

2012 Special Issue

A meta-cognitive learning algorithm for a Fully Complex-valued Relaxation Network

R. Savitha^{a,*}, S. Suresh^a, N. Sundararajan^b^a School of Computer Engineering, Nanyang Technological University, Singapore, Singapore^b Sri Jayachamarajendra College of Engineering (SJCE), Mysore, India

ARTICLE INFO

Keywords:

Complex-valued neural networks
 Fully complex-valued relaxation network
 Meta-cognition
 Self-regulated learning mechanism
 Multi-category classification

ABSTRACT

This paper presents a meta-cognitive learning algorithm for a single hidden layer complex-valued neural network called “Meta-cognitive Fully Complex-valued Relaxation Network (McFCRN)”. McFCRN has two components: a cognitive component and a meta-cognitive component. A Fully Complex-valued Relaxation Network (FCRN) with a fully complex-valued Gaussian like activation function (*sech*) in the hidden layer and an exponential activation function in the output layer forms the cognitive component. The meta-cognitive component contains a self-regulatory learning mechanism which controls the learning ability of FCRN by deciding *what-to-learn*, *when-to-learn* and *how-to-learn* from a sequence of training data. The input parameters of cognitive components are chosen randomly and the output parameters are estimated by minimizing a logarithmic error function. The problem of explicit minimization of magnitude and phase errors in the logarithmic error function is converted to system of linear equations and output parameters of FCRN are computed analytically. McFCRN starts with zero hidden neuron and builds the number of neurons required to approximate the target function. The meta-cognitive component selects the best learning strategy for FCRN to acquire the knowledge from training data and also adapts the learning strategies to implement best human learning components. Performance studies on a function approximation and real-valued classification problems show that proposed McFCRN performs better than the existing results reported in the literature.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Recent advances in areas like communication, signal processing, image processing, and radar signal processing use signals that are inherently complex-valued. Their physical characteristics can be preserved by representing these signals and their nonlinear transformations in the Complex domain. However, the representation of complex-valued signals and development of efficient complex-valued learning algorithms are restricted by the challenges imposed by operating in the Complex domain.

The appropriate selection of a complex-valued activation function is the foremost challenge in the development of an efficient complex-valued learning algorithm. Essentially, the activation functions are required to be entire and bounded (Haykin, 1998). But, Liouville's theorem (Remmert, 1991) states that an entire and bounded function is a constant in the Complex domain. Therefore, the essential properties of a complex-valued activation function have been reduced to functions that are analytic and *almost everywhere* bounded (Kim & Adali, 2002). However, the use of such

an activation function limits the usefulness of operating in the Complex plane because of their singularities. It has been shown in Savitha, Suresh, Sundararajan, and Saratchandran (2009) that the singularity of such activation functions and their derivatives will influence the convergence of the network.

Supervised learning algorithms using these activation functions and mean squared error function in batch (Chen, McLaughlin, & Mulgrew, 1994; Kim & Adali, 2002; Savitha, Suresh, & Sundararajan, 2009, 2010, in press) and sequential (Jianping, Sundararajan, & Saratchandran, 2000; Li, Huang, Saratchandran, & Sundararajan, 2006; Suresh, Savitha, & Sundararajan, 2011) learning mode are reported in the literature. However, the mean squared error function is a representation of the magnitude of the error and does not consider the phase of the error explicitly (Savitha et al., 2009). Moreover, the batch learning algorithms use slow gradient decent based update and sequential learning algorithms use computationally intensive extended Kalman filter update rule.

Complex-valued Extreme Learning Machine (C-ELM) (Li, Huang, Saratchandran, & Sundararajan, 2005) is a fast learning algorithm available for the complex-valued neural networks. C-ELM estimates the output weights by calculating the Moore–Penrose pseudo-inverse of a non-square matrix, resulting in a non-unique solution to a system of linear equations. This algorithm also uses

* Corresponding author.

E-mail address: savi0001@ntu.edu.sg (R. Savitha).

the mean-squared error function for learning. Moreover, all the above mentioned complex-valued learning algorithms address *how-to-learn* the functional relationship between the input features and their targets. They assume uniform distribution of samples in the input space and use all the samples in the training data set. However, uniform distribution of samples in the input space is not always guaranteed, especially in real-world problems. Therefore, an efficient complex-valued algorithm must enable selective participation of the samples during the learning process.

