Chapter 1
Introduction

Signals occurring in applications like medical imaging and telecommunications are inherently complex-valued, and processing them in their natural form preserves the physical characteristics of these signals. Therefore, there is a widespread research interest in developing efficient complex-valued neural networks along with their learning algorithms. However, operating in the Complex domain presents new challenges; foremost among them being the choice of an appropriate complex-valued activation function. Basically, an activation function for a neural network is required to be nonlinear, bounded and differentiable in every point on the considered plane [1]. This implies that in the Complex domain, the function has to be nonlinear, bounded and entire. However, Liouville's theorem states that an entire and bounded function in the Complex domain is a constant (function) [2]. As neither the analyticity and boundedness can be compromised, nor is a constant function acceptable as an activation function as it cannot project the input space to a non-linear higher dimensional space, choices for activation functions for complex-valued neural network are limited. In this chapter, the different complex-valued neural networks existing in the literature are discussed in detail, along with their limitations.

Complex-valued neural networks can be broadly classified into different categories as shown below:

- Nature of Complex-valued Neural Networks
  - Split complex-valued neural network
  - Fully complex-valued neural network
- Type of Learning
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning
- Mode of Learning
  - Batch learning
  - Sequential learning

S. Suresh et al.: Supervised Learning with Complex-valued Neural Networks, SCI 421, pp. 1–29.
springerlink.com

- Applications
  - Array signal processing
  - Wireless communication: QAM equalization
  - Memories
  - Real-valued classification and other applications

1.1 Nature of Complex-valued Neural Networks

Based on the nature of signals (i.e., the inputs signals/features, the output signals/variables and the weight parameters), which are real-valued or complex-valued, the complex-valued neural networks are classified into split complex-valued neural networks and fully complex-valued neural networks.

1.1.1 Split Complex-valued Neural Network

Initially, split complex-valued networks [3] were used to operate on complex-valued signals. The split complex-valued networks are further divided into two types based on the nature of weights, namely, split complex networks with real-valued weights and real-valued activation functions and split complex networks with complex-valued weights and real-valued activation functions.

1.1.1.1 Real-Valued Weights and Real-Valued Activation Functions

The neural network architecture is similar to the classical Multi-Layer Perceptron (MLP) network with the real-valued back propagation algorithm. Here, the complex-valued inputs and targets are split into two real-valued quantities (the real and imaginary components), based on either a rectangular (real-imaginary) or polar (magnitude-phase) coordinate system. For example, two complex-valued inputs and one complex-valued output problem is converted into four real-valued inputs and two real-valued outputs problem such as:

$$\begin{Bmatrix} z_1 \\ z_2 \end{Bmatrix} \rightarrow \begin{Bmatrix} z_{1R} \\ z_{1I} \\ z_{2R} \\ z_{2I} \end{Bmatrix} \tag{1.1}$$

$$\begin{Bmatrix} z_1 \\ z_2 \end{Bmatrix} \rightarrow \begin{Bmatrix} r_1 \\ \phi_1 \\ r_2 \\ \phi_2 \end{Bmatrix} \tag{1.2}$$

where z_{1R} and z_{1I} are the real and imaginary parts of the complex-valued signal z_1 and r_1 and ϕ_1 are its magnitude and phase components, respectively. Similarly, z_{2R}, z_{2I}, r_2 and ϕ_2 are the real part, imaginary part, magnitude and phase of the complex-valued signal z_2. The input and output weights of such a network are also real-valued. The architecture of a such split complex-valued multi-layer perceptron network with a single hidden layer is shown in Fig. 1.1.

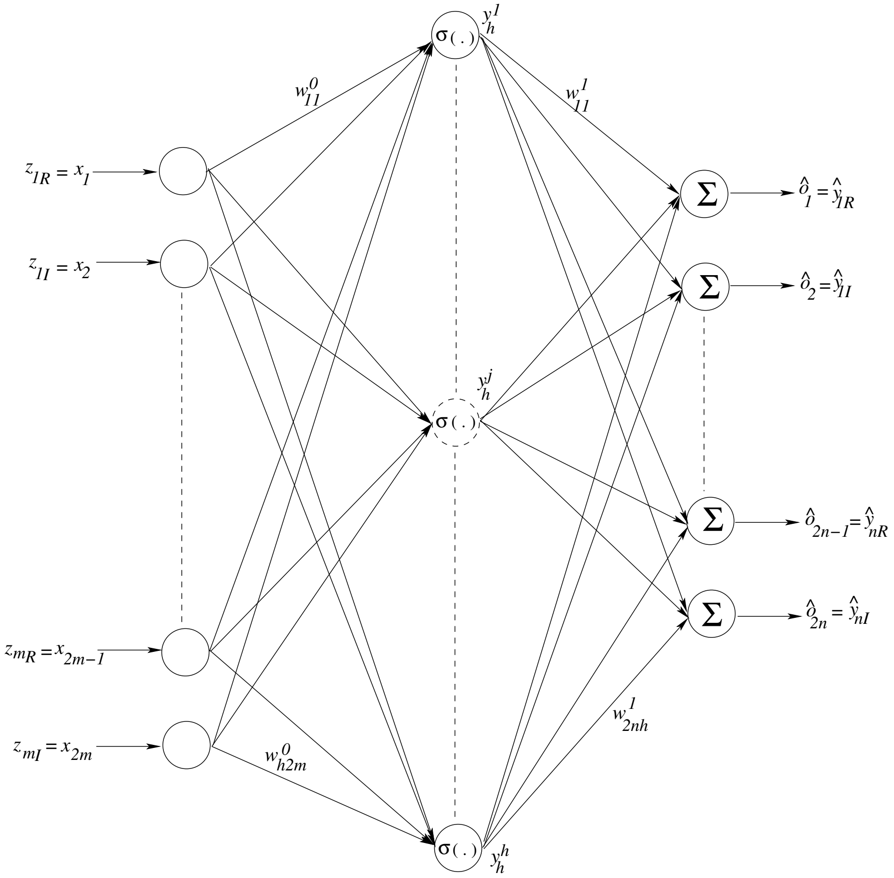

Fig. 1.1 Architecture of a Split Complex-valued Multi-Layer Perceptron Network

In the figure,

- $\mathbf{x} \in \mathbb{R}^{2m} = [x_1, x_2, \cdots, x_{2m}] = [z_{1R}, z_{1I}, \cdots z_{mR}, z_{mI}]^T$ are the real-valued inputs.
- $W^0 \in \mathbb{R}^{h\times 2m} = \begin{bmatrix} w^0_{11} & \cdots & w^0_{12m} \\ \vdots & & \\ w^0_{h1} & \cdots & w^0_{h2m} \end{bmatrix}$ are the real-valued weights connecting the input layer and hidden layer.
- $\mathbf{y}_h \in \mathbb{R}^h = [y^1_h, y^2_h, \cdots, y^h_h]^T$ are the real-valued response of the hidden neurons given by $y^k_h = \sigma(\sum_{j=1}^{2m} w_{kj}x_j)$; $\sigma(.)$ is a unipolar or bipolar sigmoidal activation function.

- $W^1 \in \mathbb{R}^{2n \times h} = \begin{bmatrix} w^1_{11} & \cdots & w^1_{12m} \\ \vdots & & \\ w^1_{h1} & \cdots & w^1_{h2m} \end{bmatrix}$ are the real-valued weights connecting the hidden layer and output layer.
- $\widehat{\mathbf{o}} \in \mathbb{R}^{2n} = [\widehat{o}_1, \; \cdots, \; \widehat{o}_{2n}]^T$ are the real-valued outputs of the network. The complex-valued predicted output of a split complex-valued multi-layer perceptron is reconstructed as $\widehat{y}_1 = \widehat{o}_1 + i\widehat{o}_2$.

Hence, it can be observed that the network is in fact an MLP network operating on real-valued signals. As each complex-valued signal is split into two real-valued signals, the number of neurons in the input and output layers are twice the number of inputs and outputs (of the function to be approximated). This results in an increased network size and hence, more parameters to be estimated. Also, irrespective of the kind of splitting the complex-valued signal, real-valued gradients are used to update the network parameters of such a split complex-valued network. These real-valued gradients do not reflect the true complex-valued gradient, as stated by Kim and Adali [4]. Hence, approximation using the split complex-valued neural networks results in inaccurate approximations and introduces phase distortion in the complex-valued function that is approximated. Moreover, from the sensitivity analysis on the split complex-valued networks presented by Yang *et al.* [5] and from the convergence study on split complex-valued networks presented by Zhang *et al.* [6], it can be observed that the convergence of split complex-valued network with real-valued weights depends on proper initialization and the choice of the learning rate [7].

1.1.1.2 Complex-Valued Weights and Real-Valued Activation Functions

To overcome the problem of phase distortions due to the splitting of complex-valued signal, complex-valued weights with real-valued activation functions have been presented by Deng *et al.* [8, 9] and Benvenuto *et al.*, [10]. The Gaussian function that maps the Complex domain into the Real domain, $f : \mathbb{C}^n \to \mathbb{R}$ is used as the basis of the activation function in these networks. Hence, these networks also use real-valued activation functions and the response of the hidden layer is real-valued. In [8, 10], the product of complex-valued error and its conjugate is used to update the weights during error-correction learning. As the Real part of error is used to update the Real part of the free parameters, and Imaginary part of error to update Imaginary part of the free parameters, the correlation between the real and imaginary components of the weight and error is lost during learning. Hence the gradients, used to update the parameters during learning are not true representation of the complex-valued gradients [4]. Therefore, approximation using such a network is also not very accurate.

1.1.2 Fully Complex-valued Neural Networks

The fully complex-valued network with a fully complex-valued activation function is capable of handling complex-valued inputs and outputs efficiently. The architecture of a fully complex-valued neural network is given in Fig. 1.2.

