

## Metacognitive Learning in a Fully Complex-Valued Radial Basis Function Neural Network

**R. Savitha**

*savi0001@ntu.edu.sg*

**S. Suresh**

*ssundaram@ntu.edu.sg*

*School of Computer Engineering, Nanyang Technological University,  
639798 Singapore*

**N. Sundararajan**

*ensundara@ntu.edu.sg*

*School of Electrical and Electronics Engineering, Nanyang Technological University,  
639735 Singapore*

Recent studies on human learning reveal that self-regulated learning in a metacognitive framework is the best strategy for efficient learning. As the machine learning algorithms are inspired by the principles of human learning, one needs to incorporate the concept of metacognition to develop efficient machine learning algorithms. In this letter we present a metacognitive learning framework that controls the learning process of a fully complex-valued radial basis function network and is referred to as a metacognitive fully complex-valued radial basis function (Mc-FCRBF) network. Mc-FCRBF has two components: a cognitive component containing the FC-RBF network and a metacognitive component, which regulates the learning process of FC-RBF. In every epoch, when a sample is presented to Mc-FCRBF, the metacognitive component decides what to learn, when to learn, and how to learn based on the knowledge acquired by the FC-RBF network and the new information contained in the sample. The Mc-FCRBF learning algorithm is described in detail, and both its approximation and classification abilities are evaluated using a set of benchmark and practical problems. Performance results indicate the superior approximation and classification performance of Mc-FCRBF compared to existing methods in the literature.

### 1 Introduction ---

Research on complex-valued neural networks is gaining widespread interest (Chen, McLaughlin, & Mulgrew, 1994; Kim & Adali, 2002; Goh & Mandic, 2004, 2007a, 2007b; Fiori, 2005; Rattan & Hsieh, 2005), with a growing need for intelligent and adaptive mechanisms to cater to the needs of

telecommunications (Li, Huang, Saratchandran, & Sundararajan, 2006; Bre-gains & Ares, 2006; Savitha, Vigneshwaran, Suresh, & Sundararajan, 2009; Shen, Lajos, & Tan, 2008; Jianping, Sundararajan, & Saratchandran, 2000), and medical imaging (Sinha, Saranathan, Ramakrishna, & Suresh, 2007), involving and operating on complex-valued signals naturally. In addition to these applications, recent findings reveal the exceptional decision-making ability of complex-valued neural networks compared to real-valued neural networks in performing real-valued classification tasks (Nitta, 2003). However, operating in the complex domain presents new challenges in the development of efficient complex-valued neural networks capable of carrying out both accurate magnitude and phase approximation and classification.

Liouville's theorem is the main bottleneck that inhibits efficient use of the complex domain. Basically a neural network requires an activation function to be nonlinear, bounded, and differentiable in every point on the considered plane (Haykins, 1999). This implies that in the complex domain, the function has to be nonlinear, bounded, and entire. But Liouville's theorem states that a bounded and entire function is a constant function in the complex domain (Remmert, 1991). This limits the choices for a complex-valued activation function. However, this limitation has been overcome by reducing the essential properties needed for an activation function to be entire and bounded to that of analytic and bounded almost everywhere by Kim & Adali (2002, 2003). Kim & Adali (2002) also suggested a set of elementary transcendental functions as possible choices of activation functions for a fully complex-valued multilayer perceptron (FC-MLP) and derived its fully complex-valued gradient descent-based learning algorithm.

Complex-valued radial basis function (CRBF) networks and their batch and sequential learning algorithms, namely, CRBF by Chen et al. (1994), complex-valued minimal resource allocation network (CMRAN) by Jian-ping et al. (2000), and complex-valued growing and pruning RBF (CGAP-RBF) by Li et al. (2006), for example, use a real-valued gaussian activation function that maps  $\mathbb{C} \rightarrow \mathbb{R}$  in the hidden layer. This results in real-valued hidden-layer responses in the forward computation and splits the complex-valued error into its real and imaginary parts during backward computation. Therefore, the gradients used for parameter updates are not fully complex valued. Hence, the resultant network does not produce accurate phase approximation due to the loss of correlation between the real and imaginary parts of the error.

To enable accurate phase approximation in the framework of complex-valued radial basis function networks, a fully complex-valued radial basis function (FC-RBF) network was recently proposed by Savitha, Suresh, and Sundararajan (2009) and Savitha, Suresh, Sundararajan, and Kim (2012). FC-RBF uses a fully complex-valued gaussian-like symmetric activation function, *sech*, that is analytic and bounded almost everywhere. Because the phase information is preserved during the forward and backward computations, FC-RBF provides a better approximation of both magnitude and phase. Recently a sequential self-regulatory learning algorithm referred to

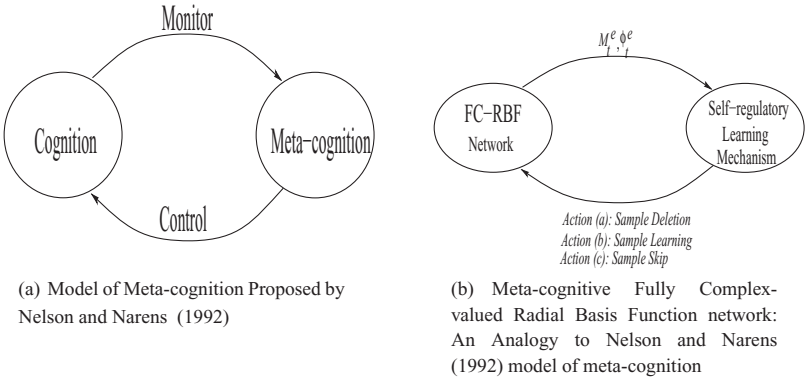


Figure 1: Analogy between the Nelson and Narens (1992) model of metacognition and Mc-FCRBF.

as a complex-valued self-regulatory resource allocation network (CSRAN) was developed using this activation function by Suresh, Savitha, and Sundararajan (2011). All the complex-valued learning algorithms address the issue of the how to learn the functional relationship between the input features and their targets. They do not address the issues of what samples are to be learned and when to use the samples for learning (Savitha, Suresh, & Sundararajan, 2010).

Recent work on human learning by Wenden (1998), Rivers (2001), and Isaacson and Fujita (2006) suggest that metacognition, which empowers learners with a self-regulated learning mechanism, is the best learning strategy. Metacognition provides a means to accurately assess one's current knowledge, identify when new knowledge is needed and provide strategies to acquire that new knowledge (Josyula, Vadali, Donahue, & Hughes, 2009). In this letter, we introduce one such metacognitive learning algorithm for the FC-RBF network. FC-RBF network with the metacognitive learning algorithm is referred to as a metacognitive fully complex-valued radial basis function (Mc-FCRBF) network.

We first briefly explain the concept of metacognition proposed by Nelson and Narens (1992) in the context of human learning. They proposed a simple model of metacognition, shown in Figure 1a. This model has two components: a cognitive and a metacognitive component. The cognitive component represents the knowledge, and the metacognitive component has a dynamic model of the cognitive component, a mental simulation of the cognitive component (Nelson & Narens, 1992). The information flow from the cognitive component to the metacognitive component is considered a monitor signal, while the information flow in the reverse direction is considered a control signal. In particular, the information flowing from the metacognitive component to the cognitive component (control) changes either the state of the cognitive component or the cognitive component

itself. As a result, one of three actions could occur at the cognitive component: initiate an action, continue an action, or terminate an action. However, because the control signal does not yield any information from the cognitive component, a monitoring signal is needed. The basic notion of monitoring is that the metacognitive component is informed about the cognitive component. This changes the state of the metacognitive component's model of the cognitive component, including no change in state. It must be noted that the monitoring signal is logically independent of the control signal.

Similar to the Nelson and Narens (1992) model of metacognition, the metacognitive FC-RBF also has two components, as shown in Figure 1b: the FC-RBF network is the cognitive component of Mc-FCRBF and a self-regulatory learning mechanism is its metacognitive component. The self-regulatory learning mechanism has a dynamic model of an FC-RBF network and controls its learning ability by deciding what to learn, when to learn, and how to learn. As a result, when a training sample is presented, one of the following actions occurs in the FC-RBF network: sample deletion, sample learning, or sample skip. Thus, during the entire training process, the self-regulatory learning mechanism enables selective participation of samples in the training process. Moreover, as each sample is presented, the self-regulatory learning mechanism is informed of the current state of the FC-RBF network (monitor) through the instantaneous magnitude error and phase error of the sample.

The self-regulatory learning mechanism controls the learning process of the FC-RBF network to enable the samples with higher information content to be learned first and samples with lower information content to be learned during the later stages of the training process. Samples with similar information content are deleted during the training process. Thus, the metacognitive component of Mc-FCRBF prevents learning similar samples in every epoch of the batch learning process, thereby avoiding overtraining and improving the generalization performance of FC-RBF network.

In this letter, we first describe the working principle of Mc-FCRBF in detail using a synthetic complex-valued function approximation problem defined by Savitha, Suresh, Sundararajan, and Saratchandran (2009) along with the evaluation of its function approximation performance. We also study the performance of Mc-FCRBF on a quadrature amplitude modulation (QAM) problem with circular signals and an adaptive beam-forming problem with noncircular signals.<sup>1</sup> In all these problems, the approximation ability of Mc-FCRBF is compared to other existing complex-valued batch learning algorithms like FC-MLP by Kim and Adali (2002), CRBF by Chen

---

<sup>1</sup>The circularity property (or properness) is an important feature of many complex random signals. At the complex signal level, this property means that the signal is statistically uncorrelated with its own complex conjugate. In case of complex random vectors, this property implies that the so-called complementary covariance matrix or the pseudocovariance matrix vanishes.

et al. (1994), complex-valued extreme learning machine (C-ELM) by Li, Huang, Saratchandran, and Sundararajan (2005), and FC-RBF network by Savitha, Suresh, and Sundararajan (2009).

Next, we investigate the classification ability of Mc-FCRBF using a set of benchmark real-valued classification problems from the University of California, Irvine machine learning repository by Blake and Merz (1998) and a practical acoustic emission signal classification problem given in Omkar, Suresh, Raghavendra, and Mani (2002). We first analytically prove that the FC-RBF network has two decision boundaries that are orthogonal to each other and help FC-RBF solve classification tasks efficiently. Since the learning conditions of Mc-FCRBF are derived based on approximation measures, they may not approximate the decision surface efficiently as Suresh, Sundararajan, and Saratchandran (2008a) explained. Moreover, recently, by Zhang (2003) and Suresh, Sundararajan, and Saratchandran (2008b) showed that in classification problems, use of a hinge error function can help in estimating the posterior probabilities more accurately than a mean-squared-error function. Hence, we next propose a class-specific learning condition for Mc-FCRBF and the hinge error function to solve classification problems using Mc-FCRBF. Finally, we evaluate the classification ability of Mc-FCRBF in comparison with other complex-valued and best-performing real-valued classifiers available in the literature. The results show the advantages of the metacognitive component of Mc-FCRBF and its better classification ability.