Recent studies on human learning have suggested that a self-regulated learning in a meta-cognitive environment is the best learning strategy (Isaacson & Fujita, 2006; Rivers, 2001; Wenden, 1998). As most of the machine learning algorithms are inspired from the principles of human learning, it is essential to impart the best human learning strategy in the machine learning framework. It was clearly shown that the generalization performance of the real-valued Self-adaptive Resource Allocation Network (SRAN) (Suresh, Dong, & Kim, 2010) using self-regulation to decide *what-to-learn*, *when-to-learn* and *how-to-learn* the training samples is much better than the algorithms that use all the samples in the training process (Babu & Suresh, 2012). The self-regulatory learning mechanism has been later introduced in a neuro-fuzzy system in Suresh and Subramanian (2011). The principles of self-regulation was extended to the Complex domain and a sequential learning algorithm was developed for a fully Complex-valued Self-regulating Resource Allocation Network (CSRAN) in Suresh et al. (2011). Moreover, a batch learning meta-cognitive fully complex-valued radial basis function network (Savitha et al., in press) has been developed based on Nelson and Narens model of meta-cognition (Nelson & Narens, 1980) in the Complex domain. This model has two components, namely, the cognitive component that represents the knowledge and the meta-cognitive component that has a dynamic model of the cognitive component. During the entire learning process, the cognitive component *monitors* the meta-cognitive component and keeps the meta-cognitive component informed about the current state of the cognitive component. The meta-cognitive component *controls* the cognitive component either by changing the state of the cognitive component or by changing the cognitive component itself. It achieves this by performing one of the following three actions: (a) initiate an action (b) continue an action or (c) terminate an action.

The FC-RBF network is the cognitive component of McFCRBF and a self-regulatory learning mechanism is its meta-cognitive component. FC-RBF that is the cognitive component of McFCRBF uses a gradient descent based batch learning algorithm based on the mean-square error function to estimate the parameters of the network. The mean-square error function used to derive the gradients represents only the magnitude of error and does not represent the phase of the complex-valued error explicitly. Moreover, FC-RBF requires the number of hidden neurons to be fixed a priori, and it has been shown in Savitha et al. (2009) that the number of neurons and the initial parameters influence the convergence of FC-RBF. In addition to these, as FC-RBF uses a gradient descent based batch learning algorithm, it takes huge computational effort to estimate the parameters of the network. Therefore, there is a need to develop a fast, meta-cognitive learning algorithm for a fully complex-valued neural network.

In this paper, we propose a meta-cognitive learning algorithm for a fully complex-valued relaxation network, referred to as, 'Meta-cognitive Fully Complex-valued Relaxation Network (McFCRN)'. McFCRN is developed based on the model of meta-cognition proposed by Nelson and Narens and implemented in the machine learning framework in Savitha et al. (in press). Similarly, McFCRN also has two components: a Fully Complex-valued Relaxation Network (FCRN) is the cognitive component and a self-regulatory learning mechanism is the meta-cognitive component.

FCRN is a fast and accurate learning single hidden layer complex-valued network with the fully complex-valued 'sech' activation function at the hidden layer and an exponential activation function at the output layer. It uses a logarithmic error function that represents both the magnitude and phase of the complex-valued error explicitly as the energy function. FCRN formulates the problem of finding optimal weights as a nonlinear programming problem and solves it with the help of Wirtinger calculus (Wirtinger, 1927). The projection based learning algorithm converts the nonlinear programming problem into a system of linear equations and provides a solution for the optimal weights corresponding to the minimum energy point of the energy function (Suresh, Savitha, & Sundararajan, 2011). This is similar to the relaxation process, where the system always returns to a minimum energy state from a given initial condition (Yu & Tsai, 1992). As the optimal parameters are obtained as a solution to a system of linear equations, McFCRN requires lesser time for training.

The self-regulatory learning mechanism that is at the meta-cognitive component of McFCRN has a dynamic model of FCRN and controls its learning process by selecting suitable strategies for each sample in the training data set. When a new sample is presented, the self-regulatory learning mechanism performs one of the following actions:

- (a) *Sample deletion*: Samples are deleted without being used in the training process.
- (b) *Sample learning*: Learning includes adding a hidden neuron or updating the network parameters.
- (c) *Sample reserve*: Samples are pushed to the rear end of the training sequence and are used at a later stage.

Thus, the self-regulatory learning mechanism decides *what-to-learn* (Sample deletion strategy), *how-to-learn* (Sample learning strategy), and *when-to-learn* (Sample reserve strategy) in a meta-cognitive framework.