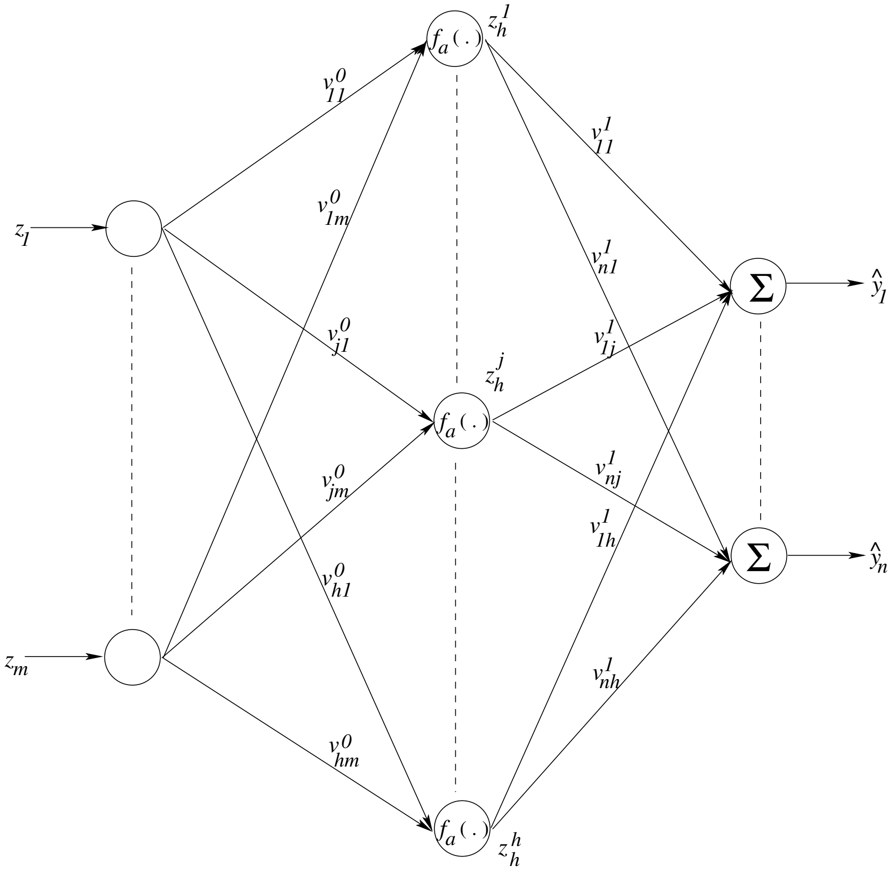

Fig. 1.2 Architecture of a Fully Complex-valued Neural Network

In the figure,

- $\mathbf{z} \in \mathbb{C}^m = [z_1,\ z_2,\ \cdots,\ z_m]^T$ are the complex-valued inputs to the network.
- $V^0 \in \mathbb{C}^{h\times m} = \begin{bmatrix} v_{11}^0 & \cdots & v_{1m}^0 \\ \vdots & & \\ v_{h1}^0 & \cdots & v_{hm}^0 \end{bmatrix}$ are the complex-valued weights connecting the input layer and hidden layer.
- $\mathbf{z}_h \in \mathbb{C}^h = [z_h^1,\ z_h^2,\ \cdots,\ z_h^h]^T$ are the complex-valued response of the hidden neurons given by $y_h^k = f_a(\sum_{j=1}^m v_{kj}^0 z_j;\ k = 1, 2, \cdots h)$. Here, $f_a(.)$ is a complex-valued activation function with sigmoidal characteristics.
- $V^1 \in \mathbb{C}^{n\times h} = \begin{bmatrix} v_{11}^1 & \cdots & v_{1m}^1 \\ \vdots & & \\ w_{h1}^1 & \cdots & w_{hm}^1 \end{bmatrix}$ are the complex-valued weights connecting the hidden layer and output layer.

- $\widehat{\mathbf{y}} \in \mathbb{C}^n = [\widehat{y}_1, \cdots, \widehat{y}_n]^T$ are the complex-valued outputs of the network given by

$$\widehat{y}_l = \sum_{k=1}^{K} v_{lk}^1 z_h^k \tag{1.3}$$

It can be observed from the figure that unlike the split complex-valued networks, the fully complex-valued networks operate on the complex-valued signals, and hence use only a fewer neurons in the input and output layers. The learning algorithm used in the fully complex-valued neural network relies on well-defined complex-valued gradients[1]. Thus, the main issue in a fully complex-valued neural network is the proper selection of the complex-valued activation function and the computation of its derivatives. For the activation function selection, the complex-valued function should satisfy the following essential properties stated by Georgiou and Koutsougeras in [12] :

- $f_a(z) = f_a(x+iy) = u(x,y) + iv(x,y)$. $u(x,y)$ and $v(x,y)$ should be nonlinear and bounded in x and y.
- The partial derivatives $\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}$ exist and are bounded.
- $f_a(z)$ is not entire[2].
- $\frac{\partial u}{\partial x}\frac{\partial v}{\partial y} \neq \frac{\partial v}{\partial x}\frac{\partial u}{\partial y}$ unless $\frac{\partial u}{\partial x} = \frac{\partial v}{\partial x} = 0$ and $\frac{\partial v}{\partial y} = \frac{\partial u}{\partial y} = 0$

These conditions were, then reduced and relaxed by Kim and Adali [4] as:

In a bounded domain of complex plane $\mathbb{C}$, a fully complex nonlinear activation function $f_a(z)$ needs to be analytic[3] and bounded almost everywhere.

The fully complex-valued neural network is classified further based on the nature of the complex-valued activation function used.

1.1.2.1 Elementary Transcendental Activation Functions

Kim and Adali introduced a Fully Complex-valued MLP neural network with any of the Elementary Transcendental Functions (ETF's) as an activation function and derived its fully complex-valued back propagation weight update rule [4]. The learning algorithm presented in [4] is a complex-valued version of the real-valued back-propagation algorithm. For complete details on the properties of different ETF's and the complex-valued back propagation algorithm, refer to [4]. The ETF's satisfy the properties needed for complex activation functions, but they along with their derivatives have essential, removable or isolated singularities at different locations in the complex domain [13]. For example, *asinh* is a simple ETF, which has a branch cut singularity along the imaginary axis. The derivative of *asinh* has an isolated

[1] If $z = x+iy$ and $f_a(z) = u(x,y)+iv(x,y)$, $\frac{\partial f_a}{\partial z} = \frac{1}{2}\left(\frac{\partial f_a}{\partial x} - i\frac{\partial f_a}{\partial y}\right)$ as defined in [11].

[2] In complex analysis, an entire function, also called an integral function, is a complex-valued function that is analytic over the whole complex plane.

[3] A complex function is said to be analytic on a region $\mathbb{C}$ if it is complex differentiable at every point in $\mathbb{C}$.

Table 1.1 ETF's and their singularities

Activation Function	Singular Point	Derivative	Singular point of the derivative
$tanh$	Pointwise: $(1/2+n)pi$	$sech^2$	Pointwise: $(1/2+n)\pi$
tan	Pointwise: $(1/2+n)pi$	sec^2	Pointwise: $(1/2+n)\pi$
$atanh$	Isolated: ± 1	$1/1-z^2$	Isolated: ± 1
$atan$	Isolated: $\pm i$	$1/1+z^2$	Isolated: $\pm i$
$asinh$	Branch Cut: along imag axis $\geq\ i$	$1/\sqrt(1+z^2)$	Isolated: $\pm i$
$asin$	Branch Cut: along real axis $\geq\ 1$	$1/\sqrt(1-z^2)$	Isolated: ± 1

singularity at $\pm\ i$. The list of various ETF's, and their derivatives, along with the singularities associated with the ETF and its derivative are presented in Table 1.1.

If the learning algorithm operates in the region of singularity, the parameter updates exhibit undesirable behavior affecting the convergence. Hence, the fully complex-valued algorithm is sensitive to the singularities of the activation functions, sample population distribution, weight initialization and the choice of the learning rate. It is also noteworthy that the convergence effects are different for different ETF's and they also depend on the nature of the problem, the initial weights and the learning rate. Hence, it is essential to identify a fully complex-valued activation function that is less sensitive to initial weights, learning rate and the the problem considered. One such activation function is proposed for FC-MLP networks in Chapter 2. Besides, there is also a need to develop a fully complex-valued activation function for a fully complex-valued radial basis function network. Such an activation function and its learning algorithm are presented in Chapter 3.

1.1.2.2 Axially Symmetric Activation Function

To overcome the difficulty of boundedness and analyticity of the complex-valued activation function, You and Hong introduced an axially symmetric complex-valued function to deal with QAM signals [14]. The general form of the axially-symmetric activation function is given by

$$f_a(z) = f_{aR}(x) + i f_{aI}(y) \tag{1.4}$$

where $z\ \varepsilon\ C$, $z_R = real(z)$, $z_I = imag(z)$ and $f_{aR}(.) and f_{aI}(.)$ are any continuous functions. You and Hong used

$$f_{aR}(x) = x + \alpha sin(\pi x);\ \text{and} f_{aI}(y) = y + \alpha sin(\pi y) \quad (1.5)$$

where α is the slope factor that determines the degree of non-linearity. Here, $0 < \alpha < \frac{1}{\pi}$. The axially symmetric function satisfies the essential properties of an activation function to be used in the Complex domain. The axially symmetric function is suitable for problems, which have symmetric targets, such as equalization in telecommunication. Even though the axially symmetric activation function does not have the singularity, it does not consider the correlation between the real and imaginary parts of a complex-valued signal. Also, selection of appropriate continuous activation function for a given problem is difficult [15, 7].

1.2 Types of Learning

As in real-valued networks, complex-valued networks can also be classified into supervised learning and unsupervised learning types, depending on the presence or absence of a teacher.

1.2.1 Supervised Learning

If the learning in a neural network occurs with a teacher, it is called *Supervised Learning*. In supervised learning, the teacher has a knowledge of the environment, with the knowledge being represented by a set of input-output samples, called the training dataset [1]. The objective of learning is to estimate the free parameters of the network, such that the output errors are minimized. Several supervised learning schemes are available for complex-valued networks in the framework of feed-forward neural networks and recurrent neural networks.

1.2.1.1 Feed-forward Neural Networks

Neural networks in which signals are transmitted from input nodes to output nodes, with no feedback or memory are defined as feed-forward neural networks [1]. Two well-known topologies of complex-valued feed-forward networks are complex-valued Multi-Layer Perceptron (cMLP) networks and complex-valued Radial Basis Function (cRBF) networks.