The letter is organized as follows. In section 2, we present a detailed description of the metacognitive fully complex-valued radial basis function network. The section also briefly discusses the FC-RBF network and its gradient descent learning algorithm. In section 3, we demonstrate the working principle of Mc-FCRBF using a synthetic complex-valued function approximation problem. Next, we evaluate the function approximation performance of Mc-FCRBF in comparison to other complex-valued learning algorithms on the above problem and also on two real-world practical problems. In section 4, we demonstrate the decision-making ability of Mc-FCRBF and evaluate its performance on a set of benchmark and practical real-valued classification problems. Finally, section 5 presents our conclusions.

## 2 A Metacognitive Fully Complex-Valued Radial Basis Function Network

---

Mc-FCRBF has two components, as shown in Figure 1b: a cognitive component and a metacognitive component. In this section, we describe these two components in detail and summarize its learning algorithm. First, we define the approximation and classification problems in the complex domain. Next, we review FC-RBF by Savitha, Suresh, and Sundararajan (2009), which is the cognitive component of Mc-FCRBF. Finally, we describe the functioning of the metacognitive component of Mc-FCRBF.

**2.1 Problem Definition.** Given a training data set  $\{(\mathbf{z}^1, \mathbf{y}^1), \dots, (\mathbf{z}^N, \mathbf{y}^N)\}$ , where  $N$  is the total number of training samples,  $\mathbf{z}^t = [z_1^t, \dots, z_m^t]^T \in \mathbb{C}^m$  is the complex-valued inputs, and  $\mathbf{y}^t = [y_1^t, \dots, y_n^t]^T \in \mathbb{C}^n$  is the complex-valued target outputs. The aim of the FC-RBF network is to approximate the relationship between the inputs and their respective target values. In a function approximation problem,  $\mathbf{z}^t \in \mathbb{C}^m$  is the  $m$ -dimensional complex-valued input vector, and  $\mathbf{y}^t \in \mathbb{C}^n$  is its desired target (i.e., the functional value for the given input  $\mathbf{z}^t$ ). In classification problems, the  $\mathbf{z}^t \in \mathbb{C}^m$  is the  $m$ -dimensional transformed complex-valued input feature vector and the target  $\mathbf{y}^t \in \mathbb{C}^n$  is the  $n$ -dimensional complex-valued coded class label obtained from the actual class label ( $c^t$ ) using:

$$y_j^t = \begin{cases} 1 + i1 & \text{if } c^t = j \\ -1 - i1 & \text{otherwise} \end{cases} \quad j = 1, 2, \dots, n; t = 1, 2, \dots, N, \quad (2.1)$$

where  $n$  is the total number of classes and  $i$  is the complex operator.

## 2.2 Cognitive Component of Mc-FCRBF: A Brief Review of FC-RBF.

The architecture of the FC-RBF network with the *sech* activation function in the hidden layer is shown in Figure 2. The network has  $m$  input neurons  $K$  hidden neurons, and  $n$  output neurons. The neurons in the input and output layers employ a linear activation function, and those in the hidden layer employ the fully complex-valued gaussian-like *sech* activation function ( $\text{sech}(z) = 2/(\exp(z) + \exp(-z))$ ).

The response of the  $k$ th hidden neuron ( $h_k^t$ ) with the *sech* activation function for an input  $\mathbf{z}^t \in \mathbb{C}^m$  is given by:

$$h_k^t = \text{sech}[\mathbf{v}_k^T (\mathbf{z}^t - \mathbf{u}_k)]; k = 1, 2, \dots, K, \quad (2.2)$$

where  $\mathbf{u}_k = [u_k^1, \dots, u_k^m]^T \in \mathbb{C}^m$  is the center of the  $k$ th hidden neuron,  $\mathbf{v}_k = [v_k^1, \dots, v_k^m]^T \in \mathbb{C}^m$  is the scaling factor of the  $k$ th hidden neuron, and the superscript  $T$  is the transpose operator.

The predicted output of the  $j$ th output neuron ( $\hat{y}_j^t$ ) for an input  $\mathbf{z}^t$  is given by:

$$\hat{y}_j^t = \sum_{k=1}^K w_{jk} h_k^t; j = 1, 2, \dots, n, \quad (2.3)$$

The residual error ( $\mathbf{e}^t$ ) of the network is given by

$$\mathbf{e}^t = \mathbf{y}^t - \hat{\mathbf{y}}^t, \quad (2.4)$$

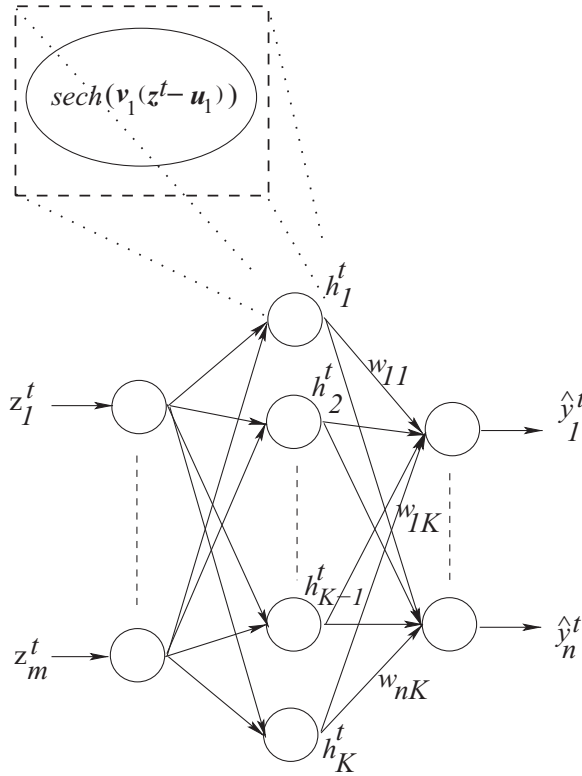


Figure 2: Architecture of the FC-RBF network.

and the mean squared error function ( $E$ ) is given by

$$E = \frac{1}{2} \mathbf{e}^t \mathbf{e}^{tH}, \quad (2.5)$$

where the superscript  $H$  represents the complex Hermitian operator.

The objective of FC-RBF learning algorithm is to estimate the following network parameters: the hidden neuron centers ( $\mathbf{u}_k$ ), the scaling factor of the hidden neurons ( $\mathbf{v}_k$ ), and the output weights ( $\mathbf{w}_k$ ), such that the error function (see equation 2.5) is minimized for all the training samples. The derivation of FC-RBF learning algorithm requires the partial derivative of both the real-valued error function and the complex-valued activation function. Using Cauchy Riemann equations, the derivation of the gradient of a complex-valued activation function that is analytic and bounded almost everywhere has been developed by Savitha, Suresh, and Sundararajan (2009).

But the error function shown in equation 2.5 is a real-valued function of the complex-valued error. Hence, this error function does not satisfy the Cauchy Riemann equations and does not have a well-defined complex-valued gradient based on the Cauchy Riemann equations defined by Remmert (1991). Therefore, the gradient descent-based learning algorithm of FC-RBF has been derived in Savitha, Suresh, and Sundararajan (2009) starting from the first principles of partial derivatives, and this derivation is very cumbersome. Hence, in this letter, we use the  $\mathbb{C} - \mathbb{R}$  calculus introduced by Wirtinger (1927) and Brandwood (1983), which has well-defined complex-valued derivatives for both the holomorphic functions (activation function) and nonholomorphic functions (error function) to obtain the complex-valued gradients of both the error function and the activation function<sup>2</sup>. Thus, the complexity of the learning algorithm presented in Savitha, Suresh, and Sundararajan (2009) by using a cumbersome derivation procedure is significantly simplified in this letter.

Let  $f_R(z_c, \bar{z}_c)$  be a real-valued function of a complex-valued variable  $z_c = x_r + iy_r$  and its complex-valued conjugate  $\bar{z}_c = x_r - iy_r$ . Then the following pair of derivatives are defined by the  $\mathbb{C} - \mathbb{R}$  calculus:

$$\mathbb{R}\text{-derivative of } f_R(z_c, \bar{z}_c) = \frac{\partial f_R}{\partial z_c} \Big|_{\bar{z}_c=\text{constant}}, \quad (2.6)$$

$$\overline{\mathbb{R}}\text{-derivative of } f_R(z_c, \bar{z}_c) = \frac{\partial f_R}{\partial \bar{z}_c} \Big|_{z_c=\text{constant}}. \quad (2.7)$$

Remmert (1991) proved that the  $\mathbb{R} -$  derivative, equation 2.6, and the  $\overline{\mathbb{R}} -$  derivative, equation 2.7, can be equivalently written as

$$\begin{aligned} \frac{\partial f_R}{\partial z_c} &= \frac{1}{2} \left( \frac{\partial f_R}{\partial x_r} - i \frac{\partial f_R}{\partial y_r} \right), \\ \frac{\partial f_R}{\partial \bar{z}_c} &= \frac{1}{2} \left( \frac{\partial f_R}{\partial x_r} + i \frac{\partial f_R}{\partial y_r} \right), \end{aligned} \quad (2.8)$$

where the partial derivatives with respect to  $x_r$  and  $y_r$  are true partial derivatives of the function  $f_R(z_c) = f_R(x_r, y_r)$ , which is differentiable with respect to  $x_r$  and  $y_r$ .

Using these definitions of partial derivatives, we can derive the gradient update rule for the free parameters of the network in a straightforward manner as

$$\Delta \mathbf{w}_k = \eta_w \bar{h}_k \mathbf{e}^t; k = 1, 2, \dots, K, \quad (2.9)$$

---

<sup>2</sup>A holomorphic function is a complex-valued function of one or more complex variables that is complex differentiable in a neighborhood of every point in its domain.



$$\Delta \mathbf{v}_k = \eta_v \bar{\mathbf{w}}_k^T \mathbf{e}^t \overline{f'}(\mathbf{v}_k^T(\mathbf{z}^t - \mathbf{u}_k))(\overline{\mathbf{z}^t - \mathbf{u}_k}), \quad (2.10)$$

$$\Delta \mathbf{c}_k = -\eta_c \bar{\mathbf{w}}_k^T \mathbf{e}^t \overline{f'}(\mathbf{v}_k^T(\mathbf{z}^t - \mathbf{u}_k))\bar{\mathbf{v}}_k. \quad (2.11)$$

It can be observed from equations 2.9, 2.10 and 2.11 that the gradients are the same as those derived in Savitha, Suresh, and Sundararajan (2009), although the derivation procedure for the gradients has been significantly simplified by the use of  $\mathbb{C} - \mathbb{R}$  calculus.