The function approximation performance of McFCRN is evaluated using a synthetic complex-valued function approximation problem. Since complex-valued neural networks have been shown to possess exceptional decision making abilities due to their orthogonal decision boundaries, the decision making ability of McFCRN is also studied in comparison to other complex-valued and real-valued classifiers available in the literature using a set of benchmark/practical classification problems from the UCI machine learning repository (Blake & Merz, 1998). Performance study results show the superior approximation and classification ability of McFCRN.

The paper is organized as follows: Section 2 describes the working principles of the cognitive and meta-cognitive components of McFCRN in detail. In Section 3, the approximation and classification ability of McFCRN is studied on a set of benchmark/practical problems available in the literature. Finally, Section 5 presents the conclusions from this study.

2. Meta-cognitive Fully Complex-valued Relaxation Network (McFCRN)

In this section, we present the description of "Meta-cognitive Fully Complex-valued Relaxation Network" (McFCRN). We first describe the architecture and learning algorithm of FCRN that is the cognitive component of McFCRN. Then, we present in detail the meta-cognitive component of McFCRN and its various control strategies.

2.1. Cognitive component: a Fully Complex-valued Relaxation Network (FCRN)

We describe the fully complex-valued relaxation network and its projection based learning algorithm in detail in this

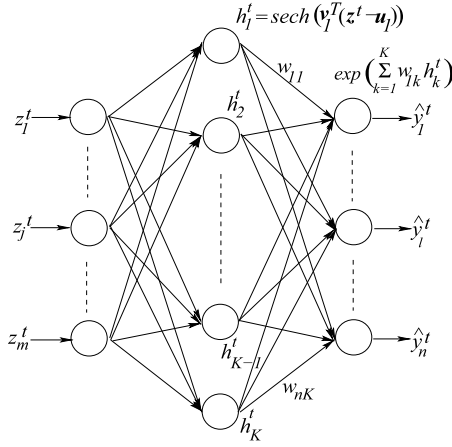


Fig. 1. The architecture of FCRN.

section. Fully Complex-valued Relaxation Network (FCRN) is a single hidden layer feed forward network. FCRN uses a projection based learning algorithm that does not require the number of hidden neurons to be fixed a priori. Instead, it begins with zero hidden neurons and builds a minimal network architecture to approximate the function defined by the training data set.

Let the function to be approximated be given by: $f_1: \mathbf{z} \rightarrow \mathbf{y}$, where $\mathbf{z} = [z_1 \dots z_m]^T \in \mathbb{C}^m$ and $\mathbf{y} = [y_1 \dots y_n]^T \in \mathbb{C}^n$. Let the m -dimensional complex-valued input feature be $\{\mathbf{z}^1, \dots, \mathbf{z}^t, \dots, \mathbf{z}^N\}$ and their functional values be $\{\mathbf{y}^1, \dots, \mathbf{y}^t, \dots, \mathbf{y}^N\}$, where N is the total number of samples. The objective of FCRN is to estimate the functional relationship $f_1(\mathbf{z})$, given the randomly sampled training data set with N samples, $\{(\mathbf{z}^1, \mathbf{y}^1), \dots, (\mathbf{z}^t, \mathbf{y}^t), \dots, (\mathbf{z}^N, \mathbf{y}^N)\}$.

Without loss of generality, let us assume that K neurons are added to the network after learning $t-1$ samples. The architecture of FCRN with K hidden neurons is presented in Fig. 1.

The neurons in the hidden layer of the network employ the fully complex-valued activation function of the type of hyperbolic secant (Savitha et al., 2009) ($\text{sech}(z) = 2/(e^z + e^{-z})$) to map the input features to a hyperdimensional Complex plane, i.e., $\mathbb{C}^m \rightarrow \mathbb{C}^K$. Thus, the response of the j -th hidden neuron for a given input $\mathbf{z}^t \in \mathbb{C}^m$ is given by

$$h_j^t = \text{sech}[\mathbf{v}_j^T(\mathbf{z}^t - \mathbf{u}_j)], \quad j = 1, 2, \dots, K \quad (1)$$

where $\mathbf{v}_j \in \mathbb{C}^m$ is the complex-valued scaling factor and $\mathbf{u}_j \in \mathbb{C}^m$ is the complex-valued center of the hidden neurons.

The neurons in the output layer of FCRN employ an 'exp' activation function and the predicted output ($\hat{\mathbf{y}}^t$) of FCRN with K hidden neurons is given by

$$\hat{y}_k^t = \exp\left(\sum_{j=1}^K w_{kj} h_j^t\right), \quad k = 1, 2, \dots, n \quad (2)$$

where w_{kj} is the output weight connecting the j -th hidden neuron and the k -th output neuron.