Complex-valued MLP networks: Similar to the real-valued networks, complex-valued MLP networks are popular learning paradigms based on error correction learning. The training/learning is based on a set of inputs-outputs, and learning occurs based on the minimization of the error functions. Complex-valued Back Propagation (CBP) algorithm for the complex-valued MLP network was first presented by Leung and Haykin [3]. Georgiou and Koutsougeras [12] listed the essential properties of a fully complex-valued activation function and presented an improved version of the complex-valued back-propagation algorithm. Later, Kim and Adali [4] relaxed these conditions and presented an improved version of the CBP using the Cauchy Riemann's equations[4]. Besides these, the Fully Complex-valued Extreme Learning Machine (C-ELM) presented by Li *et al.* [16] is also a direct extension of the real-valued Extreme Learning Machine (ELM) presented by Huang *et al.* [17, 18]. C-ELM algorithm determines the free parameters of the network in an analytical way and does learning faster compared to other algorithms. Similarly, the complex-valued resilient propagation network presented by Kantsila *et al.* [19] is a direct extension of the real-valued resilient propagation network [20].

In all these CBP algorithms, the activation functions are bounded *almost everywhere (a.e).* This is because, according to Liouville's theorem,a bounded entire activation function in the Complex domain is a constant function . To overcome the controversy of boundedness and analyticity of complex-valued activation function, an axially symmetric complex-valued function is introduced, to deal with QAM signals, by You and Hong [14]. However, the issue here, is the selection of appropriate functions for the real and imaginary parts of the function (f_{aR} and f_{aI} (1.5)). Therefore, there is a need to identify a fully complex-valued activation function, which is bounded in the bounded domain of the Complex plane, with its singular point away from the operating region of the fully complex-valued neural network. In other words, a fully complex-valued activation function with its singularity at $\pm\infty$ is preferred. One such activation function is discussed in Chapter 2. book.

Complex-valued RBF networks: As the real-valued radial basis function networks form another popular architecture and are well-known for their localization properties, several real-valued RBF networks and their supervised learning algorithms have been extended to the Complex domain. In [21, 22], Chen *et al.* first presented the Complex-valued RBF (CRBF) networks, which are direct extensions of the real-valued RBF networks. The structure of such a complex-valued RBF network is presented in Fig. 1.3.

In the figure,

- $\mathbf{z} \in \mathbb{C}^m = [z_1 z_2 \cdots z_m]^T$ are the complex-valued inputs to the network.
- $\mathbf{y}_h \in \mathbb{R}^h = [y_h^1,\ y_h^2,\ \cdots,\ y_h^h]^T$ are the real-valued response of the hidden neurons given by

$$y_h^k = f_g(\mathbf{z}) = exp\frac{(-\mathbf{z}-\mathbf{c})^H(\mathbf{z}-\mathbf{c})}{2\sigma^2};\ \ k = 1,2,\cdots,h \tag{1.7}$$

where, $\mathbf{c} \in \mathbb{C}^m$ is the m-dimensional complex-valued centers of the Gaussian function in the hidden neurons and $\sigma \in \mathbb{R}$ is the width of the Gaussian function in the hidden neuron and H denotes the Hermitian operator.

---

[4] For a function $f(z) = u(x,y) + iv(x,y);\ z = x + iy$, the Cauchy Riemann equations are given by:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y};\ \frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y} \tag{1.6}$$

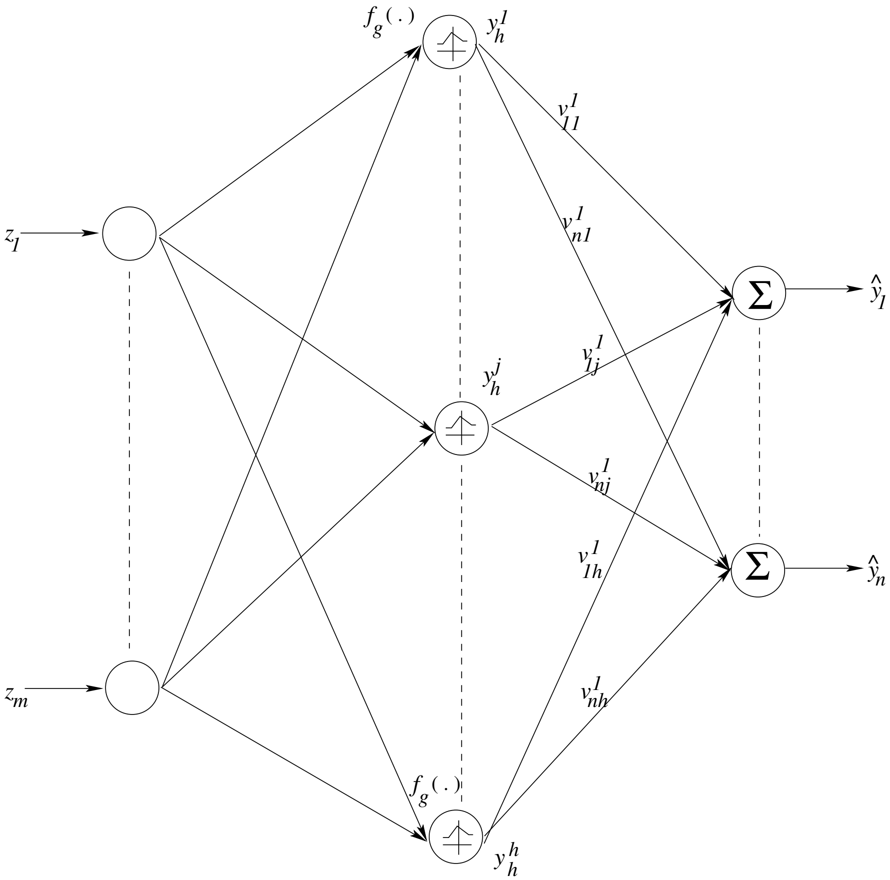

Fig. 1.3 Architecture of a Complex-valued RBF Network

- $V^1 \in \mathbb{C}^{n\times h} = \begin{bmatrix} v_{11}^1 & \cdots & v_{1m}^1 \\ \vdots & & \\ w_{h1}^1 & \cdots & w_{hm}^1 \end{bmatrix}$ are the complex-valued weights connecting the hidden layer and output layer.
- $\widehat{\mathbf{y}} \in \mathbb{C}^n = [\widehat{y}_1, \cdots, \widehat{y}_l, \cdots, \widehat{y}_n]^T$ are the complex-valued outputs of the network, given by: $\widehat{y}_l = \sum_{k=1}^{K} v_{lk}^1 y_h^k$; $l = 1, ..., l, ..., n$.

It can be seen that as the activation function maps $\mathbb{C}^m \to \mathbb{R}$, the responses of the neurons in the hidden layer are real-valued. Similarly, Complex-valued Minimal Resource Allocation Network (CMRAN), presented by Deng *et al.* [8] and Complex-valued Growing and Pruning Network (CGAP-RBF) proposed by Li *et al.* [23] are also direct extensions of the real-valued Minimal Resource Allocation Network (MRAN) proposed by Yingwei *et al.* [24] and Growing and Pruning RBF Network (GAP-RBF) presented by Huang *et al.* [25] respectively. However, in all these algorithms, the Gaussian function that maps $\mathbb{C}^m \to \mathbb{R}$ is used as the basis of the activation function. Hence, despite the centers of the RBF function being complex-valued, the response at the hidden layer is real-valued, resulting in inaccurate phase approximation. Therefore, developing a fully complex-valued RBF network with a fully complex-valued symmetric activation function, capable of better phase approximation, is very important. To address this need, chapter 3 of this book presents a fully complex-valued RBF learning algorithm.

1.2.1.2 Recurrent Neural Networks

A recurrent neural network is a class of neural network where connections between units form a directed cycle. This creates an internal state of the network which allows it to exhibit dynamic temporal behavior. Wang [26] presented a split complex valued recurrent neural network to solve complex-valued linear equations. The complex-valued coefficients were split into their real and imaginary parts and real-valued recurrent neural networks were used to solve the equations. Later, similar to the MLP and RBF framework, the algorithm for the real-valued recurrent neural network [27] is also extended to the Complex domain. Li *et al.* [28] presented a new algorithm for complex-valued recurrent neural network, where each recurrent neuron is modelled as an infinite impulse response filter. Goh and Mandic [29] introduced an augmented complex-valued extended Kalman filter algorithm for the class of nonlinear adaptive filters realized as fully connected recurrent neural networks. The structure of a complex-valued recurrent neural network which consists of N_f neurons, with p external inputs and N feedback connections is shown in Fig. 1.4 [29]. The network has two distinct layers- a feedback layer and a layer of processing elements. Let $\widehat{y}_{l,k}$ denote the complex-valued output of a neuron, $l = 1,...,n$ at time index k and $\mathbf{s}$ be the $(1 \times m)$ external complex-valued input vector (i.e., $\mathbf{s} \in \mathbb{C}^m$). The overall input to the network $\mathbf{u}_k$ then represents a concatenation of vectors $\widehat{y}_k$, s and the bias input $(1+i)$, and is given by

$$\mathbf{u}_k = [s_{k-1}, \ \cdots \ s_{k-m}, \ 1+i, \ \widehat{y}_{1,k-1} \ \cdots \ \widehat{y}_{n,k-1}]^T \tag{1.8}$$

For the l^{th} neuron, its weights form a $(m+n+1) \times 1$ dimensional weight vector $\mathbf{v}_l^T = [v_{l,1}, \cdots, v_{l,m+n+1}]$, $l = 1, \cdots, N_f$, which are encompassed in the complex-valued weight matrix of the network $V = [\mathbf{v}_1, ..., \mathbf{v}_{N_f}]^T$. The output of every neuron can be expressed as

$$y_{l,k} = f_a(net_{l,k}), \ \ l = 1, \cdots, n \tag{1.9}$$

where $f_a(.)$ is a complex-valued nonlinear activation function of a neuron and

$$net_{l,k} = \sum_{d=1}^{m+n+1} w_{ld} u_{d,k} \tag{1.10}$$

is the net input to the l^{th} neuron at time index k, where w_{ld} is the weight connecting the l^{th} output neuron and the d^{th} input neuron and $u_{d,k}$ is the d^{th} input at time index k.