**2.3 Metacognitive Component of Mc-FCRBF: The Self-Regulatory Learning Mechanism.** In this section, we describe the working principles of the metacognitive component of Mc-FCRBF. As shown in Figure 1a and 1b, the metacognitive component of the Mc-FCRBF controls the learning ability of the FC-RBF network (cognitive component) by selecting suitable learning strategies for each sample (control signal) in each epoch of the training process. The instantaneous magnitude and phase errors based on the residual error of FC-RBF network, equation 2.4, for each sample act as the monitory signal (information flow from cognitive to metacognitive component). They are defined by

- The instantaneous magnitude error,

$$M_t^e = \frac{1}{n} \sqrt{\mathbf{e}^t \mathbf{e}^{tH}} \quad (2.12)$$

- The instantaneous phase error:

$$\phi_t^e = \frac{1}{n} \sum_{l=1}^n |\arg(y_l^t \text{con } j(\hat{y}_l^t))| \quad (2.13)$$

where the function  $\arg(\cdot)$  returns the phase of a complex-valued number in  $[-\pi, \pi]$ , and is given by

$$\arg(z) = \text{atan}\left(\frac{\text{imag}(z)}{\text{real}(z)}\right). \quad (2.14)$$

When a sample is presented to the FC-RBF network, the metacognitive component decides what to learn, when to learn, and how to learn by taking one of the following three actions (control signals):

*Action a: Sample deletion:* Delete those samples from the training data set that contain information similar to that already learned by the network. This action addresses the what to learn component of metacognition.

*Action b: Sample learning:* Use the sample to update the network parameters in the current epoch. This represents how to learn the sample in the metacognitive framework.

*Action c: Sample skip:* Skip the sample from learning in the current epoch and retain the sample in the training data set, thereby deciding when to learn the sample in the context of metacognition.

These three actions of the metacognitive learning are described in detail below:

- *Sample deletion:* If  $M_t^e < E_d^M$  and  $\phi_t^e < E_d^\phi$ , where  $E_d^M$  is the delete magnitude threshold and  $E_d^\phi$  is the delete phase threshold, the sample  $t$  is deleted from the training data set. The thresholds  $E_d^M$  and  $E_d^\phi$  are chosen based on the desired accuracy.
- *Sample learning:* If the sample learning condition given by

$$\text{If } M_t^e \geq E_l^M \text{ or } \phi_t^e \geq E_l^\phi \quad (2.15)$$

is satisfied in the current epoch, then the parameters of the network are updated using the gradient descent-based parameter update rules (given in equations 2.10, 2.11, and 2.9) in the current epoch only. Here,  $E_l^M$  is the parameter update magnitude threshold and  $E_l^\phi$  is the parameter update phase threshold. It must be noted that the parameter update magnitude threshold ( $E_l^M$ ) and parameter update phase threshold ( $E_l^\phi$ ) are not fixed. They are self-regulated based on the residual error of the sample in the current epoch according to the following conditions:

$$\text{if } M_t^e \geq E_l^M, \text{ then } E_l^M := \delta E_l^M - (1 - \delta)M_t^e, \quad (2.16)$$

$$\text{if } \phi_t^e \geq E_l^\phi, \text{ then } E_l^\phi := \delta E_l^\phi - (1 - \delta)\phi_t^e, \quad (2.17)$$

where  $\delta$  is the slope at which the thresholds are self-regulated. A larger value of  $\delta$  results in a slow decay of the thresholds from their initial values. This helps fewer samples with significant information to be learned first and samples containing less significant information to be learned last. Therefore, larger values of  $\delta$  ensure that the metacognitive principles are emulated efficiently. Usually  $\delta$  is set close to 1. Note that the sample learning condition defined in equation 2.15 is based on the approximation measures. Suresh et al. (2008a) have shown that algorithms developed using approximation measures may not perform well in classification problems. Hence, one needs to include classification measures like misclassification, and hinge error in the sample learning condition while solving classification problems. More details on this are presented in section 4.2.

- *Sample skip:* If a sample does not satisfy the sample deletion or sample learning condition in the current epoch, the sample is skipped in the current epoch and retained in the training data set as such. Due to the self-regulating nature of the parameter update thresholds, the sample might be used in learning in subsequent epochs.

The learning algorithm of Mc-FCRBF is summarized in pseudocode 1.

**Pseudocode 1:** A Metacognitive Fully Complex-Valued Radial Basis Function Network.

*Input:* The data set  $\{(\mathbf{z}^1, \mathbf{y}^1), (\mathbf{z}^2, \mathbf{y}^2), \dots, (\mathbf{z}^N, \mathbf{y}^N)\}$   
of the function to be approximated.

*Output:* Parameters of the network:  $\mathbf{w}_j$ ,  $\mathbf{u}_k$  and  $\mathbf{v}_k$ .

**START**

*Initialization:*

Choose the number of hidden neurons.

Initialize  $\mathbf{u}_k$ ,  $\mathbf{v}_k$  and  $\mathbf{w}_j$ ;  $k = 1, 2, \dots, K$ ,  $j = 1, 2, \dots, n$ .

Initialize the number of epochs OR

Specify the errorbased stopping criterion.

**WHILE STOPPING CRITERION**(Epoch)

**FOR**  $t = 1, 2, \dots, N$ (Sample)

Compute the network output using equation 2.3.

Compute the instantaneous magnitude error ( $M_t^e$ )

using equations 2.12 and phase error ( $\phi_t^e$ ) using  
equations 2.13.

**IF**  $M_t^e < E_d^M$  AND  $\phi_t^e < E_d^\phi$  **THEN**

Delete the sample from the training data set.

**ELSEIF**  $M_t^e \geq E_l^M$  OR  $\phi_t^e \geq E_l^\phi$  **THEN**

Update the network parameters

using equations 2.9, 2.10, and 2.11.

The thresholds  $E_l^M$  and  $E_l^\phi$  are self-regulated  
according to equations 2.16 and 2.17 respectively.

**ELSE**

Reserve the sample in the training data set.

It may be used in learning/deleted in the  
subsequent epochs.

**ENDIF**

**END FOR**(Sample)

**END WHILE**(Epoch)

**STOP**

### 3 Performance Evaluation of Mc-FCRBF for Function Approximation Problems

---

In this section, we first explain the working principles of the metacognitive components of Mc-FCRBF described in section 2.3 using a synthetic complex-valued function approximation problem (CFAP) (Savitha, Suresh, Sundararajan, & Saratchandran, 2009). Next, we evaluate the function approximation performance of Mc-FCRBF on two real-world problems consisting of a complex-valued QAM channel equalization problem (Cha & Kassam, 1995) and an adaptive beam-forming problem (Suksmono & Hirose, 2004). In these problems, we compare the performance of Mc-FCRBF with the other existing complex-valued batch learning algorithms: the CRBF by Chen et al. (1994), the FC-MLP by Kim and Adali (2002), the C-ELM (with gaussian hidden nodes) by Li et al. (2005), and FC-RBF by Savitha, Suresh, and Sundararajan (2009).

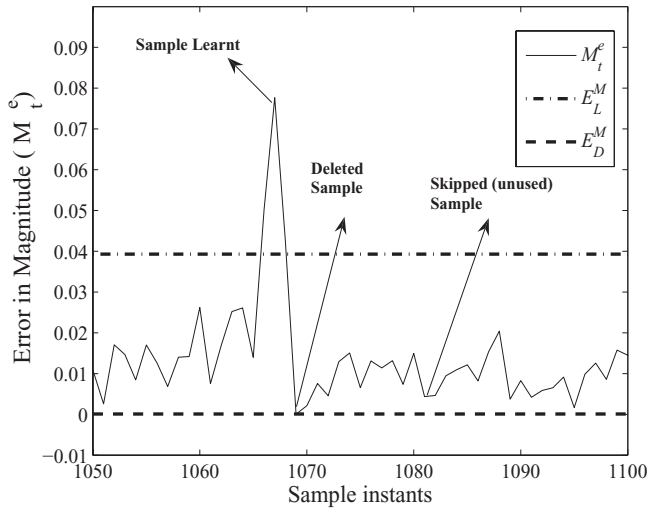
**3.1 Complex-valued Function Approximation Problem.** The synthetic complex-valued function approximation problem that Savitha, Suresh, Sundararajan, and Saratchandran (2009) defined is to approximate the following function:

$$f(\mathbf{z}) = \frac{1}{1.5} \left( z_3 + 10z_1z_4 + \frac{z_2^2}{z_1} \right), \quad (3.1)$$

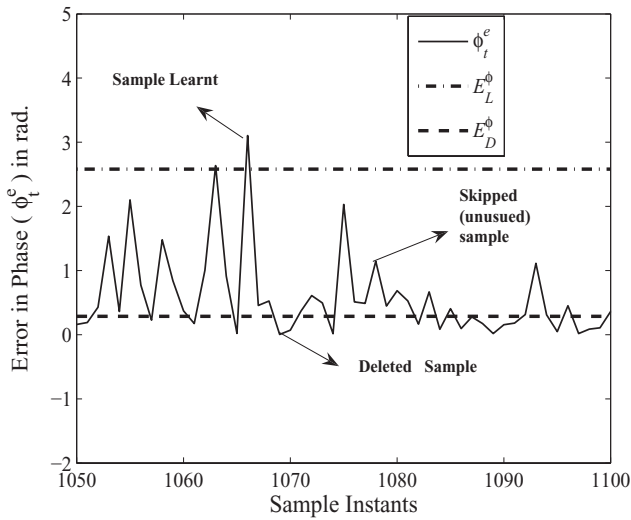
where  $z_i \in \mathbb{C}$  are initialized within a ball of radius 2.5. A training data set of 3000 randomly chosen samples and a testing data set of 1000 randomly chosen samples are used for the study. The number of hidden neurons ( $K$ ) is chosen using a heuristic procedure similar to that presented by Suresh, Omkar, Mani, and Guru Prakash (2003) for real-valued networks.