The essential properties for a fully complex-valued activation function require that it has to be non-linear, analytic and bounded almost everywhere (a.e.) (Kim & Adali, 2002). Both the activation functions, 'sech' and 'exp' functions, employed in the hidden and output layers of FCRN satisfy these essential properties as shown in Savitha et al. (2009) and Savitha et al. (2009), respectively. Hence, when operated in the bounded region of the Complex plane (Kim & Adali, 2002), FCRN with at most K hidden neurons is capable of learning N distinct samples with random and constant hidden layer parameters (\mathbf{u}_j and \mathbf{v}_j).

We begin the projection based learning algorithm of FCRN with the following propositions:

Proposition 1. The responses of the neurons in the hidden layer are unique. i.e. $\forall \mathbf{z}^t; t = 1, \dots, N, h_p^t \neq h_j^t; p, j = 1, 2, \dots, K, p \neq j$.

Proof. Let us assume that for a given $\mathbf{z}^t, h_p^t = h_j^t; i \neq j$. This assumption is valid if and only if the hidden layer parameters $\mathbf{u}_p = \mathbf{u}_j$ and $\mathbf{v}_p = \mathbf{v}_j$. But the parameters \mathbf{u}_p and \mathbf{u}_j (\mathbf{v}_p and \mathbf{v}_j) are uncorrelated events of a random sequence and cannot be equal. Hence, the hidden layer responses, $h_p^t \neq h_j^t; i, j = 1, 2, \dots, K, p \neq j$. \square

Proposition 2. The responses of the neurons in the hidden layer are non-zero. i.e. $\forall \mathbf{z}, h_p^t \neq 0; i = 1, 2, \dots, K$.

Proof. Let the hidden layer response $h_p^t = 0$. Then the value $\mathbf{v}_p(\mathbf{z}^t - \mathbf{u}_p) = \infty$. This is possible if and only either if the neuron p is infinitely far away from all the samples or if one of the parameters \mathbf{z}, \mathbf{u}_p , or \mathbf{v}_p tends to ∞ i.e.,

$$h_p^t = 0 \quad \text{iff } \mathbf{z} \rightarrow \infty, \text{ or } \mathbf{v}_p \rightarrow \infty, \text{ or } \mathbf{u}_p \rightarrow \infty. \quad (3)$$

However, the activation function of FCRN is used in a bounded region of the Complex plane to ensure efficient learning by the single hidden layer network. Hence, the parameters \mathbf{z}, \mathbf{u}_p , and \mathbf{v}_p do not assume infinite values. Moreover, if a neuron is infinitely far away from all the samples, it is a redundant neuron and can be hence, removed from the network. Therefore, the responses of the neurons in the hidden layer $h_p^t \neq 0$. \square

These propositions will be used in deriving the learning algorithm for FCRN and is presented next.

FCRN network is a complex-valued network that is used to approximate both the magnitude and phase of the complex-valued signals accurately. Hence, the error function should be an explicit representation of both the magnitude and the phase of the complex-valued error, of the form:

$$J = g(\mathbf{y}^t, \hat{\mathbf{y}}^t) \quad (4)$$

$$\text{where } \mathbf{y}^t = M^t \cdot \exp(\phi^t); \hat{\mathbf{y}}^t = \hat{M}^t \cdot \exp(\hat{\phi}^t)$$

where M^t and ϕ^t are the true/target magnitude and phase of the t -th sample and \hat{M}_k^t and $\hat{\phi}_k^t$ are the predicted magnitude and phase of the t -th sample.

One possible choice of the error function J that is an explicit representation of both magnitude and phase can be taken as

$$J = \left[\ln \left(\frac{M^t}{\hat{M}^t} \right)^2 + (\phi^t - \hat{\phi}^t)^2 \right] \quad (5)$$

which can be equivalently written as

$$J = \left[\ln \left(\frac{\mathbf{y}^t}{\hat{\mathbf{y}}^t} \right) \ln \left(\frac{\mathbf{y}^t}{\hat{\mathbf{y}}^t} \right) \right] \quad (6)$$

where $\hat{\mathbf{y}}^t$ is the predicted output of the network as defined in Eq. (2)

and $\ln \left(\frac{\mathbf{y}^t}{\hat{\mathbf{y}}^t} \right)$ is the complex conjugate of $\ln \left(\frac{\mathbf{y}^t}{\hat{\mathbf{y}}^t} \right)$.