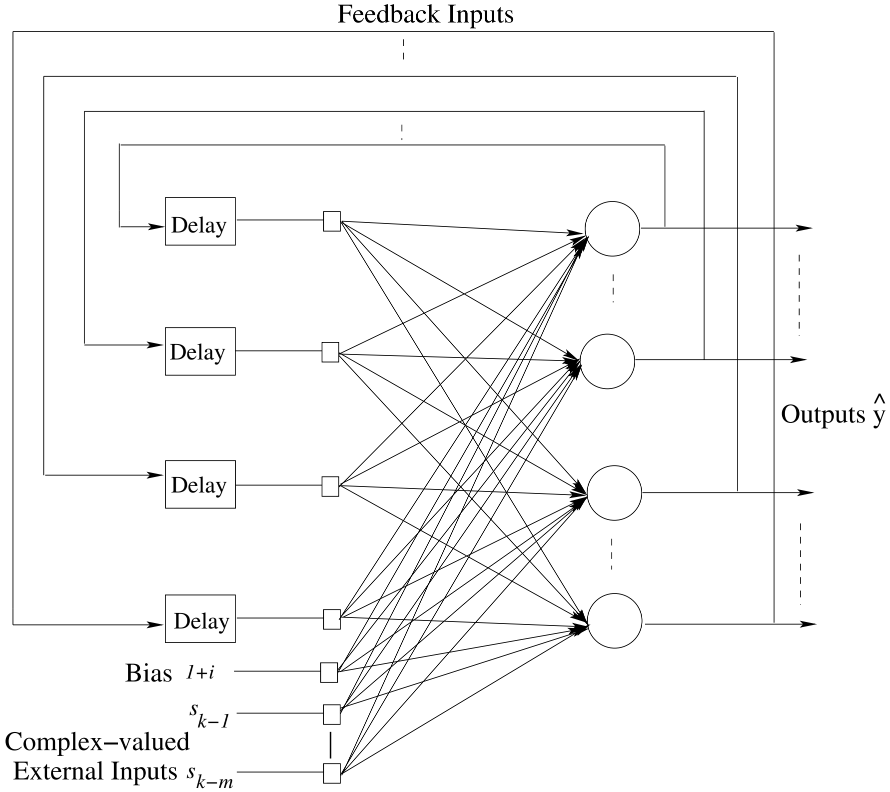

Fig. 1.4 Architecture of a Fully Connected Complex-valued Recurrent Neural Network

Mandic [30] also presented a recurrent neural network based on nonlinear autoregressive moving average models, suitable for processing the generality of complex signals. Later, Zhou and Zurada [31] addressed the boundedness, global attractivity and complete stability of the recurrent neural networks and derived some conditions for those properties. An important application of the complex-valued recurrent neural networks is the estimation of wind profile and wind power as shown by Goh *et al.* [32]. In their work, a complex-valued pipelined recurrent neural network architecture is used, and the network is trained by the complex-valued real-time recurrent learning algorithm with a fully complex-valued activation function to forecast wind signal in its complex form (speed and direction).

1.2.1.3 Error Functions for Supervised Learning

Another area of research interest in CVNN is to identify an efficient error function that minimizes both the magnitude and phase of the complex-valued error signals during learning. The mean squared error function that considers only the magnitude of the complex-valued error is the most commonly used error function. As it is an explicit representation of only the magnitude of the complex-valued error, using this error function results in inaccurate phase approximation. A list of possible choices of error functions for the complex-valued networks is presented by Gangal *et al.* [33]. Chen *et al.* [34] presented a modified error back propagation algorithm for CVNN. They added a term, corresponding to the hidden layer error, to the conventional error function to speed up the learning process. In chapter 2 of this book, we propose a logarithmic error function that uses an explicit representation of both the magnitude and phase of the complex-valued error as an error function for FC-MLP. This makes learning efficient and hence results in more accurate approximation of both the magnitude and phase of the complex-valued signals.

1.2.2 Unsupervised Learning

In unsupervised learning or self-organized learning, there is no external teacher to oversee the learning process [1]. The goal of unsupervised learning techniques is to build representations of the inputs that can be used for decision making, predicting future inputs etc. The Principal Component Analysis (PCA) and the Independent Component Analysis (ICA) are the two commonly used techniques to address this representation. In this section, we discuss the PCA and ICA learning algorithms, available in the literature.

1.2.2.1 Complex-valued Principal Component Analysis

Principal component analysis is an unsupervised learning technique used to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining the variation present in the data as much as possible. In a PCA, the multi-variate data is transformed to a new co-ordinate system using a linear orthogonal transformation.

Rattan and Hsieh [35] extended the real-valued PCA [1] to the Complex-domain. They presented a complex-valued PCA and a Non-linear Complex-valued PCA (NLCPCA) for extraction of nonlinear features from the dataset. The complex-valued neural network model for NLCPCA is an auto-associative feed-forward multi-layer perceptron model. A structure of this NLCPCA is presented in Fig. 1.5 [35].

There are m input and output neurons or nodes corresponding to the m variables. Sandwiched between the input and output layers are 3 hidden layers (starting with the encoding layer, then the bottleneck layer and finally the decoding layer) containing q, 1 and q neurons respectively. The network is composed of two parts: The first part from the input to the bottleneck maps the input z tothe single nonlinear complex principal component $f(\mathbf{z})$. The second part from the bottleneck to the output z_0 is the inverse mapping $g(f(\mathbf{z}))$. Here, the higher dimensional space (m) of the input is reduced linearly to a one-dimensional space at the bottleneck layer given by $f : \mathbb{C}^m \to \mathbb{C}^1$ and a linear inverse mapping $g : \mathbb{C}^1 \to \mathbb{C}^m$ maps from bottleneck layer to the m-dimensional output $\mathbf{z}'$, such that the least squares error function (E).

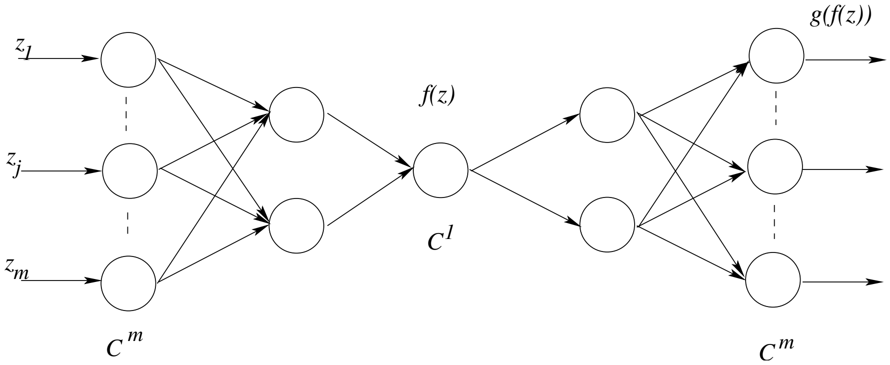

Fig. 1.5 Architecture of a NLCPCA Network

$$E = \sum_{j=1}^{n} || \mathbf{z}_j - \mathbf{z}'_j ||^2 = || \mathbf{z}_j - g(f(\mathbf{z}_j)) ||^2 \tag{1.11}$$

is a minimum (with $\mathbf{z}_j$ the j^{th} column of Z). For any input vector $\mathbf{z}$,

$$f(z) = \mathbf{v}^H \mathbf{z} \tag{1.12}$$

where $\mathbf{v}^H$ is the weight vector between the inputs and the bottleneck layer. For auto-associative networks, the target for the output neurons are simply the input data. Increasing the number of neurons in the encoding and decoding layers increases the nonlinear modelling capability of the network.

A review of the various complex-valued principal, minor components/subspace linear/nonlinear rules for complex-weighted neural structures are presented [36]. Several applications of the various unsupervised learning paradigms in the complex domain are also presented in [36].

1.2.2.2 Complex-valued Independent Component Analysis

Independent Component Analysis (ICA) is a signal-processing method used to extract independent sources given only the observed data which is a mixture of the unknown sources. The goal of the ICA can be defined as finding a linear representation of the non-Gaussian data such that the components are statistically independent or as independent as possible [37]. A number of real-valued ICA algorithms are available in the literature and a survey of the real-valued ICA algorithms and their applications are presented in [38].

Complex-valued signals arise frequently in a wide range of applications like communications [39], [40], [41], [42], [43], radar, and biomedicine [44], [45], [46], as most practical modulation formats are of complex type and applications such as radar and magnetic resonance imaging lead to data that are inherently complex-valued. The non-Gaussian nature of the complex-valued data in these applications require the development of the complex-valued ICA algorithms. Bingham *et. al.*, [47, 48] first presented the fast fixed-point type ICA algorithm that is capable of separating the complex-valued linearly mixed source signals, assuming that the original, complex-valued source signals are mutually statistically independent. Fiori *et. al.*, [49] presented an ICA algorithm to solve problems involving complex-valued signals using the maximum-mismatch learning principle. Yang *et. al.*, [50] presented an ICA method for suppression of image and co-channel interference in wireless receivers such that the channel capacity is increased and the receiver's front end is simplified.

The conditions for identifiability, separability and the uniqueness of the linear complex-valued ICA models are established in [51] by extending the well-known condition for the real-valued ICA models. Sallberg *et. al.*, presented the complex-valued ICA with the Kurtosis contrast function in [52]. This ICA algorithm does not exhibit the divergent behavior for Gaussian-only sources that occurs in the Fast ICA method. Later, Li *et. al.*, [53] derived the kurtosis maximization using a gradient update, kurtosis maximization using a fixed-point update, and kurtosis maximization using a Newton update algorithms to perform the complex independent component analysis based on the maximization of the complex kurtosis cost function. The complex maximization of the non-Gaussian cost function (Novey *et. al.*, [54]) and the entropy bound minimization (Li *et. al.*, [55]) are some of the other cost functions used in developing the ICA.

The above works focus on the development and application of the complex-valued ICA to signals of circular sources. The circularity property or properness is an important feature of many complex random signals. At the complex signal level, circularity means that the signal is statistically uncorrelated with its own complex-conjugate. In case of complex random vectors it means that the so-called complementary covariance matrix or pseudo-covariance matrix vanishes. Many widely used signals such as M-QAM and 8-PSK signals and standard complex AWGN possess this circularity property. However, practical imperfections in transmitters and receivers such as I/Q imbalance may cause departures from that property. Moreover, some well known modulation schemes such as BPSK and GMSK are non-circular. Therefore, complex-valued ICA algorithms have been developed to address the non-circularity of the complex-valued signals [56], [57]. As the conventional covariance matrix does not completely describe the second order properties of the non-circular components, Ollilaa *et. al.*, [56] used the generalized uncorrelating transformation instead of the conventional whitening transformation to develop an ICA algorithm for non-circular signals.