**3.1.1 Working Principle of Mc-FCRBF.** We first show the self-regulating nature of the parameter update magnitude ( $E_l^M$ ) and phase threshold ( $E_l^\phi$ ). Figure 3 shows the instantaneous magnitude and phase errors, the parameter update magnitude and phase thresholds, and the delete magnitude and phase thresholds over a window of 50 samples (sample instants 1050–1100) during epoch 50. Figure 3a gives a snapshot of the instantaneous magnitude error, the parameter update, and the delete magnitude thresholds. The figure clearly shows the self-regulating nature of the parameter update thresholds and the selective participation of samples in the learning process—for example:

- A few samples whose instantaneous magnitude error is greater than the parameter update magnitude threshold ( $E_l^M$ ) at the time of their



(a) A snap shot of instantaneous magnitude error from 1,050-1,100 sample instants



(b) A snap shot of instantaneous phase error from 1,050-1,100 sample instants

Figure 3: Working principle of the metacognitive component of Mc-FCRBF.

presentation to the network are selected for participation in the learning process (e.g., sample instant 1067).

- A few samples whose instantaneous magnitude error is less than the delete magnitude threshold ( $E_d^M$ ) are deleted from the training data set (e.g., sample instant 1069).
- There are a few samples whose instantaneous magnitude error is greater than  $E_d^M$  and lesser than  $E_l^M$  (e.g., sample instant 1081). These samples do not take part in the parameter update and are not deleted in the current epoch. Instead, they are skipped in the current epoch and retained in the training sample set to be presented to the network in future epochs.

Similarly, Figure 3b gives a snapshot of the instantaneous phase error, the parameter update phase threshold ( $E_l^\phi$ ), and the delete phase threshold ( $E_d^\phi$ ) over 50 samples (sample instants 1050–1100) during the learning process in epoch 50. The effect of self-regulation based on the instantaneous phase error of the samples is clearly seen from this plot:

- A few samples whose instantaneous magnitude error is greater than the parameter update magnitude threshold ( $E_l^\phi$ ) participated in the learning process (e.g., sample instant 1065).
- A few samples whose instantaneous phase errors are less than the delete phase threshold ( $E_d^\phi$ ) are deleted from the training data set (e.g., sample instant 1069).
- There are a few samples whose instantaneous magnitude error is greater than  $E_d^\phi$  and less than  $E_l^\phi$  (e.g., sample instant 1068). These samples do not take part in learning and are not deleted in the current epoch. Instead, they are skipped in the current epoch and retained in the training data set, to be presented to the network in the future epochs.

Figure 4 gives the sample history for the number of samples that participated in learning and those that are not used during the learning process over 5000 epochs. From Figure 4, it can be observed that on average, only 600 samples participated in the learning process in each epoch. The remaining samples are redundant samples that are deleted or samples that do not contribute significantly to the learning process and are reserved in the training sample set without being learned. Thus, from the discussion in this section, it is evident that the metacognitive components of Mc-FCRBF control the learning process of the FC-RBF by selecting samples for participation in the learning process of the FC-RBF network.

*3.1.2 Performance Study on the Complex-Valued Function Approximation Problem.* In this section, we present the performance of Mc-FCRBF on CFAP in comparison with other complex-valued batch learning algorithms available in the literature. The performance comparison results of these networks

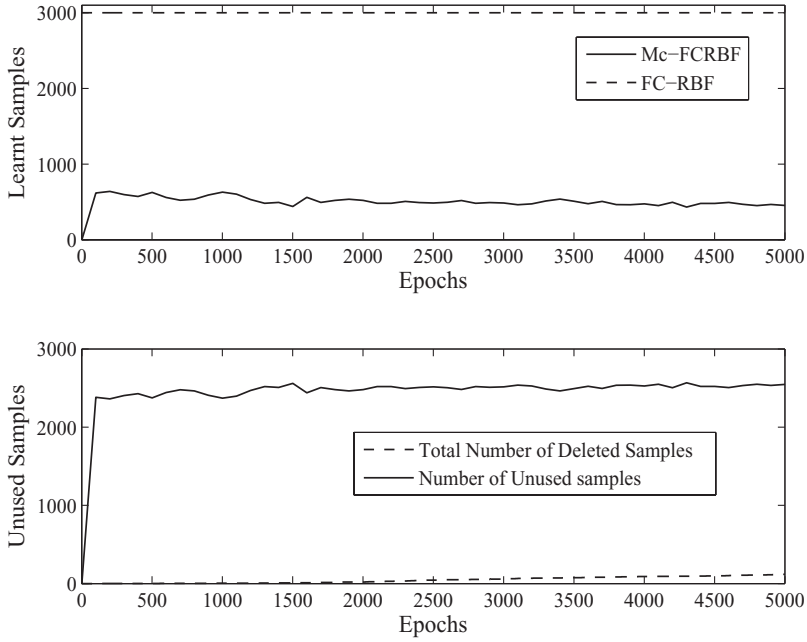


Figure 4: Sample history of used and unused samples.

and algorithms are presented in Table 1. Comparison of these complex-valued learning algorithms is done based on the number of neurons, the training time, and the training and testing root mean square error ( $J_{Me}$ ) and absolute average phase error ( $\Phi_e$ ). The performance measures, the root mean square magnitude error ( $J_{Me}$ ) and the absolute average phase error ( $\Phi_e$ ) are defined as

$$J_{Me} = \sqrt{\frac{1}{N \times n} \sum_{t=1}^N \sum_{l=1}^n (e_l^t \cdot \bar{e}_l^t)}, \quad (3.2)$$

$$\Phi_e = \frac{1}{N \times n} \sum_{t=1}^N \sum_{l=1}^n |\arg(y_l^t \text{conj}(\hat{y}_l^t))|. \quad (3.3)$$

$e_l^t$  is the error of the  $k$ th output neuron for the  $t$ th sample, and  $N$  is the total number of samples.

From the table, it can be observed that the Mc-FCRBF has a significantly better generalization ability and requires lower computational effort than

Table 1: Performance Comparison for CFAP.

Algorithm	Time (sec)	$K$	Training		Testing	
			$J_{Me}$	$\phi_e$ (deg.)	$J_{Me}$	$\phi_e$ (deg.)
Mc-FCRBF	5620	15	0.0009	0.5	0.0009	0.6
FC-RBF	6013	20	0.019	16	0.048	16
FC-MLP	1857	15	0.029	15.7	0.05	15.6
CRBF	9233	15	0.15	51	0.18	52
CELM	0.2	15	0.19	90	0.23	88

that of the FC-RBF network. Because the metacognitive component of Mc-FCRBF controls the learning process by deleting samples with similar information, it avoids overtraining and hence improves the generalization ability of the FC-RBF network. Earlier, it was observed from Figure 4 that on average, only 600 samples participate in the learning process. Therefore, there is a reduction in computational time by nearly 400 seconds. Figure 5 gives the magnitude and phase error convergence plots of the FC-RBF network for the CFAP, with and without the metacognitive component, over a window of 1000 epochs. It can be observed from the plots that the metacognitive component of Mc-FCRBF has accelerated the convergence of both the magnitude and phase errors of the FC-RBF network. The faster convergence also contributes to the overall generalization performance as shown in Table 1, where the errors of Mc-FCRBF are lower by at least an order compared to those of FC-RBF network. It must be noted here that like any other batch learning algorithm, the performance of Mc-FCRBF is also influenced by the presence of outliers. However, such outliers can be detected by amending the self-regulating conditions. If the current sample error is greater than, say,  $6\sigma$  (where  $\sigma$  is the standard deviation), then we can classify samples as outliers. These samples will not participate in the learning process (current epoch). In addition, such samples that consistently produce more than  $6\sigma$  errors over  $L$  consecutive epochs can be deleted from the data set.

**3.2 QAM Channel Equalization Problem.** Quadrature amplitude modulation (QAM) is a widely used analog/digital modulation scheme that modulates the amplitudes of two carrier waves (quadrature carriers) that are 90 degrees out of phase with each other to convey two message signals. Thus, the signals in the QAM schemas are complex valued. The nonlinear characteristics of the channel cause spectral spreading, intersymbol interference, and constellation warping when the QAM signals are transmitted over a channel. Hence, an equalizer is essential at the receiver end of the communication channel to reduce the precursor intersymbol interference without any substantial degradation in the signal-to-noise ratio.



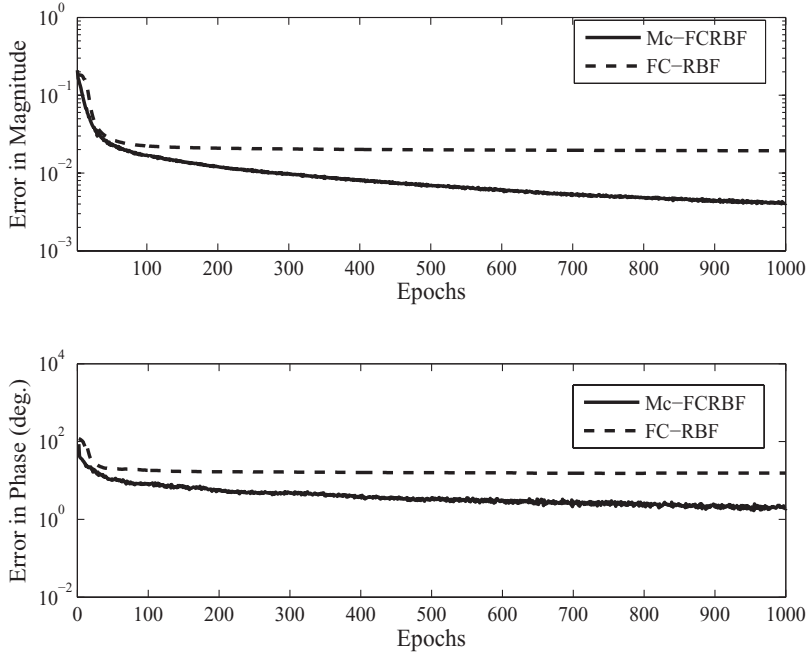


Figure 5: Error convergence plots for CFAP.

In this study, we consider the well-known nonlinear complex-valued model presented by Cha & Kassam (1995). It is a nonlinear complex-valued channel equalization model used in 4-QAM signaling to transmit one of the following symbols— $1 + i$ ,  $1 - i$ ,  $-1 + i$ , or  $-1 - i$ —to represent two bits. Because it requires complex-valued channel observation at three consecutive instants to determine the transmitted symbol at time  $t - \tau$ , the Cha and Kassam (1995) channel model is a third-order channel model. In this study, the equalizer delay,  $\tau$ , is set to 1. The channel output at time  $t$  that is the input to the equalizer at time  $t$ , as defined by Cha and Kassam (1995), is given as

$$\begin{aligned} z_t &= o_t + 0.1\sigma_t^2 + 0.05\sigma_t^3 + v_t, \\ o_t &= (0.34 - 0.27i)s_t + (0.87 + 0.43i)s_{t-1} + (0.34 - 0.21i)s_{t-2}, \end{aligned} \quad (3.4)$$

where  $v_t(0, 0.01)$  is a white gaussian noise with zero-mean and 0.01 variance,  $s_t$  is the transmitted symbol at time  $t$ .

