.

[6] T. J. Abatzoglou, "Fast maximnurm likelihood joint estimation of frequency and frequency rate," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-22, no. 6, pp. 708–715, Nov. 1986.

[7] R. Liang and K. Arun, "Parameter estimation for superimposed chirp signals," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 5, Mar. 1992, pp. 273–276.

[8] X. Lv, M. Xing, S. Zhang, and Z. Bao, "Keystone transformation of the Wigner–Ville distribution for analysis of multicomponent LFM signals," *Signal Process.*, vol. 89, no. 5, pp. 791–806, May 2009.

[9] T. J. Abatzoglou, "Fast maximnurm likelihood joint estimation of frequency and frequency rate," *IEEE Trans. Aerosp. Electron. Syst.*, no. 6, pp. 708–715, 1986.

[10] S. Peleg and B. Porat, "Linear FM signal parameter estimation from discrete-time observations," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 27, no. 4, pp. 607–616, Jul. 1991.

[11] R. A. Altes, "Detection, estimation, and classification with spectrograms," *J. Acoust. Soc. Amer.*, vol. 67, no. 4, pp. 1232–1246, Apr. 1980.

[12] M. Sanghadasa, P. S. Erbach, C. C. Sung, D. A. Gregory, and W. A. Friday, "Wavelet transform applied to synthetic aperture radar—Optical implementation and adaptive techniques," *Opt. Eng.*, vol. 33, no. 7, pp. 2282–2290, 1994.

[13] V. Katkovnik, "A new form of the Fourier transform for time-varying frequency estimation," *Signal Process.*, vol. 47, no. 2, pp. 187–200, 1995.

[14] S. Kay and G. Boudreaux-Bartels, "On the optimality of the Wigner distribution for detection," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 10, Apr. 1985, pp. 1017–1020.

[15] J.-C. Cexus and A.-O. Boudraa, "Nonstationary signals analysis by Teager-Huang transform (THT)," in *Proc. 14th Eur. Signal Process. Conf.*, 2006, pp. 1–5.

[16] N. E. Huang, N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. Roy. Soc. London Ser. A, Math., Phys. Eng. Sci.*, vol. 454, no. 1971, pp. 903–995, Mar. 1998.

[17] R. G. Stockwell, L. Mansinha, and R. P. Lowe, "Localization of the complex spectrum: The S transform," *IEEE Trans. Signal Process.*, vol. 44, no. 4, pp. 998–1001, Apr. 1996.

[18] T. P. Zielinski, "Joint time-frequency resolution of signal analysis using Gabor transform," *IEEE Trans. Instrum. Meas.*, vol. 50, no. 5, pp. 1436–1444, Oct. 2001.

[19] L. Qi, R. Tao, S. Zhou, and Y. Wang, "Detection and parameter estimation of multicomponent LFM signal based on the fractional Fourier transform," *Sci. China F, Inf. Sci.*, vol. 47, no. 2, p. 184, 2004.

[20] R. Tao, Y.-L. Li, and Y. Wang, "Short-time fractional Fourier transform and its applications," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2568–2580, May 2010.

[21] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, 1981.

[22] G. Beylkin, "Discrete radon transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 35, no. 2, pp. 162–172, Feb. 1987.

[23] P. Milne and P. E. Pace, "Wigner distribution detection and analysis of FMCW and p-4 polyphase LPI waveforms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 4, May 2002, pp. IV-3944–IV-3947.

[24] S. Barbarossa, "Analysis of multicomponent LFM signals by a combined Wigner-Hough transform," *IEEE Trans. Signal Process.*, vol. 43, no. 6, pp. 1511–1515, Jun. 1995.

[25] A. Bouchikhi, A.-O. Boudraa, J.-C. Cexus, and T. Chonavel, "Analysis of multicomponent LFM signals by Teager Huang-Hough transform," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 2, pp. 1222–1233, Apr. 2014.

[26] B. Du, W. Xiong, J. Wu, L. Zhang, L. Zhang, and D. Tao, "Stacked convolutional denoising auto-encoders for feature representation," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1017–1027, Apr. 2016.

[27] B. Du, Z. Wang, L. Zhang, L. Zhang, and D. Tao, "Robust and discriminative labeling for multi-label active learning based on maximum correntropy criterion," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1694–1707, Apr. 2017.