The ICA algorithms presented in the above-mentioned papers have been developed for holomorphic functions[5], that satisfy the Cauchy Riemann conditions. The Cauchy Riemann equations were earlier given in Section 1.2.1, Eq. (1.6) and are reproduced here for convenience .

Consider a function

[5] A complex-valued function f(z) of a complex variable z is said to be holomorphic at a point a if it is differentiable at every point within some open disk centered at a.

$$f(z) = u(x,y) + iv(x,y);\ z\ =\ x + iy. \tag{1.13}$$

Then the Cauchy Riemann equations are given by:

$$\frac{\partial u}{\partial x}\ =\ \frac{\partial v}{\partial y};\ \frac{\partial v}{\partial x}\ =\ -\ \frac{\partial u}{\partial y} \tag{1.14}$$

The holomorphic function $f(z)$ that satisfies the Cauchy Riemann equations requires that the functions $u(x,y)$ and $v(x,y)$ are harmonic[6]. But, some commonly encountered useful functions like

$$f_m(z)\ =\ \overline{z} \tag{1.16}$$
$$f_n(z)\ =\ \frac{z+\overline{z}}{2} \tag{1.17}$$
$$f_e(z)\ =\ |z|^2 = \overline{z}z = x^2 + y^2 \tag{1.18}$$
$$f_p(z)\ =\ |z| = \sqrt{\overline{z}z} = \sqrt{x^2 + y^2} \tag{1.19}$$

are not harmonic functions and are not differentiable in the standard complex variables sense. It can be noted here that the derivative of the squared error function, which is similar to f_e (Eq. (1.18)), usually employed in gradient descent based/analytical optimization algorithms does not exist in the conventional sense of the complex derivative [58],[59], [60], [61]. However, the functions in Eqs. (1.16), (1.17), (1.18) and (1.19) can be represented in the form of $f(z,\overline{z})$, where they are holomorphic in $z\ = x + iy$ for fixed $\overline{z}$ and are holomorphic in $z\ = x - iy$ for fixed z. *i.e.,*

$$\mathbb{R}\text{-derivative of } f(z,\overline{z})\ =\ \frac{\partial f}{\partial z}\ |_{\overline{z}=\text{constant}} \tag{1.20}$$
$$\overline{\mathbb{R}}\text{-derivative of } f(z,\overline{z})\ =\ \frac{\partial f}{\partial \overline{z}}\ |_{z=\text{constant}} \tag{1.21}$$

This fact underlies the development of the $\mathbb{CR}$**- calculus or the Wirtinger calculus** [2], [62]. Eq. (4.12) is called the $\mathbb{R} - derivative$ (the $real - derivative$) and Eq. (4.13) is called the $\overline{\mathbb{R}} - derivative$ (the $conjugate\, \mathbb{R} - derivative$). It is proved in [2] and [11] that the $\mathbb{R} - derivative$ and the $\overline{\mathbb{R}} - derivative$ can be equivalently written as

$$\begin{aligned}\frac{\partial f}{\partial z} &= \frac{1}{2}\left(\frac{\partial f}{\partial x} - j\frac{\partial f}{\partial y}\right)\\ \frac{\partial f}{\partial \overline{z}} &= \frac{1}{2}\left(\frac{\partial f}{\partial x} + j\frac{\partial f}{\partial y}\right)\end{aligned} \tag{1.22}$$

[6] Functions $u(x,y)$ and $v(x,y)$ are harmonic functions, if they satisfy the Laplace equations given by

$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = 0 \text{ and } \frac{\partial^2 v(x,y)}{\partial x^2} + \frac{\partial^2 v(x,y)}{\partial y^2} = 0 \tag{1.15}$$

where the partial derivatives with respect to x and y are *true* partial derivatives of the function $f(z) = f(x,y)$, which is differentiable with respect to x and y. The $\mathbb{R}-derivative$ and the $\overline{\mathbb{R}}-derivative$ are both linear differential operators that obey the product rule of differentiation and the differential rule [2]. Therefore, the $\mathbb{R}-derivative$ of $f_e(z)$ (Eq. (1.18)) with respect to a complex-valued variable a can be derived as:

$$\frac{\partial f_e}{\partial \overline{a}} = \frac{\partial (z\overline{z})}{\partial \overline{a}} = \left(\frac{\partial \overline{z}}{\partial \overline{a}}\right) z + \left(\frac{\partial z}{\partial \overline{a}}\right) \overline{z} \tag{1.23}$$

Thus the pair of partial derivatives of a non-holomorphic functions, defined by Eq. (1.22), is the natural generalization of the single Complex derivative ($\mathbb{C}-derivative$) of a complex-valued holomorphic function. Therefore, the following fact is an easy consequence of the definition in Eq. (1.22), as discussed in [11]:

- A necessary and sufficient condition for a real-valued function $f(z) = f(x,y)$; $z = x+iy$ to have a stationary point with respect to the real-valued parameters $r = (x,y)^T \in \mathbb{R}^2$ is that its $\overline{\mathbb{R}}$ vanishes.
- A necessary and sufficient condition for a real-valued function $f(z) = f(x,y)$; $z = x+iy$ to have a stationary point with respect to the real-valued parameters $\overline{r} = (x,y)^T \in \mathbb{R}^2$ is that its $\mathbb{R}$ vanishes.

This approach that is used to apply the calculus of real variables to make statements about functions of complex variables is known as Wirtinger Calculus. Differentiation of a complex function using the Wirtinger derivatives has been extensively discussed in [63].

Adali *et. al.*, [64] established the theory for Complex Independent Component Analysis (CICA) for nonlinear complex-valued signal processing based on Wirtinger calculus, such that all computations can be directly carried out in the complex domain. Two main approaches for performing ICA within the framework of Wirtinger calculus: maximum likelihood and maximization of non-Gaussianity, have been developed in [64]. The stability of the maximum likelihood complex ICA derived presented in [64] is studied in [64].

1.3 Mode of Learning

In this section, the various complex-valued neural network algorithms available in the literature are classified based on the mode of the learning algorithm. Depending on the sequence in which samples are presented to the network, a learning algorithm can be either a batch-learning algorithm or a sequential learning algorithm.

1.3.1 Complex-valued Batch Learning Algorithms

Batch learning is a learning scheme where all the samples in the training set are presented repeatedly to the network. The free parameters of the network are estimated gradually, over a series of epochs until a specified mean squared error of all the samples is achieved. Such a learning scheme requires the complete training dataset to be available apriori. Several real-valued batch learning algorithms have been extended to the Complex domain. Complex-valued batch learning algorithms are available in the framework of feedforward MLP and RBF networks. Initially, split complex-valued MLP neural networks [3, 65], which treat the complex-valued signals and weights as two real-valued signals were used to operate on complex-valued signals, and the real-valued back propagation algorithm was used to estimate the free parameters of the network. In [66], a neural network based on adaptive activation functions for SC-MLP is presented. But, from the analytical study on sensitivity and initial values of the SC-MLP [5, 67], it is observed that, the process of splitting the complex-valued signals into real and imaginary components introduces phase distortions in complex-valued approximations.

The complex-valued back-propagation algorithm based on the squared error criterion for the complex-valued feedforward network was first derived by Nitta [68], by Benvenuto and Piazza [10], and by Hirose [69]. Ever since, several activation functions have been presented for the feedforward network and the complex-valued back propagation algorithm has been modified accordingly. In [12], Georgiou and Koutsougeras listed the conditions required for a function to serve as a complex-valued activation function and proposed few activation functions for the batch learning MLP network. Later, Kim and Adali [4, 13] relaxed these conditions and suggested elementary transcendental functions as activation functions for FC-MLP. As these functions satisfy Cauchy Riemann conditions, the complex-domain back propagation algorithm was altered with the inclusion of the Cauchy Riemann conditions. Similarly, Chen *et al.* [34] developed a modified error back back-propagation algorithm to solve the problem of local minima, which is inherent to MLP networks. Kim and Guest [70] modified the complex-valued back propagation algorithm to suit to complex-valued signal processing in the frequency domain. You and Hong [14] presented an axially symmetric activation function for solving the QAM equalization problem. However, the axially symmetric activation function is specific to the application considered and choosing the axially symmetric functions for each application is a cumbersome procedure.

There are also other batch learning algorithms in the complex-domain which are direct extensions of their real-valued counterparts. For example, Li *et al.* [16] presented the Complex-valued Extreme Learning Machines (C-ELM), which is the complex-valued extension of the real-valued extreme learning machines. However, all these algorithms compromise either on the analyticity or boundedness or are application specific. Besides, the CBP algorithm presented in all these networks are derived based on the squared error function which does not consider the phase of the complex-valued error explicitly. Hence, one needs to identify an error function which is indicative of both the magnitude and phase error explicitly. In chapter 2 of this book, one such error function is proposed for the complex-valued MLP networks and an Improved Complex-valued MLP (IC-MLP) has been developed in [7] .

On the other hand, although several real-valued RBF learning algorithms [71, 72, 73, 74] have been developed for different applications, only a few of these real-valued neural networks have been extended to the Complex domain. Complex-valued RBF networks, which are direct extensions of the real-valued RBF network, was first presented by Chen *et al.* [21, 22]. For regression problems, [75], Chen *et al.* extended the locally regularized orthogonal least squares of the real-valued RBF network to the Complex domain. Chen [76] presented two training algorithms for symmetrical complex-valued RBF network. These algorithms have been applied to a non-linear beamforming problem. While one of the methods is based on a modified version of the cluster-variation enhanced algorithm, the other method is derived by modifying the orthogonal-forward-selection procedure based on the Fisher ratio of class separability measure. These networks used the Gaussian RBF with Euclidean norm $\|(x-c)\|$ to process complex-valued signals. However, in these networks, the input is not efficiently transmitted to the output, as the activation functions maps $\mathbb{C}^m \to \mathbb{R}$. Hence, all these networks do not approximate phase accurately. Therefore, it is imperative that a fully complex-valued symmetric function that maps $\mathbb{C}^m \to C$ be used as a basis for the activation function of a complex-valued RBF network. In chapter 3 of this book, one such function is proposed as a fully complex-valued activation function, based on which a fully complex-valued RBF network is presented and its gradient descent based learning algorithm is derived in [77] .