Thus, the objective of a complex-valued channel equalizer is to estimate the transmitted symbol  $s(t - \tau)$ , given the channel observation at time  $t$ . In this study, the complex-valued equalizers are trained with a stream of 5000

Table 2: Performance Comparison of CRBF Equalizers on the Cha and Kassam Model.

Algorithm	Time (sec)	K	Training Error		Testing Error	
			$J_{Me}$	$\phi_e$ (deg.)	$J_{Me}$	$\phi_e$ (deg.)
Mc-FCRBF	1281.4	15	0.1428	9.57	0.1664	9.539
FC-RBF	3840.4	14	0.3399	8.18	0.3476	11.46
FC-MLP	3862	15	0.2	6.47	0.72	31.1
CRBF	8106.6	15	0.5630	35.1911	0.5972	39.86
C-ELM	0.3605	15	0.572	34.14	0.5772	35.11

transmitted symbols corrupted with noise having a signal-to-noise ratio (SNR) of 20 dB. The performance of the equalizers is then tested using  $10^5$  samples corrupted with a noise having SNRs between 4 dB and 20 dB. The performance of the various complex-valued equalizers is shown in Table 2. They are compared with respect to the number of neurons used (K), the time taken for training, and the training and testing root mean square magnitude and average absolute phase errors. From the table, it can be seen that the Mc-FCRBF equalizer outperforms other complex-valued equalizers in performing the QAM channel equalization problem with a reduced computational time compared to the FC-RBF equalizer.

The symbol error rate (SER) plot of these equalizers is presented in Figure 6. From the figure, it can be observed that Mc-FCRBF has the lowest symbol error rate compared to the other complex-valued equalizers. Moreover, it can also be noted that Mc-FCRBF equalizer achieves an SER of about  $-2.5$  dB at an SNR of nearly 16 dB, while the optimum Bayesian equalizer achieves this SER at an SNR of nearly 14 dB.

**3.3 Adaptive Beam-Forming Problem.** Adaptive beam forming is an antenna array signal processing problem where the beams are directed to desired signal directions (beam pointing) and the nulls are directed to the interference directions (null pointing) (Suksmono & Hirose, 2004). A set of  $M$  single-transmit antennas operating at the same carrier frequency and  $L$  uniformly spaced receiver elements constitute a typical beam-forming mechanism. The spacing between the receiver elements ( $d$ ) is usually set at half the wavelength ( $\lambda$ ) of the received signal. Let  $\theta$  be the angle of incidence that an incoming signal makes with the receiver array broadside. Then, from the basic trigonometric identities and the geometry of the sensor array, the signal received at the  $k$ th receiver antenna element can be derived as

$$\bar{x}_k = e^{\frac{j2\pi k d \sin\theta}{\lambda}}. \quad (3.5)$$

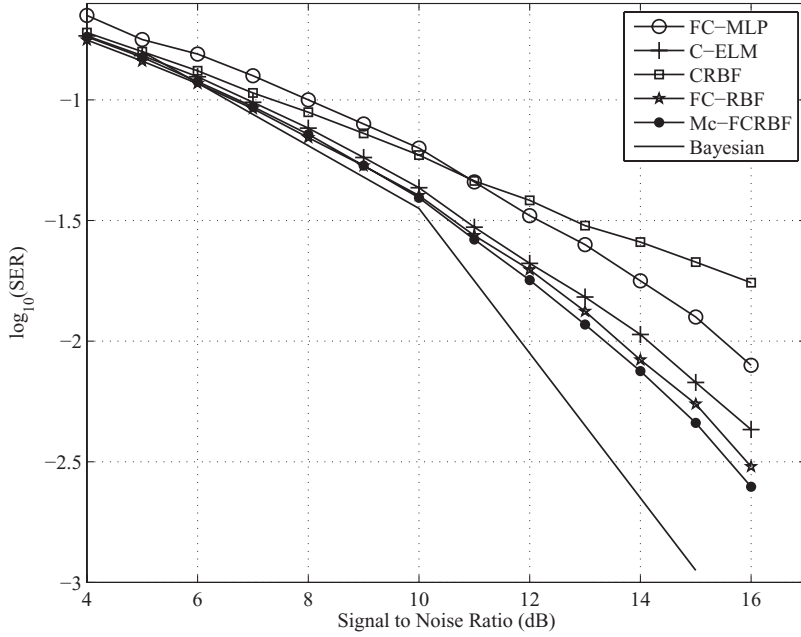


Figure 6: Error probability curve for the various complex-valued equalizers.

Let  $\eta_k$  be the noise at the  $k$ th receiver element. Then the total signal induced at the  $k$ th element at a given instant is the input to the beam former and is given by

$$\mathbf{z} = [\bar{x}_0 + \eta_0 \quad \bar{x}_1 + \eta_1 \quad \bar{x}_2 + \eta_2 \quad \dots \quad \bar{x}_L + \eta_L]^T, \quad (3.6)$$

where superscript  $T$  represents the transpose of the vector.

Let  $\mathbf{b} = [b_1 b_2 \dots b_k]^T$  be a weight vector for the sensor array. The actual signal transmitted at a given instant ( $y$ ) is given by

$$y = \mathbf{b}^H \mathbf{z}. \quad (3.7)$$

This transmitted signal ( $y$ ) forms the target of the beam former. Thus, the objective of an adaptive beam former is to estimate the weights ( $\mathbf{b}$  of equation 3.7), given the signal transmitted ( $y$ ) and the signal received by the beam former antenna array ( $\mathbf{z}$ ).

In this letter, a five-sensor uniform linear array is considered. The desired signal directions are set as  $-30$  degrees and  $+30$  degrees and the directions of interferences considered are  $-15$  degrees,  $0$  degrees, and  $+15$  degrees, as shown by Suksmono and Hirose (2004). The received signal at array

Table 3: Performance Comparison of Various Complex-Valued Neural Network-Based Beam Formers.

Direction of Arrival	Gain (dB)				
	CRBF	C-ELM	FC-RBF	Mc-FCRBF	Matrix Method
Beam-1: $-30^\circ$	-17.94	-18.05	-16.99	-16.98	-13.98
Null-1: $-15^\circ$	-27.53	-48.28	-58.45	-60.54	-57.02
Null-2: $0^\circ$	-27	-41.64	-57.23	-59.6	-57
Null-3: $15^\circ$	-28.33	-47.5	-56.32	-59.6	-57.02
Beam-2: $30^\circ$	-17.92	-16.68	-17.00	-16.9	-13.98

elements ( $\mathbf{z}$ ) is corrupted with an additive gaussian noise at 50 dB SNR. A training data set of 250 randomly chosen samples with 50 for each signal or interference angle is used to train the various beam formers. A 5-5-1 network (a network with five input neurons, five hidden neurons, and one output neuron) is used for the batch learning FC-MLP (Kim & Adali, 2002), CRBF (Chen et al., 1994), C-ELM (Li et al., 2005), FC-RBF (Savitha, Suresh, & Sundararajan 2009) and Mc-FCRBF. The gains for the signals and interference nulls for different beam formers are summarized in Table 3. From the table, it can be observed that the Mc-FCRBF beam former outperforms all the other complex-valued beam formers, and its interference nulling performance is better than that of the conventional optimal matrix method explained by Monzingo and Miller (2003).

#### 4 Performance Evaluation of Mc-FCRBF for Classification Problems —

Recent research studies have revealed that the complex-valued neural networks are better decision makers than the real-valued networks due to their orthogonal decision boundaries (Nitta, 2004). The orthogonality of decision boundaries of a three-layered complex-valued network with a split-type of activation function has been proven in Nitta (2004). In this section, we first prove that an FC-RBF network with a *sech* activation function also has orthogonal decision boundaries and then verify the classification ability of Mc-FCRBF on benchmark and practical real-valued classification problems.

##### 4.1 Orthogonality of Decision Boundaries of the FC-RBF Network.

It has been proven in the literature that the presence of orthogonal decision boundaries provides better decision making regarding complex-valued classifiers. Hence, study of the decision-making ability of Mc-FCRBF classifiers requires studying the nature of its decision boundaries and checking if they are orthogonal. Therefore, we first present an analytical study on the nature of the decision boundaries in McFCRBF before evaluating its

performance on benchmark and practical problems. In this section, we drop the superscript  $t$  for notational convenience.

Let the  $m$ -dimensional input be  $\mathbf{z} \in \mathbb{C}^m = \mathbf{z}^R + i\mathbf{z}^I$ ; the hidden-layer responses be  $\mathbf{h} \in \mathbb{C}^K = \mathbf{h}^R + i\mathbf{h}^I$ ; the input to each hidden neuron be  $O_k = O_k^R + O_k^I = \mathbf{v}_k^T(\mathbf{z} - \mathbf{u}_k)$ ,  $k = 1, \dots, K$ , the output weights be  $w_{kj} = w_{kj}^R + iw_{kj}^I$ ; and the predicted output of the network be  $\hat{\mathbf{y}} \in \mathbb{C}^n = \hat{\mathbf{y}}^R + i\hat{\mathbf{y}}^I$ , where the superscripts  $R$  and  $I$  represent the real and imaginary parts of the complex signal, respectively. Since the FC-RBF network is the basic building block of Mc-FCRBF, we consider in this section a single output neuron of the FC-RBF network and show that the decision boundaries formed by the output neurons of FC-RBF network are orthogonal.