[28] Z. Wang, B. Du, and Y. Guo, "Domain adaptation with neural embedding matching," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.

[29] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[30] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Artificial Neural Networks and Machine Learning—ICANN*, T. Honkela, W. Duch, M. Girolami, and S. Kaski, Eds. Berlin, Germany: Springer, 2011, pp. 52–59.

[31] D. CireşAn, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Netw.*, vol. 32, pp. 333–338, Aug. 2012.

[32] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 37, F. Bach and D. Blei, Eds. Lille, France: PMLR, Jul. 2015, pp. 448–456. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html

[33] M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, and S. Mougiakakou, "Lung pattern classification for interstitial lung diseases using a deep convolutional neural network," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1207–1216, 2016.

[34] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, p. 115, 2017.

[35] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2267–2273.

[36] M. Hu *et al.*, "Memristor-based analog computation and neural network classification with a dot product engine," *Adv. Mater.*, vol. 30, no. 9, p. 1705914, 2018.

[37] Z.-M. Liu, C. Zhang, and P. S. Yu, "Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections," *IEEE Trans. Antennas Propag.*, vol. 66, no. 12, pp. 7315–7327, Dec. 2018.

[38] D. George and E. Huerta, "Deep neural networks to enable real-time multimessenger astrophysics," *Phys. Rev. D, Part. Fields*, vol. 97, no. 4, p. 044039, 2018.

[39] M. A. Razi and K. Athappilly, "A comparative predictive analysis of neural networks (NNs), nonlinear regression and classification and regression tree (CART) models," *Expert Syst. Appl.*, vol. 29, no. 1, pp. 65–74, 2005.

[40] G. Shi, M. M. Shanechi, and P. Aarabi, "On the importance of phase in human speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 14, no. 5, pp. 1867–1874, Sep. 2006.

[41] 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," in *Proc. Int. Conf. Learn. Represent.*, 2018. [Online]. Available: https://openreview.net/forum?id=H1T2hmZAb

[42] J. Gao, B. Deng, Y. Qin, H. Wang, and X. Li, "Enhanced radar imaging using a complex-valued convolutional neural network," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 1, pp. 35–39, Jan. 2019.

[43] Z. Zhang, H. Wang, F. Xu, and Y.-Q. Jin, "Complex-valued convolutional neural network and its application in polarimetric SAR image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7177–7188, Dec. 2017.

[44] G. M. Georgiou and C. Koutsougeras, "Complex domain backpropagation," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 39, no. 5, pp. 330–334, May 1992.

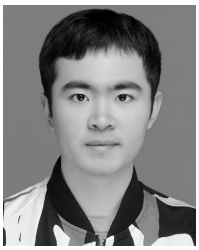
[45] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi, "Learning activation functions to improve deep neural networks," Dec. 2014, *arXiv:1412.6830*. [Online]. Available: https://arxiv.org/abs/1412.6830

[46] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object reognition?" in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 2146–2153.

[47] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognit. Modeling*, vol. 5, no. 3, p. 1, 1988.

[48] R. Hänsch and O. Hellwich, "Classification of polarimetric SAR data by complex valued neural networks," in *Proc. Workshop High-Resolution Earth Imag. Geospatial Inf. (ISPRS)*, vol. 38, 2009, pp. 4–7.

[49] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[50] S. Barbarossa and A. Zanalda, "A combined Wigner-Ville and Hough transform for cross-terms suppression and optimal detection and parameter estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 5, Mar. 1992, pp. 173–176.



HANNING SU received the B.S. degree in electronic engineering from Xidian University, in 2018. He is currently pursuing the M.A. degree with the ATR Laboratory, National University of Defense Technology. His current research interests include array signal processing, deep learning, and passive radar.

QINGLONG BAO received the B.S. and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 2003 and 2010, respectively. He is currently an Associate Professor with the National University of Defense Technology. His current research interests include radar data acquisition and signal processing.

ZENGPING CHEN received the B.S. and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 1987 and 1994, respectively. He is currently a Professor and a Ph.D. Supervisor with the Sun Yat-sen University. His current research interests include signal processing, radar systems, and automatic target recognition.