However, all the aforementioned batch learning algorithms have the following drawbacks:

- *Training dataset:* The batch learning algorithms require the training/testing dataset to be available apriori, hence the network can be trained over several epochs. However, training data may not be available apriori in most real world applications. A few applications like the cancer classification allow temporal changes to the task being learnt.
- *Network structure:* Another critical issue in batch learning algorithms is that the network structure has to be fixed apriori before learning occurs. While fewer neurons in the network result in inaccurate approximation, large network size may result in poor generalization performance and increased computational effort.

These issues in batch learning algorithms motivated the development of sequential learning schemes for neural networks. A few of these sequential learning algorithms are also extended to the Complex domain, and they are discussed briefly in the next section.

1.3.2 Complex-valued Sequential Learning Algorithms

In sequential/online learning algorithms, samples are presented one-by-one and only once to the network. They are discarded after they are presented and learnt by the network. Also, the network structure may evolve during learning, by adding and deleting neurons, as it acquires information from the training sample dataset. Few sequential learning algorithms of the above type have been extended from the Real domain to the Complex domain.

The complex-valued minimal resource allocation network [8] is the first of the kind to be extended from the real-valued minimal resource allocation network [24] sequential learning algorithm. In the CMRAN algorithm, the samples in the training dataset are used to either add a neuron or to delete a neuron, based on the magnitude error of the sample. If neither of these conditions are satisfied, the sample is used to update the parameters of the network using the real-valued extended Kalman filter.

Similarly, the complex-valued growing and pruning (CGAP-RBF) algorithm [23] is another sequential learning algorithm extended from the real-valued growing and pruning RBF [25] learning algorithm. The major differences between the CMRAN and the CGAP-RBF algorithms are:

- In CMRAN, only the sample error and its distance from the nearest neuron is considered for addition of a neuron. However, in the CGAP-RBF, in addition to these parameters, the significance of the sample to the learning accuracy is also considered for addition of a neuron. Thus while the CMRAN uses only the past history of the samples defined by the sliding window, the CGAP-RBF uses the entire past history of the samples.
- While in CMRAN, all the parameters of all the hidden neurons are updated during learning, in the CGAP-RBF algorithm, only the parameters of the nearest neuron are updated.

However, these algorithms are similar in that they use the Gaussian activation function that maps $\mathbb{C}^m \to \mathbb{R}$ and a real-valued EKF for the parameter updates during learning. Though the centers and weights are complex-valued, these algorithms use real-valued Real and Imaginary components of the complex-valued error and weights during the parameter update. Thus, it does not preserve the correlation between the Real/Imaginary components of the complex-valued error/weight information. Recently, Huang *et al.* [78] extended the Incremental Extreme Learning Machine (I-ELM) from the Real domain to the Complex domain. The algorithm randomly adds hidden nodes incrementally and analytically determines the output weights. It has been shown that in spite of the hidden nodes being generated randomly, the network constructed by I-ELM remains as an universal approximator. They show that as long as the hidden layer activation function is complex continuous discriminatory or complex bounded nonlinear piecewise continuous, I-ELM can still approximate any target functions in the complex domain [78]. However, in I-ELM, a few ETFs are used as activation functions at the hidden layer. As the ETFs and their derivatives are known to have their singularities in the finite region of the Complex domain that might interfere with the operating region of the network, identifying a suitable activation function becomes a challenging task. Therefore, there is a need to develop a fully complex-valued sequential learning algorithm, that preserves the complex-valued information, with good generalization performance and approximates phase more accurately. In chapter 7 of this book, we propose one such fully complex-valued sequential learning algorithm called the Complex-valued Self-regulatory Resource Allocation Network [79].

1.4 Applications

In this section, the various applications of the CVNNs are discussed in detail. The applications of CVNNs range from wireless communication to medical imaging. Several complex-valued memories have also been reported in the literature. A complete survey of the various applications of CVNN is discussed by Akira Hirose [80], [81]. In this section, some of the most common applications are highlighted.

1.4.1 Digital Communication: QAM Equalization

Quadrature amplitude modulation is an analog/digital modulation scheme that conveys two message signals by modulating the amplitudes of two carrier waves (quadrature carriers) that are 90° out of phase with each other. Thus, the signals in the QAM schemas are complex-valued. When the QAM signals are transmitted over a channel, the nonlinear characteristics of the channel cause spectral spreading, inter-symbol interference and constellation warping. Hence, an equalizer is essential at the receiver of the communication channel to reduce the precursor inter-symbol interference without any substantial degradation in the signal-to-noise ratio.

In the literature, several works where the complex-valued neural networks are used to solve the QAM equalization problem are reported. Many complex-valued batch learning and sequential learning schemes are used to solve the non-linear, communication channel equalization problems. For example, Chen *et al.* [22] used the batch learning complex-valued RBF network, presented in [21] to solve a digital communication channel equalization. Cha and Kassam [82] used the complex-valued radial basis function network to solve the QAM equalization problem which is considered as a non-linear classification problem. Rajoo Pandey [83] used the complex-valued feed forward neural network for blind equalization with M-ary phase shift keying signals. You and Hong [14] developed an axially symmetric activation function for solving the QAM equalization problem using a complex-valued feed-forward neural network. Deng *et al.* [9], used the sequential learning complex-valued minimal resource allocation network to solve the equalization problem. Li *et al.* [23] used the complex-valued growing and pruning RBF algorithm to solve the equalization of several models, like the Patra model [84], complex-valued linear Chen's model [22], the complex-valued non-linear Cha and Kassam model [82]. As complex-valued neural networks approximate phase more accurately than real-valued neural networks [4, 15, 7], they are more efficient in classifying the complex-valued signals in their quadrants than other real-valued algorithms. In chapter 4 of this book, the different complex-valued neural network based equalizers are used to solve the QAM equalization problem.

1.4.2 Array Signal Processing

The classical problem in array signal processing is to determine the location of an energy radiating planar source relative to the location of the receiver array. Array signal processing comprises of two components viz., Direction of arrival estimation and beam forming. A brief review of the various neural network based antenna array processing is presented by Du *et al.* [85]. As the signals involved in the array signal processing are complex enveloped signals, employing complex-valued neural networks is a wise choice to estimate the direction of arrival and to form the transmitted beam at the receiver.

With the evolution of complex-valued neural networks, they have been employed to solve the array signal processing problems. Bregains and Ares [86] showed that complex-valued neural networks can be incorporated as a very powerful and effective tool in the analysis, synthesis, and diagnostics of antenna arrays. Several works are reported in the literature using complex-valued neural networks for array signal processing applications. For example, Yang *et al.* [87] used the CVNN to solve the array signal processing problem of direction of arrival estimation. Chen *et al.* [75] used a complex-valued RBF network to solve a nonlinear beam forming for multiple-antenna aided communication systems that employ complex-valued quadrature phase shift keying modulation scheme. Chen *et al.* [88] developed a novel complex-valued symmetric radial basis function network based detector, which is capable of approaching the optimal Bayesian performance using channel-impaired training data. They presented a nonlinear beamforming assisted detector for multiple-antenna-aided wireless systems employing complex-valued quadrature phase shift-keying modulation. Similarly, Suksmono and Hirose [89] solved the beamforming problem using the fully complex-valued MLP network. In chapter 4 of this book, we use the fully complex-valued algorithms developed herein to solve the adaptive beam forming problem.

1.4.3 Real-Valued Classification

Nitta [68] showed that the linearly inseparable XOR problem in the Real domain can be easily solved linearly with a single hidden neuron in the Complex domain. As this shows the improved classification ability of the CVNNs, they are also used to solve real-valued classification problems, by phase encoding the real-valued signal, such as the one presented by Amin and Murase [90]. Buchholz and Bihan [91] classified the polarized signals using complex-valued and quaternionic multi-layer perceptron networks. Ozbay *et al.* [92] and Ceylan *et al.* [93] used complex-valued neural networks to classify Doppler signals, which are important in medical applications. On the other hand, Sinha *et al.* [94] used the split complex-valued MLP networks in parallel magnetic resonance image reconstruction which is another important application in medical imaging application.

The multi-layer feed-forward network based on multi-valued neurons has been developed by Aizenberg and Moraga [95]. It is observed that using a traditional architecture of multi-layer feed forward neural network and the high functionality of the multi-valued neurons, it is possible to obtain a new powerful neural network. Its training does not require a derivative of the activation function and its functionality is higher than the functionality of multi-layer feed forward neural network containing the same number of layers and neurons. These advantages of multi-layer feed-forward network based on multi-valued neurons are confirmed by testing using parity n, two spirals, sonar benchmarks and the Mackey-Glass chaotic time series prediction problem. With the introduction of complex-valued multi-valued neurons, Aizenberg *et al.* [96] extended these multi-valued neurons to solve several real-valued classification problems such as in bio-informatics as shown by Aizenberg and Zurada [95], pattern recognition as shown by [97], blur identification as shown by Aizenberg *et al.* [98] etc.

Amin and Murase [90, 99] presented complex-valued neuron models for real-valued classification problems by phase encoding the real-valued inputs. Activation functions that map complex-valued input to real-valued outputs are presented and the gradient descent based learning rules for the activation functions are derived. It is observed that the classification efficiency of such networks are comparable to that of real-valued networks, and the convergence of these networks are faster than real-valued networks.

1.4.4 Memories

As neural networks are good at learning by example, they are used as memories to learn and recall information/signals [1]. As complex-valued neural networks are also capable of learning and recalling, research focus is also on developing efficient complex-valued logic gates and memory. Complex-valued associative memory, which is a complex-valued Hopfield associative memory, was first introduced by Noest [100]. The capacity of the complex-valued associative memory was improved using the pseudo-relaxation algorithm developed by Kobayashi [101]. Muezzinoglu *et al.* [102] introduced a method to store each element of an integral memory set M subset of 1, 2, ..., $K(n)$ as a fixed point into complex-valued multistate Hopfield network. This method employs a set of inequalities to render each memory pattern as a strict local minimum of a quadratic energy landscape. A novel logic gate is suggested by Kawata and Hirose [103]. This logic gate is capable of learning multiple functions at frequencies different from each other, and analyzing the frequency-domain multiplexing ability in the learning based on complex-valued Hebbian rule. Associative memory based on quaternionic Hopfield network are investigated by Isokawa *et al.* [104]. Quaternion is a class of hypercomplex number systems, and the networks used in [104] are composed of quaternionic neurons and the input, output, threshold and connection weights are all represented in quaternions. The concept of associative memories were then extended to the complex domain by Tanaka Gouhei and Aihara Kazuyuki [105] and are used in gray level image reconstruction.