The predicted output of the  $l$ th output neuron of the network ( $\hat{y}_l$ ) is given by

$$\hat{y}_l = \sum_{k=1}^K w_{lk} h_k; l = 1, \dots, n \quad (4.1)$$

$$\text{where } h_k = h_k^R + ih_k^I = \text{sech}(O_k) = \text{sech}(O_k^R + iO_k^I) \quad (4.2)$$

$$= \left( \frac{2(\cos(O_k^I)\cosh(O_k^R) - i\sin(O_k^I)\sinh(O_k^R))}{\cos(2O_k^I) + \cosh(2O_k^R)} \right). \quad (4.3)$$

Using the laws of differentiation and the trigonometric and hyperbolic product of sum identities in equation 4.3, the following can be derived for the  $k$ th hidden neuron:

$$\frac{\partial h_k^R}{\partial O_k^R} = \frac{\sinh(O_k^R)\cos(3O_k^I) - \sinh(3O_k^R)\cos(O_k^I)}{(\cos(2O_k^I) + \cosh(2O_k^R))^2}, \quad (4.4)$$

$$\frac{\partial h_k^I}{\partial O_k^I} = \frac{\sinh(O_k^R)\cos(3O_k^I) - \sinh(3O_k^R)\cos(O_k^I)}{(\cos(2O_k^I) + \cosh(2O_k^R))^2}, \quad (4.5)$$

$$\frac{\partial h_k^R}{\partial O_k^I} = \frac{\cosh(O_k^R)\sin(3O_k^I) - \cosh(3O_k^R)\sin(O_k^I)}{(\cos(2O_k^I) + \cosh(2O_k^R))^2}, \quad (4.6)$$

$$\frac{\partial h_k^I}{\partial O_k^R} = \frac{\cosh(3O_k^R)\sin(O_k^I) - \cosh(O_k^R)\sin(3O_k^I)}{(\cos(2O_k^I) + \cosh(2O_k^R))^2}. \quad (4.7)$$

From equations 4.4 to 4.7, it can be observed that

$$\frac{\partial h_k^R}{\partial O_k^R} = \frac{\partial h_k^I}{\partial O_k^I}, \quad (4.8)$$

$$\frac{\partial h_k^R}{\partial O_k^I} = -\frac{\partial h_k^I}{\partial O_k^R}. \quad (4.9)$$

Hence, the *sech* activation function satisfies the Cauchy-Riemann equations.<sup>3</sup>

Equation 4.1 can be written as

$$\widehat{y}_I = \widehat{y}_I^R + i\widehat{y}_I^I = \sum_{k=1}^K (w_{lk}^R \cdot h_k^R - w_{lk}^I \cdot h_k^I) + i(w_{lk}^R \cdot h_k^I + w_{lk}^I \cdot h_k^R). \quad (4.10)$$

It can be observed from the equation 4.10 that

$$\widehat{y}_I^R = \sum_{k=1}^K w_{lk}^R \cdot h_k^R - w_{lk}^I \cdot h_k^I \rightarrow \mathbf{C}^R, \quad (4.11)$$

$$\text{and } \widehat{y}_I^I = \sum_{k=1}^K w_{lk}^R \cdot h_k^I + w_{lk}^I \cdot h_k^R \rightarrow \mathbf{C}^I \quad (4.12)$$

are the decision boundaries for the real and imaginary parts of the output, respectively. The normal vectors for the decision surfaces given by equations 4.11 and 4.12 are given by

$$\mathbf{Q}^R(O_k^R, O_k^I) = \frac{\partial \widehat{y}_I^R}{\partial O_k^R} \frac{\partial \widehat{y}_I^R}{\partial O_k^I} \quad (4.13)$$

$$\text{and } \mathbf{Q}^I(O_k^R, O_k^I) = \frac{\partial \widehat{y}_I^I}{\partial O_k^R} \frac{\partial \widehat{y}_I^I}{\partial O_k^I}. \quad (4.14)$$

From equations 4.11 and 4.12, it can be seen that  $\frac{\partial \widehat{y}_I^R}{\partial h_k^R} = w_{lk}^R$ ,  $\frac{\partial \widehat{y}_I^I}{\partial h_k^R} = w_{lk}^I$ ,  $\frac{\partial \widehat{y}_I^R}{\partial h_k^I} = -w_{lk}^I$ , and  $\frac{\partial \widehat{y}_I^I}{\partial h_k^I} = w_{lk}^R$ . Using these expressions and the Cauchy-Riemann equations shown in equations 4.8 and 4.9, the following can be derived:

$$\frac{\partial \widehat{y}_I^R}{\partial O_k^R} = \frac{\partial \widehat{y}_I^R}{\partial h_k^R} \frac{\partial h_k^R}{\partial O_k^R} + \frac{\partial \widehat{y}_I^R}{\partial h_k^I} \frac{\partial h_k^I}{\partial O_k^R} = w_{lk}^R \frac{\partial h_k^R}{\partial O_k^R} + w_{lk}^I \frac{\partial h_k^R}{\partial O_k^I}, \quad (4.15)$$

---

<sup>3</sup>The objective here is to prove that the two decision boundaries formed by the real and imaginary parts of the output neuron are orthogonal to each other. This requires that the derivatives of the real and imaginary parts of the outputs with respect to its inputs be separated. Therefore, we use the complex-valued derivative defined by Cauchy-Riemann equation, instead of  $\mathbb{C} - \mathbb{R}$  calculus.

$$\frac{\partial \hat{y}_l^R}{\partial O_k^I} = \frac{\partial \hat{y}_l^R}{\partial h_k^R} \frac{\partial h_k^R}{\partial O_k^I} + \frac{\partial \hat{y}_l^R}{\partial h_k^I} \frac{\partial h_k^I}{\partial O_k^I} = w_{lk}^R \frac{\partial h_k^R}{\partial O_k^I} - w_{lk}^I \frac{\partial h_k^R}{\partial O_k^R}, \quad (4.16)$$

$$\frac{\partial \hat{y}_l^I}{\partial O_k^R} = \frac{\partial \hat{y}_l^I}{\partial h_k^R} \frac{\partial h_k^R}{\partial O_k^R} + \frac{\partial \hat{y}_l^I}{\partial h_k^I} \frac{\partial h_k^I}{\partial O_k^R} = w_{lk}^I \frac{\partial h_k^R}{\partial O_k^R} - w_{lk}^R \frac{\partial h_k^R}{\partial O_k^I}, \quad (4.17)$$

$$\frac{\partial \hat{y}_l^I}{\partial O_k^I} = \frac{\partial \hat{y}_l^I}{\partial h_k^R} \frac{\partial h_k^R}{\partial O_k^I} + \frac{\partial \hat{y}_l^I}{\partial h_k^I} \frac{\partial h_k^I}{\partial O_k^I} = w_{lk}^I \frac{\partial h_k^R}{\partial O_k^I} + w_{lk}^R \frac{\partial h_k^R}{\partial O_k^R}, \quad (4.18)$$

Substituting equations 4.15 to 4.18 in equations 4.13 and 4.14, the inner product of the normal vectors shown in equations 4.13 and 4.14 will be

$$\begin{aligned} Q^R(O_k^R, O_k^I) \cdot Q^I(O_k^R, O_k^I) &= \frac{\partial \hat{y}_l^R}{\partial O_k^R} \cdot \frac{\partial \hat{y}_l^I}{\partial O_k^R} + \frac{\partial \hat{y}_l^R}{\partial O_k^I} \cdot \frac{\partial \hat{y}_l^I}{\partial O_k^I} \\ &= w_{lk}^R w_{lk}^I \left( \frac{\partial h_k^R}{\partial O_k^R} \right)^2 + (w_{lk}^I)^2 \left( \frac{\partial h_k^R}{\partial O_k^I} \right) \left( \frac{\partial h_k^R}{\partial O_k^R} \right) \\ &\quad - (w_{lk}^R)^2 \left( \frac{\partial h_k^R}{\partial O_k^I} \right) \left( \frac{\partial h_k^R}{\partial O_k^I} \right) - w_{lk}^R w_{lk}^I \left( \frac{\partial h_k^R}{\partial O_k^I} \right)^2 \\ &\quad + w_{lk}^R w_{lk}^I \left( \frac{\partial h_k^R}{\partial O_k^I} \right)^2 - (w_{lk}^I)^2 \left( \frac{\partial h_k^R}{\partial O_k^I} \right) \left( \frac{\partial h_k^R}{\partial O_k^R} \right) \\ &\quad + (w_{lk}^R)^2 \left( \frac{\partial h_k^R}{\partial O_k^I} \right) \left( \frac{\partial h_k^R}{\partial O_k^I} \right) - w_{lk}^R w_{lk}^I \left( \frac{\partial h_k^R}{\partial O_k^R} \right)^2 \\ &= 0. \end{aligned} \quad (4.20)$$

Thus, it can be inferred from equation 4.20 that the decision boundaries of the real and imaginary axes formed by an output neuron of FC-RBF network are orthogonal to each other, and this can also be extended to all the output neurons of the network.

Based on the above proof, we state the following lemma:

**Lemma 1.** *The decision boundaries formed by the real and imaginary parts of an output neuron of FC-RBF network are orthogonal to each other.*

**4.2 Modifications in the Mc-FCRBF Learning Algorithm to Solve Real-Valued Classification Problems.** The learning algorithm of Mc-FCRBF described in section 2.3 has been used to solve complex-valued function approximation problems. Although it can also be used to approximate the decision surface to solve real-valued classification problems, we modify the metacognitive component to improve the classification performance of McFCRBF. In this respect, the mean squared error defined in

equation 2.5 is replaced by the hinge loss function and the criteria for parameter update are modified to incorporate a classification measure as well.

Recently, Zhang (2003) and Suresh et al. (2008b) showed that in real-valued classifiers, the hinge loss function helps the classifier estimate the posterior probability more accurately than the mean squared error function. Hence, in this letter, we use the hinge loss error function defined as

$$e_l^t = \begin{cases} 0 & \text{if } (y_l^{tR})(\hat{y}_l^{tR}) > 0 \\ y_l^t - \hat{y}_l^t & \text{otherwise} \end{cases} \quad l = 1, 2, \dots, n, \quad (4.21)$$

where the superscript  $R$  refers to the real part of the complex signal.

While solving real-valued classification problems in the complex domain, Mc-FCRBF has to ensure that the predicted class label ( $\hat{c}^t$ ) is the same as that of the target class label ( $c^t$ ). Therefore, we have modified the parameter update conditions to accommodate this class label information also. Accordingly, the sample learning condition of Mc-FCRBF (see equation 2.15) to solve real-valued classification problems is

$$\text{If } \hat{c}^t \neq c^t \text{ OR } \left( M_t^e \geq E_t^M \text{ AND } \phi_t^e \geq E_t^\phi \right). \quad (4.22)$$

Then, update the network parameters according to equations 2.9 to 2.11. Here,  $\hat{c}^t$  is the predicted class label for the sample  $t$  defined as

$$\hat{c}^t = \max_{l=1,2,\dots,C} \text{real}(\hat{y}_l^t). \quad (4.23)$$

In the next section, we evaluate the classification performance of Mc-FCRBF with the above modifications on a set of benchmark and practical classification problems and verify the improved performance due to the orthogonal decision boundaries of McFCRBF.

**4.3 Problems Considered for This Study.** The classification performance of Mc-FCRBF is studied on a set of benchmark real-valued classification problems from the UCI Machine Learning Repository (Blake & Merz, 1998). The problems considered for the study are listed in Table 4. The table also presents descriptions of these data sets, including the number of input features, the number of training and testing samples, and the imbalance factor of the data set. The imbalance factor of the data sets as defined by Suresh et al. (2008b) is

$$\text{I. F.} = 1 - \frac{n}{N} \min_{l=1,\dots,n} N_l, \quad (4.24)$$



Table 4: Description of Benchmark Data Sets Selected from Blake and Merz (1998) for Performance Study.

Type of Data set	Problem	Number of Features	Number of Classes	Number of samples		Imbalance Factor
				Training	Testing	
Multi-categories	Image segmentation	19	7	210	2,100	0
	Vehicle classification	18	4	424	422	0.1
	Glass identification	9	6	109	105	0.68
Binary	Liver disorder	6	2	200	145	0.17
	PIMA data	8	2	400	368	0.225
	Breast cancer	9	2	300	383	0.26
	Ionosphere	34	2	100	251	0.28

where  $N_l$  is the number of samples belonging to a class  $l$ . Note that  $N = \sum_{l=1}^n N_l$ .

**4.4 Transformation of Input Features from the Real Domain to the Complex Domain.** The multilayer, multivalued network (MLMVN) by Aizenberg and Moraga (2007) and the phase-encoded complex-valued neural network (PE-CVNN) by Amin, Islam, and Murase (2009) are two recent complex-valued classifiers developed in the literature. A multivalued neuron in the MLMVN uses multiple-valued threshold logic to map the complex-valued input to  $C$  discrete outputs using a piecewise continuous activation function where  $C$  is the total number of classes. The transformation used in the MLMVN is not unique, leading to misclassification. The misclassification is further enhanced by the increase in the number of sectors inside the unit circle in multicategory classification problems with more classes ( $C$ ). In the PE-CVNN, the complex-valued input features are obtained by phase encoding the real-valued input features between  $[0, \pi]$  using the following transformation:

$$\begin{aligned}
 z &= \exp(i\phi) = \cos\phi + i\sin\phi, \text{ where } \phi \\
 &= \frac{\pi(x-a)}{b-a}; a, b \in \mathbb{R} \text{ and } x \in [a, b].
 \end{aligned} \tag{4.25}$$

Thus, the transformation used in the PE-CVNN is unique. Therefore, in our work, we use the phase-encoded transformation to map the real-valued input features to the complex domain.

In the next section, we study the classification performance of Mc-FCRBF in comparison with the other well-known classifiers in the literature on a set of benchmark and practical data sets.

Table 5: Performance Comparison for the Image Segmentation Problem (Multicategory Classification Problems with Balanced Data Set).

Classifier Domain	Classifier	Number of Neurons	Training Time (sec.)	Testing $\eta$	
				$\eta_o$	$\eta_a$
Real valued	MRAN	76	783	86.52	86.52
	GAP-RBF	83	365	87.19	87.19
	OS-ELM	100	21	90.67	90.67
	ID-SVM	88 <sup>a</sup>	421	89	89
	SMC-RBF	43	142	91	91
	SRAN	47	22	92.3	92.3
Complex valued	PE-CVNN	-	-	93.2 <sup>b</sup>	-
	MLMVN	80	1384	83	83
	FC-MLP	80	374	91.57	91.57
	FC-RBF	38	421	92.33	92.33
	Mc-FCRBF	36	362	92.9	92.9

<sup>a</sup> Support vector machines.

<sup>b</sup> Amin et al. (2009) performed a 10-fold validation with a single-layer network using 90% of the total samples in each validation of training. In our work, we use the training and testing samples according to Table 4.

**4.5 Performance Study: Multicategory Classification Problems.** In this section, we study the classification performances of the various classifiers on the three multicategory classification problems described in Table 4. From Table 4, it can be seen that the image segmentation problem is a multicategory classification problem with a balanced data set, while the vehicle classification and glass identification problems have unbalanced data sets. The performances of the various classifiers are compared using the overall classification accuracy ( $\eta_o$ ) and the average per-class classification accuracy ( $\eta_a$ ) defined by Suresh et al. (2008a). The number of neurons used in training and the average and overall testing efficiencies of Mc-FCRBF classifier for the three multicategory real-valued classification problems are presented in Tables 5 and 6. In all of these problems, the performance of Mc-FCRBF classifier is studied in comparison with the complex-valued classifiers: the MLMVN classifier by Aizenberg and Moraga (2007), the PE-CVNN classifier by Amin et al. (2009) and the FC-MLP classifier by and FC-RBF classifier by Savitha et al. (2012) with hinge loss error function defined by Suresh et al. (2008b). The tables also present the classification performance comparison of Mc-FCRBF classifier with a few real-valued classifiers: the minimal resource allocation network (MRAN), the growing and pruning radial basis function (GAP-RBF) network, the online sequential extreme learning machine (OS-ELM), the incremental decremental support vector machine (ID-SVM), the sequential multicategory classifier using radial basis function (SMC-RBF) network, the RCGA-ELM classifier by Suresh, Venkatesh Babu, and Kim (2009), and the SRAN classifier by Suresh, Keming Dong, and

Table 6: Performance Comparison for the Multicategory Classification Problems with Unbalanced Data Sets.

Problem	Classifier		Number of Neurons	Training Time (sec.)	Testing $\eta$	
	Domain	Classifier			$\eta_o$	$\eta_a$
Vehicle classification	Real valued	MRAN	100	520	59.94	59.83
		GAP-RBF	81	452	59.24	58.23
		OS-ELM	300	36	68.95	67.56
		ID-SVM	150 <sup>a</sup>	350	55.45	58.23
		SMC-RBF	75	120	74.18	73.52
		SRAN	55	113	75.12	76.86
	Complex valued	PE-CVNN <sup>b</sup>	-	-	78.7	-
		MLMVN	90	1396	78 <sup>a</sup>	77.25
		FC-MLP	75	530	76.07	77.49
		FC-RBF	70	678	77.01	77.46
		Mc-FCRBF	90	638	79.38	78.25
		MRAN	51	520	63.81	70.24
Glass identification	Real valued	GAP-RBF	75	410	58.29	72.41
		OS-ELM	60	15	67.62	70.12
		ID-SVM	78 <sup>a</sup>	212	54.22	50.01
		SMC-RBF	58	97	78.09	77.96
		SRAN	59	28	86.2	80.95
		PE-CVNN <sup>b</sup>	-	-	65.5	-
	Complex valued	MLMVN	85	1421	73.24	66.83
		FC-MLP	70	338	80.95	79.6
		FC-RBF	90	452	83.76	80.95
		Mc-FCRBF	90	364	84.75	83.33

<sup>a</sup>Support vector machines.

<sup>b</sup>Amin et al. (2009) performed a 10-fold validation with a single-layer network using 90% of the total samples in each validation of training.

Kim (2010) for these problems. The performance results for SVM, MRAN, GAP-RBF, OS-ELM, ID-SVM, SMC-RBF are reproduced from Suresh et al. (2008a).

From Table 5, it can be observed that the Mc-FCRBF classifier has better classification ability than the FC-RBF classifier. It can also be noted that the Mc-FCRBF classifier outperforms the best-performing real-valued classifiers and other complex-valued classifiers in the image segmentation problem. Moreover, it requires only 36 neurons to perform the image segmentation. From the performance results on the vehicle classification problem shown in Table 6, it can be observed that Mc-FCRBF outperforms all of the real-valued classifiers and other complex-valued classifiers. It can also be observed from Table 6 that in the glass identification problem, the Mc-FCRBF classifier performs better than all of the complex-valued and real-valued classifiers. Hence, it can be inferred that the

Table 7: Performance Comparison on Benchmark Binary Classification Problems.

Problem	Classifier Domain	Classifier	K	Training Time (s)	Testing Efficiency( $\eta_o$ )
Liver disorders	Real valued	SVM	158	0.0972	68.24
		ELM	132	0.1685	71.79
		SRAN	91	3.38	66.9
	Complex valued	FC-RBF	20	133	74.6
		Mc-FCRBF	20	112	76.6
PIMA data	Real valued	SVM	209	0.205	76.43
		ELM	218	0.2942	76.54
		SRAN	97	12.24	78.53
	Complex valued	FC-RBF	20	130.3	78.53
		Mc-FCRBF	20	103	79.89
Breast cancer	Real valued	SVM	190	0.1118	94.20
		ELM	65	0.1442	96.28
		SRAN	7	0.17	96.87
	Complex valued	FC-RBF	10	158.3	97.12
		Mc-FCRBF	10	125	97.4
Iono sphere	Real valued	SVM	30	0.0218	90.18
		ELM	25	0.0396	88.78
		SRAN	21	3.7	90.84
	Complex valued	FC-RBF	10	186.2	89.48
		Mc-FCRBF	10	152	90

metacognitive component of Mc-FCRBF helps it to outperform other best-performing classifiers available in the literature.

**4.6 Performance Study: Binary Classification Problems.** The performance of the Mc-FCRBF classifier is also studied on four binary benchmark data sets from the UCI Machine Learning Repository (Blake & Merz, 1998). The number of neurons, the training time, and the testing efficiencies of the various classifiers for the binary classification data sets are presented in Table 7. From the table, it can be observed that the performance of the Mc-FCRBF classifier is better than the classification performance of the FC-RBF classifier. In other words, the metacognitive system of the Mc-FCRBF has helped to improve the classification performance of the FC-RBF networks. Moreover, Mc-FCRBF also outperforms the real-valued SVM, ELM, and SRAN classifiers.

**4.7 Performance Study: A Practical Classification Problem.** Acoustic emission signals are electrical version of the stress or pressure waves produced by the transient energy release caused by irreversible deformation processes in the material (Suresh, Omkar, Mani, & Menaka, 2004). The

Table 8: Performance Comparison Results for the Acoustic Emission Problem.

Classifier Domain	Classifier	Testing	
		$\eta_o$	$\eta_{av}$
Real valued	Fuzzy C-means clustering		93.34
Complex valued	FC-RBF	96.35	95.2
	<b>Mc-FCRBF</b>	<b>98.54</b>	<b>97.83</b>

classification of the different sources of the acoustic emission is difficult due to the superficial similarities that exist among them. The existence of ambient noise and pseudo-acoustic emission signals increases the difficulty in their classification

In this study, the acoustic emission signal data collected by Suresh et al. (2004) are used to study the classification performance of the Mc-FCRBF classifier. (For details on the input features and the experimental setup used, see Omkar et al., 2002.) Table 8 presents the performance results of the Mc-FCRBF classifier. The results of the the Mc-FCRBF classifier are compared against a fuzzy K-means clustering algorithm by Omkar et al. (2002) and the FC-RBF classifier. It can be seen that the Mc-FCRBF classifier requires only 10 neurons to achieve an overall testing efficiency of 98.54%, which is better than the classification efficiency of the FC-RBF classifier by nearly 2.5%. Moreover, the Mc-FCRBF classifier clearly outperforms all of the other classifiers available in the literature in the classification of the acoustic emission signals.