1.4.5 Other Applications

Another important application of the complex-valued neural networks is the estimation of wind profile and wind power by Goh *et al.* [32]. A complex-valued pipelined recurrent neural network architecture is developed, and the network is trained by the complex-valued real-time recurrent learning algorithm with a fully complex-valued activation function to forecast wind signal in its complex form (speed and direction). The complex-valued neural networks are also used in real-valued image processing applications like image recognition as shown by Pande and Goel [106], blur identification as shown by Aizenberg in [98] etc.

In this chapter, a brief survey of the existing literature on complex-valued neural networks has been presented. The CVNNs can be classified based on various parameters like the nature of signals, the type of learning, the mode of learning and the applications in which they were used. These different classes of complex-valued neural networks have been discussed in this chapter.

References

1. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, New Jersey (1998)
2. Remmert, R.: Theory of Complex Functions. Springer, New York (1991)
3. Leung, H., Haykin, S.: The complex backpropagation algorithm. IEEE Transactions on Signal Processing 39(9), 2101–2104 (1991)
4. Kim, T., Adali, T.: Fully complex multi-layer perceptron network for nonlinear signal processing. Journal of VLSI Signal Processing 32(1/2), 29–43 (2002)
5. Yang, S.-S., Ho, C.-L., Siu, S.: Sensitivity analysis of the split-complex valued multilayer perceptron due to the errors of the i.i.d. inputs and weights. IEEE Transactions on Neural Networks 18(5), 1280–1293 (2007)
6. Zhang, H., Zhang, C., Wu, W.: Convergence of batch split-complex backpropagation algorithm for complex-valued neural networks. Discrete Dynamics in Nature and Society 16, Article ID 329173 (2009), Online Journal, `http://www.hindawi.com/journals/ddns/2009/329173.html`
7. Savitha, R., Suresh, S., Sundararajan, N., Saratchandran, P.: A new learning algorithm with logarithmic performance index for complex-valued neural networks. Neurocomputing 72(16-18), 3771–3781 (2009)
8. Jianping, D., Sundararajan, N., Saratchandran, P.: Complex-valued minimal resource allocation network for nonlinear signal processing. International Journal of Neural Systems 10(2), 95–106 (2000)
9. Deng, J.P., Sundararajan, N., Saratchandran, P.: Communication channel equalization using complex-valued minimal radial basis function neural networks. IEEE Transactions on Neural Networks 13(3), 687–696 (2002)
10. Benvenuto, N., Piazza, F.: On the complex backpropagation algorithm. IEEE Transactions on Signal Processing 40(4), 967–969 (1992)
11. Brandwood, D.H.: A complex gradient operator and its application in adaptive array theory. IEE Proceedings 130, 11–16 (1983)
12. Georgiou, G.M., Koutsougeras, C.: Complex domain backpropagation. IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing 39(5), 330–334 (1992)
13. Kim, T., Adali, T.: Approximation by fully complex multi-layer perceptrons. Neural Computation 15(7), 1641–1666 (2003)
14. You, C., Hong, D.: Nonlinear blind equalization schemes using complex-valued multilayer feedforward neural networks. IEEE Transactions on Neural Networks 9(6), 1442–1455 (1998)
15. Savitha, R., Suresh, S., Sundararajan, N., Saratchandran, P.: Complex-valued function approximation using an improved BP learning algorithm for feed-forward networks. In: IEEE International Joint Conference on Neural Networks (IJCNN 2008), June 1-8, pp. 2251–2258 (2008)

16. Li, M.B., Huang, G.-B., Saratchandran, P., Sundararajan, N.: Fully complex extreme learning machine. Neurocomputing 68(1-4), 306–314 (2005)
17. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: a new learning scheme of feedforward neural networks. In: IEEE International Joint Conference on Neural Networks (IJCNN 2004), 25-29, vol. 2, pp. 985–990 (2004)
18. Huang, G.B., Siew, C.K.: Extreme learning machine with randomly assigned RBF kernels. Int. J. Inf. Technol. 11(1) (2005)
19. Kantsila, A., Lehtokangas, M., Saarinen, J.: Complex RPROP-algorithm for neural network equalization of GSM data bursts. Neurocomputing 61, 339–360 (2004)
20. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In: Proceedings of the IEEE International Conference on Neural Networks, vol. 1-3, pp. 586–591 (1993)
21. Chen, S., McLaughlin, S., Mulgrew, B.: Complex valued radial basis function network,part I: Network architecture and learning algorithms. EURASIP Signal Processing Journal 35(1), 19–31 (1994)
22. Chen, S., McLaughlin, S., Mulgrew, B.: Complex valued radial basis function network, part II: Application to digital communications channel equalization. Signal Processing 36(2), 175–188 (1994)
23. Li, M.B., Huang, G.B., Saratchandran, P., Sundararajan, N.: Complex-valued growing and pruning RBF neural networks for communication channel equalisation. IEE Proceedings- Vision, Image and Signal Processing 153(4), 411–418 (2006)
24. Yingwei, L., Sundararajan, N., Saratchandran, P.: A sequential learning scheme for function approximation using minimal radial basis function neural networks. Neural Computation 9(2), 461–478 (1997)
25. Huang, G.B., Saratchandran, P., Sundararajan, N.: A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation. IEEE Transactions on Neural Networks 16(1), 57–67 (2005)
26. Wang, J.: Recurrent neural networks for solving systems of complex-valued linear equations. Electronics Letters 28(18), 1751–1753 (1992)
27. Mandic, D., Chambers, J.: Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability. John Wiley and Sons, West Sussex (2001)
28. Li, C., Liao, X., Yu, J.: Complex-valued recurrent neural network with IIR neuron model: training and applications. Circuits Systems Signal Processing 21(5), 461–471 (2002)
29. Goh, S.L., Mandic, D.P.: An augmented extended kalman filter algorithm for complex-valued recurrent neural networks. Neural Computation 19(4), 1039–1055 (2007)
30. Mandic, D.P.: Complex valued recurrent neural networks for noncircular complex signals. In: International Joint Conference on Neural Networks (IJCNN 2009), June 14-19, pp. 1987–1992 (2009)
31. Zhou, W., Zurada, J.M.: Discrete-time recurrent neural networks with complex-valued linear threshold neurons. IEEE Transactions on Circuits and Systems 56(8), 669–673 (2009)
32. Mandic, D.P., Javidi, S., Goh, S.L., Kuh, A., Aihara, K.: Complex-valued prediction of wind profile using augmented complex statistics. Renewable Energy 34(1), 196–201 (2009)
33. Gangal, A.S., Kalra, P.K., Chauhan, D.S.: Performance evaluation of complex valued neural networks using various error functions. Proceedings of the World Academy of Science, Engineering and Technology 23, 27–32 (2007)
34. Chen, X.M., Tang, Z., Variappan, C., Li, S.S., Okada, T.: A modified error backpropagation algorithm for complex-valued neural networks. International Journal of Neural Systems 15(6), 435–443 (2005)

35. Rattan, S.S.P., Hsieh, W.W.: Complex-valued neural networks for nonlinear complex principal component analysis. Neural Networks 18(1), 61–69 (2005)
36. Fiori, S.: Nonlinear complex-valued extensions of Hebbian learning: an essay. Neural Computation 17(4), 779–838 (2005)
37. Hyvarinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley and Sons, New York (2001)
38. Hyvarinen, A., Oja, E.: Independent component analysis: Algorithms and applications. Neural Networks 13(4-5), 411–430 (2000)
39. Lv, Q., Zhang, X., Jia, Y.: Blind Separation Combined Frequency Invariant Beamforming and ICA for Far-field Broadband Acoustic Signals. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) ISNN 2005. LNCS, vol. 3497, pp. 538–543. Springer, Heidelberg (2005)
40. Chang, A.-C., Jen, C.-W.: Complex-valued ICA utilizing signal-subspace demixing for robust DOA estimation and blind signal separation. Wireless Personal Communications 43(4), 1435–1450 (2007)
41. Lee, I., Kim, T., Lee, T.-W.: Fast fixed-point independent vector analysis algorithms for convolutive blind source separation. Signal Processing 87(8), 1859–1871 (2007)
42. He, Z., Xie, S., Ding, S., Cichocki, A.: Convolutive blind source separation in the frequency domain based on sparse representation. IEEE Transactions on Audio, Speech, and Language Processing 15(5), 1551–1563 (2007)
43. Jen, C.-W., Chen, S.-W., Chang, A.-C.: High-resolution DOA estimation based on independent noise component for correlated signal sources. Neural Computing and Applications 18(4), 381–385 (2008)
44. Calhoun, V.D., Adali, T., Pearlson, G.D., van Zijl, P.C., Pekar, J.J.: Independent component analysis of fMRI data in the complex domain. Magnetic Resonance in Medicine 48(1), 180–192 (2002)
45. Calhoun, V., Adali, T.: Complex infomax: Convergence and approximation of infomax with complex nonlinearities. The Journal of VLSI Signal Processing 44(1-2), 173–190 (2006)
46. Adali, T., Calhoun, V.D.: Complex ICA of brain imaging data. IEEE Signal Processing Magazine 24(5), 136–139 (2007)
47. Bingham, E., Hyvarinen, A.: Ica of complex valued signals: A fast and robust deflationary algorithm. In: International Joint Conference on Neural Networks (IJCNN 2000), vol. 3 (2000)
48. Bingham, E., Hyvarinen, A.: A fast fixed-point algorithm for independent component analysis of complex valued signals. International Journal of Neural Systems 10(1), 1–8 (2000)
49. Fiori, S.: Neural independent component analysis by maximum-mismatch learning principle. Neural Networks 16(8), 1201–1221 (2003)
50. Yang, T., Mikhael, W.B.: A general approach for image and co-channel interference suppression in diversity wireless receivers employing ICA. Circuits, Systems, and Signal Processing 23(4), 317–327 (2004)
51. Erikkson, J., Koivunen, V.: Complex random vectors and ICA models: Identifiability, uniqueness, and separability. IEEE Transactions on Information Theory 52(3), 1017–1029 (2006)
52. Sallberg, B., Grbic, N., Claesson, I.: Complex-valued independent component analysis for online blind speech extraction. IEEE Transactions on Audio, Speech, and Language Processing 16(8), 1624–1632 (2008)
53. Li, H., Adali, T.: A class of complex ICA algorithms based on the kurtosis cost function. IEEE Transactions on Neural Networks 19(3), 408–420 (2008)