5 Conclusion

This letter presents a metacognitive fully complex-valued radial basis function network and its batch learning algorithm. Mc-FCRBF has two components: the cognitive and the metacognitive components. The FC-RBF network forms the cognitive component, and a self-regulatory mechanism that controls the learning of FC-RBF forms the metacognitive component. In each epoch of the learning process, when a sample is presented to FC-RBF network, the self-regulatory learning mechanism decides when to learn, what to learn and how to learn, thereby addressing the three components of metacognition. Thus, Mc-FCRBF employs the best learning learning strategy to obtain better generalization performance. The working principle of the metacognitive component of Mc-FCRBF is explained using a complex-valued function approximation problem. The function approximation performance of the proposed algorithm is evaluated using the function approximation problem, a QAM channel equalization problem, and an adaptive beam-forming problem. The decision-making ability of Mc-FCRBF is evaluated using a set of benchmark and practical real-valued classification data

sets. The performance study results show that the metacognitive component of Mc-FCRBF improves the approximation and classification performance of the FC-RBF network with reduced computational effort.

## Acknowledgments

---

We thank the reviewers for their critical comments, which improved the quality of the letter significantly. The first and second authors also wish to thank the Ministry of Education (MoE), Singapore, for financial support through tier I (No. M58020020) funding to conduct this study.

## References

---

- Aizenberg, I., & Moraga, C. (2007). Multilayer feedforward neural network based on multi-valued neurons (MLMVN) and a backpropagation learning algorithm. *Soft Computing*, 11(2), 169–193.
- Amin, M. F., Islam, M. M., & Murase, K. (2009). Ensemble of single-layered complex-valued neural networks for classification tasks. *Neurocomputing*, 72(10–12), 2227–2234.
- Blake, C., & Merz, C. (1998). *UCI repository of machine learning databases*. Irvine: Department of Information and Computer Sciences, University of California, Irvine. <http://archive.ics.uci.edu/ml/>.
- Brandwood, D. H. (1983). A complex gradient operator and its application in adaptive array theory. *IEEE Proceedings F: Radar and Signal Processing*, 130(1), 11–16.
- Bregains, J. C., & Ares, F. (2006). Analysis, synthesis, and diagnostics of antenna arrays through complex-valued neural networks. *Microwave and Optical Technology Letters*, 48(8), 1512–1515.
- Cha, I., & Kassam, S. A. (1995). Channel equalization using adaptive complex radial basis function networks. *IEEE Journal on Selected Areas in Communications*, 13(1), 122–131.
- Chen, S., McLaughlin, S., & Mulgrew, N. (1994). Complex-valued radial basis function network, part I: Network architecture and learning algorithms. *EURASIP Signal Processing Journal*, 35(1), 19–31.
- Fiori, S. (2005). Nonlinear complex-valued extensions of Hebbian learning: An essay. *Neural Computation*, 17(4), 779–838.
- Goh, S. L., & Mandic, D. P. (2004). A complex-valued RTRL algorithm for recurrent neural networks. *Neural Computation*, 16(12), 2699–2713.
- Goh, S. L., & Mandic, D. P. (2007a). An augmented extended Kalman filter algorithm for complex-valued recurrent neural networks. *Neural Computation*, 19(4), 1039–1055.
- Goh, S. L., & Mandic, D. P. (2007b). An augmented CRTRL for complex-valued recurrent neural networks. *Neural Networks*, 20(10), 1061–1066.
- Haykins, S. (1999). *Neural networks: A comprehensive foundation*. Upper Saddle River, NJ: Prentice Hall.

- Isaacson, R., & Fujita, F. (2006). Metacognitive knowledge monitoring and self-regulated learning: Academic success and reflections on learning. *Journal of the Scholarship of Teaching and Learning*, 6(1), 39–55.
- Jianping, D., Sundararajan, N., & Saratchandran, P. (2000). Complex-valued minimal resource allocation network for nonlinear signal processing. *International Journal of Neural Systems*, 10(2), 95–106.
- Joysula, D. P., Vadali, H., Donahue, B. J., & Hughes, F. C. (2009). Modeling metacognition for learning in artificial systems. In *Proceedings of 2009 World Congress on Nature and Biologically Inspired Computing (NABIC 2009)* (pp. 1419–1424). Piscataway, NJ: IEEE.
- Kim, T., & Adali, T. (2002). Fully-complex multilayer perceptron network for nonlinear signal processing. *Journal of VLSI, Signal Processing*, 32(1–2), 29–43.
- Kim, T., & Adali, T. (2003). Approximation by fully complex multilayer perceptrons. *Neural Computation*, 15(7), 1641–1666.
- Li, M. -B., Huang, G. -B., Saratchandran, P., & Sundararajan, N. (2005). Fully complex extreme learning machines. *Neurocomputing*, 68(1–4), 306–314.
- Li, M.-B., Huang, G.-B., Saratchandran, P., & Sundararajan, N. (2006). Complex-valued growing and pruning RBF neural networks for communication channel equalization. *IEE Proc.-Vision, Image and Signal Processing*, 153(4), 411–418.
- Monzingo, R. A., & Miller, T. W. (2003). *Introduction to adaptive arrays*, Raleigh, NC: SciTech Publishing.
- Nelson, T. O., & Narens, L. (1992). Metamemory: A theoretical framework and new findings. In T. O. Nelson (Ed.), *Metacognition: Core readings* (pp. 9–24). Needham Heights, MA: Allyn and Bacon.
- Nitta, T. (2003). Solving the XOR problem and the detection of symmetry using a single complex-valued neuron. *Neural Networks*, 16(8), 1101–1105.
- Nitta, T. (2004). Orthogonality of decision boundaries of complex-valued neural networks. *Neural Computation*, 16(1), 73–97.
- Omkar, S. N., Suresh, S., Raghavendra, T. R., & Mani, V. (2002). Acoustic emission signal classification using fuzzy C-means clustering. In *Proc. of the ICONIP'02, 9th International Conference on Neural Information Processing*, 4 (pp. 1827–1831). Piscataway, NJ: IEEE.
- Rattan, S.S.P., & Hsieh, W. W. (2005). Complex-valued neural networks for nonlinear complex principal component analysis. *Neural Networks*, 18(1), 61–69.
- Remmert, R. (1991). *Theory of complex functions*. Berlin: Springer Verlag.
- Rivers, W. P. (2001). Autonomy at all costs: An ethnography of metacognitive self-assessment and self-management among experienced language learners. *Modern Language Journal*, 85(2), 279–290.
- Savitha, R., Suresh, S., & Sundararajan, N. (2009). A fully complex-valued radial basis function network and its learning algorithm. *International Journal of Neural Systems*, 19(4), 253–267.
- Savitha, R., Suresh, S., & Sundararajan, N. (2010). A self-regulated learning in fully complex-valued radial basis function networks. In *Proceedings of the International Joint Conference on Neural Networks* (pp. 1–8). Piscataway, NJ: IEEE.
- Savitha, R., Suresh, S., Sundararajan, N., & Kim, H. J. (2012). A fully complex-valued radial basis function classifier for real-valued classification problems. *Neurocomputing*, 78(1), 104–110.

- Savitha, R., Suresh, S., Sundararajan, N., & Saratchandran, P. (2009). A new learning algorithm with logarithmic performance index for complex-valued neural networks. *Neurocomputing*, 72(16–18), 3771–3781.
- Savitha, R., Vigneshwaran, S., Suresh, S., & Sundararajan, N. (2009). Adaptive beamforming using complex-valued radial basis function neural networks. *Proceedings of the IEEE Region 10 Conference* (pp. 1–6). Piscataway, NJ: IEEE.
- Shen, C., Lajos, H., & Tan, S. (2008). Symmetric complex-valued RBF receiver for multiple-antenna-aided wireless systems. *IEEE Trans. on Neural Networks*, 19(9), 1659–1665.
- Sinha, N., Saranathan, M., Ramakrishna, K. R., & Suresh, S. (2007). Parallel magnetic resonance imaging using neural networks. *IEEE International Conference on Image Processing*, 3, 149–152.
- Suksmono, A. B., & Hirose, A. (2004). Intelligent beamforming by using a complex-valued neural network. *Journal of Intelligent and Fuzzy Systems*, 15(3–4), 139–147.
- Suresh, S., Keming, Dong, & Kim, H. J. (2010). A sequential learning algorithm for self-adaptive resource allocation network classifier. *Neurocomputing*, 73(16–18), 3012–3019.
- Suresh, S., Omkar, S. N., Mani, V., & Guru Prakash, N. (2003). Lift coefficient prediction at high angle of attack using recurrent neural network. *Aerospace Science and Technology*, 7(8), 595–602.
- Suresh, S., Omkar, S. N., Mani, V., & Menaka, C. (2004). Classification of acoustic emission signal using genetic programming. *Journal of Aerospace Science and Technology*, 56(1), 26–41.
- Suresh, S., Savitha, R., & Sundararajan, N. (2011). A sequential learning algorithm for complex-valued self-regulating resource allocation network-CSRAN. *IEEE Transactions on Neural Networks*, 22(7), 1061–1072.
- Suresh, S., Sundararajan, N., & Saratchandran, P. (2008a). A sequential multi-category classifier using radial basis function networks. *Neurocomputing*, 71(7–9), 1345–1358.
- Suresh, S., Sundararajan, N., & Saratchandran, P. (2008b). Risk-sensitive loss functions for sparse multi-category classification problems. *Information Sciences*, 178(12), 2621–2638.
- Suresh, S., Venkatesh Babu, R., & Kim, H. J. (2009). No-reference image quality assessment using modified extreme learning machine classifier. *Applied Soft Computing*, 9(2), 541–552.
- Wenden, A. L. (1998). metacognitive knowledge and language learning. *Applied Linguistics*, 19(4), 515–537.
- Wirtinger, W. (1927). Zur formalen theorie der funktionen von mehr komplexen veränderlichen. *Annals of Mathematics*, 97.
- Zhang, T. (2003). Statistical behavior and consistency of classification methods based on convex risk minimization. *Ann. Stat.*, 32(1), 56–85.