54. Novey, M., Adali, T.: Complex ICA by negentropy maximization. IEEE Transactions on Neural Networks 19(4), 596–609 (2008)
55. Li, X.-L., Adali, T.: Complex independent component analysis by entropy bound minimization. IEEE Transactions on Circuits and Systems I 57(7), 1417–1430 (2010)
56. Ollilaa, E., Koivunen, V.: Complex ICA using generalized uncorrelating transform. Signal Processing 89(4), 365–377 (2009)
57. Novey, M., Adali, T.: On extending the complex fast ICA algorithm to noncircular sources. IEEE Transactions on Signal Processing 56(5), 2148–2154 (2008)
58. Brown, J., Churchill, R.: Complex Variables and Applications. McGrawHill, New York (1996)
59. Flanigan, F.: Complex Variables: Harmonic and Analytic Functions. Dover Publications, New York (1983)
60. Le Page, W.: Complex Variables and the Laplace Transforms for Engineers. Dover Publications, New York (1980)
61. Fisher, S.: Complex Variables, 2nd edn. Dover Publications, New York (1999)
62. Wirtinger, W.: Zur formalen theorie der funktionen von mehr komplexen vernderlichen. Annals of Mathematics 97 (1927)
63. Hjorungnes, A., Gesbert, D.: Complex-valued matrix differentiation: Techniques and key results. IEEE Transactions on Signal Processing 55(6), 2740–2746 (2007)
64. Adali, T., Li, H., Novey, M., Cardoso, J.-F.: Complex ICA using nonlinear functions. IEEE Transactions on Signal Processing 56(9), 4536–4544 (2008)
65. Loss, D.V., de Castro, M.C.F., Franco, P.R.G., de Castro, E.C.C.: Phase transmittance RBF neural networks. Electronics Letters 43(16), 882–884 (2007)
66. Uncini, A., Vecci, L., Campolucci, P., Piazza, F.: Complex-valued neural networks with adaptive spline activation function for digital radio links nonlinear equalization. IEEE Transactions on Signal Processing 47(2), 505–514 (1999)
67. Yang, S.S., Siu, S., Ho, C.L.: Analysis of the initial values in split-complex backpropagation algorithm. IEEE Transactions on Neural Networks 19(9), 1564–1573 (2008)
68. Nitta, T.: An extension of the back-propagation algorithm to complex numbers. Neural Networks 10(8), 1391–1415 (1997)
69. Hirose, A.: Continuous complex-valued back-propagation learning. Electronic Letters 28(20), 1854–1855 (1992)
70. Kim, M.S., Guest, C.C.: Modification of back propagation networks for complex-valued signal processing in frequency domain. In: International Joint Conference on Neural Networks (IJCNN 1990), vol. 3, pp. 27–31 (1990)
71. Karim, A., Adeli, H.: Comparison of the fuzzy-wavelet RBFNN freeway incident detection model with the california algorithm. Journal of Transportation Engineering 128(1), 21–30 (2002)
72. Jogensen, T.D., Haynes, B.P., Norlund, C.C.F.: Pruning artificial neural networks using neural complexity measures. International Journal of Neural Systems 18(5), 389–403 (2008)
73. Mayorga, R.V., Carrera, J.: A radial basis function network approach for the computational of inverse continuous time variant functions. International Journal of Neural Systems 17(3), 149–160 (2007)
74. Pedrycz, W., Rai, P., Zurada, J.: Experience-consistent modeling for radial basis function neural networks. International Journal of Neural Systems 18(4), 279–292 (2008)
75. Chen, S., Hong, X., Harris, C.J., Hanzo, L.: Fully complex-valued radial basis function networks: Orthogonal least squares regression and classification. Neurocomputing 71(16-18), 3421–3433 (2008)

76. Chen, S.: Information Science Reference. Complex-valued Neural Networks: Utilizing High-dimensional Parameters, ch. VII. IGI Global snippet, PA (2009)
77. Savitha, R., Suresh, S., Sundararajan, N.: A fully complex-valued radial basis function network and its learning algorithm. International Journal of Neural Systems 19(4), 253–267 (2009)
78. Huang, G.B., Li, M.B., Chen, L., Siew, C.K.: Incremental extreme learning machine with fully complex hidden nodes. Neurocomputing 71(4-6), 576–583 (2008)
79. Suresh, S., Savitha, R., Sundararajan, N.: A sequential learning algorithm for a complex-valued self-regulatory resouce allocation network-csran. IEEE Transactions on Neural Networks 22(7), 1061–1072 (2011)
80. Hirose, A.: Complex-valued neural networks for more fertile electronics. Journal of the Institute of Electronics, Information and Communication Engineers (IEICE) 87(6), 447–449 (2004)
81. Hirose, A.: Complex-valued Neural Networks: Theories and Applications. Series on Innovative Intelligence, vol. 5. World Scientific Publishing Company, Singapore (2004)
82. Cha, I., Kassam, S.A.: Channel equalization using adaptive complex radial basis function networks. IEEE Journal on Selected Areas in Communications 13(1), 122–131 (1995)
83. Pandey, R.: Feedforward neural network for blind equalization with PSK signals. Neural Computing and Applications 14(4), 290–298 (2005)
84. Patra, J.C., Pal, R.N., Baliarsingh, R., Panda, G.: Nonlinear channel equalization for QAM constellation using artificial neural networks. IEEE Transactions on System, Man and Cybernetics, Part B: Cybernetics 29(2), 262–271 (1999)
85. Du, K.L., Lai, A.K.Y., Cheng, K.K.M., Swamy, M.N.S.: Neural methods for antenna array signal processing: A review. Signal Processing 82(4), 547–561 (2002)
86. Bregains, J.C., Ares, F.: Analysis, synthesis and diagnosis of antenna arrays through complex-valued neural networks. Microwave and Optical Technology Letters 48(8), 1512–1515 (2006)
87. Yang, W.H., Chan, K.K., Chang, P.R.: Complex-valued neural network for direction of arrival estimation. Electronics Letters 30(7), 574–575 (1994)
88. Shen, C., Lajos, H., Tan, S.: Symmetric complex-calued RBF receiver for multiple-antenna-aided wireless systems. IEEE Transactions on Neural Networks 19(9), 1659–1665 (2008)
89. Suksmono, A.B., Hirose, A.: Intelligent beamforming by using a complex-valued neural network. Journal of Intelligent and Fuzzy Systems 15(3-4), 139–147 (2004)
90. Amin, M.F., Murase, K.: Single-layered complex-valued neural network for real-valued classification problems. Neurocomputing 72(4-6), 945–955 (2009)
91. Buchholz, S., Bihan, N.L.: Polarized signal classification by complex and quaternionic multi-layer perceptron. International Journal of Neural Systems 18(2), 75–85 (2008)
92. Ozbay, Y., Kara, S., Latifoglu, F., Ceylan, R., Ceylan, M.: Complex-valued wavelet artificial neural network for doppler signals classifying. Artificial Intelligence in Medicine 40(2), 143–156 (2007)
93. Ceylan, M., Ceylan, R., Ozbay, Y., Kara, S.: Application of complex discrete wavelet transform in classification of doppler signals using complex-valued artificial neural network. Artificial Intelligence in Medicine 44(1), 65–76 (2008)
94. Sinha, N., Saranathan, M., Ramakrishna, K.R., Suresh, S.: Parallel magnetic resonance imaging using neural networks. In: IEEE International Conference on Image Processing (ICIP 2007), vol. 3, pp. 149–152 (2007)
95. Aizenberg, I., Moraga, C.: Multilayer feedforward neural network based on multi-valued neurons (MLMVN) and a backpropagation learning algorithm. Soft Computing 11(2), 169–183 (2007)

96. Aizenberg, I., Moraga, C., Paliy, D.: A feedforward neural network based on multi-valued neurons. In: Computational Intelligence, Theory and Applications. Advances in Soft Computing, XIV, pp. 599–612. Springer, Berlin (2005)
97. Aizenberg, I., Aizenberg, N.: Pattern Recognition Using Neural Network Based on Multi-Valued Neurons. In: Mira, J. (ed.) IWANN 1999. LNCS, vol. 1607, pp. 383–392. Springer, Heidelberg (1999)
98. Aizenberg, I., Paliy, D.V., Zurada, J.M., Astola, J.T.: Blur identification by multi-layer neural network based on multivalued neurons. IEEE Transactions on Neural Networks 19(5), 883–898 (2008)
99. Amin, M.F., Islam, M.M., Murase, K.: Ensemble of single-layered complex-valued neural networks for classification tasks. Neurocomputing 72(10-12), 2227–2234 (2009)
100. Noest, A.J.: Phasor neural networks. Neural Information Processing Systems 2, 584–591 (1989), Online Journal, http://books.nips.cc/nips02.html
101. Kobayashi, M.: Pseudo-relaxation learning algorithm for complex-valued associative memory. International Journal of Neural Systems 18(2), 147–156 (2008)
102. Muezzinoglu, M.K., Guzelis, C., Zurada, J.M.: A new design method for the complex-valued multistate Hopfield associative memory. IEEE Transactions on Neural Networks 14(4), 891–899 (2003)
103. Kawata, S., Hirose, A.: Frequency-multiplexing ability of complex-valued Hebbian learning in logic gates. International Journal of Neural Systems 18(2), 173–184 (2008)
104. Isokawa, T., Nishimura, H., Kamiura, N., Matsui, N.: Associative memory in quaternionic Hopfield neural network. International Journal of Neural Systems 18(2), 135–145 (2008)
105. Tanaka, G., Aihara, K.: Complex-valued multistate associative memory with nonlinear multilevel functions for gray-level image reconstruction. IEEE Transactions on Neural Networks 20(9), 1463–1473 (2009)
106. Pande, A., Goel, V.: Complex-valued neural network in image recognition: A study on the effectiveness of radial basis function. Proceedings of World Academy of Science, Engineering and Technology 20, 220–225 (2007)