Given the training data set $\{(\mathbf{z}^1, \mathbf{y}^1), \dots, (\mathbf{z}^t, \mathbf{y}^t), \dots, (\mathbf{z}^N, \mathbf{y}^N)\}$, the objective is to estimate the output weights w_{kj} with h_j^t obtained using randomly chosen hidden layer parameters such that the minimum energy state of Eq. (6) is estimated, or,

$$W^* = \arg \min_w J(W) \quad (7)$$

where the \arg in the above equation represents the function argument. Estimation of the minimum energy state of Eq. (7) yields the optimum output weight (W^*) for the selected random hidden layer parameters. The complex-valued neural network used to

determine this minimum energy state for a given training data set is termed as a, 'fully complex-valued relaxation network'.

Substituting for $\hat{\mathbf{y}}_t$ from Eq. (2) to Eq. (6), we have,

$$J = \frac{1}{2} \sum_{t=1}^N \sum_{k=1}^C \left(\ln(y_k^t) - \left(\sum_{j=1}^K w_{kj} h_j^t \right) \right) \times \left(\ln(\bar{y}_k^t) - \left(\sum_{j=1}^K \bar{w}_{kj} \bar{h}_j^t \right) \right) \quad (8)$$

where h_j^t is the response of the j -th hidden neuron for the t -th sample and y_k^t is target to the k -th neuron of the t -th sample.

Using the definition of the partial derivatives of a real-valued function of complex-valued variables, the derivative of the error function J (Eq. (8)) with respect to the output weights w_{kj} is:

$$\frac{\partial J}{\partial w_{kj}} = \sum_{t=1}^N h_j^t \left[\ln(\bar{y}_k^t) - \sum_{p=1}^K \bar{w}_{kp} \bar{h}_p^t \right]. \quad (9)$$

The minimum energy state of the error function J is obtained by equating its first order derivative to 0, i.e.,

$$\frac{\partial J}{\partial w_{kj}} = \sum_{t=1}^N h_j^t \left[\ln(\bar{y}_k^t) - \sum_{p=1}^K \bar{w}_{kp} \bar{h}_p^t \right] = 0. \quad (10)$$

Rearranging the above equation, we can obtain

$$\sum_{p=1}^K \bar{w}_{kp} \sum_{t=1}^N h_j^t \bar{h}_p^t = \sum_{t=1}^N \ln(\bar{y}_k^t) h_j^t. \quad (11)$$

Eq. (11) is reduced to:

$$\sum_{p=1}^K \bar{w}_{kp} A_{jp} = B_{jk}. \quad (12)$$

The above equation can be represented in matrix form as

$$\bar{\mathbf{W}} \mathbf{A} = \mathbf{B} \quad (13)$$

where $\mathbf{A} \in \mathbb{C}^{K \times K}$ is the projection matrix given by

$$A_{jp} = \sum_{t=1}^N h_j^t \bar{h}_p^t; \quad j, p = 1, \dots, K \quad (14)$$

and $\mathbf{B} \in \mathbb{C}^{C \times K}$ is the output matrix given by

$$B_{jk} = \sum_{t=1}^N \ln(\bar{y}_{kt}) h_j^t; \quad j = 1, \dots, K, \quad k = 1, \dots, n. \quad (15)$$

Eq. (13) is a system of linear equations and can be easily solved by the inversion of the matrix \mathbf{A} . It can be observed from the definition of the projection matrix in Eq. (14) that the matrix \mathbf{A} is a Hermitian square matrix. It can be inferred from Propositions 1 and 2 that the square and Hermitian projection matrix \mathbf{A} is a non-square positive definite matrix. Thus, the matrix \mathbf{A} is non-singular, and is hence, invertible.

Hence, the unique and optimum output weights of FCRN network can be estimated as:

$$\mathbf{W}^* = \bar{\mathbf{B}} \mathbf{A}^{-1}. \quad (16)$$

Eq. (16) is the linear least square solution to the set of linear equations given by Eq. (11). The output weights given in Eq. (16) are the optimal weights for a given random hidden layer parameters. Thus, FCRN estimates the minimum energy state of the error function defined in Eq. (5) and is computationally less intensive.

2.2. Meta-cognitive component: a self-regulatory learning mechanism

In this section, we describe the working principles of the meta-cognitive component of McFCRN that controls the learning process of FCRN (cognitive component) by selecting suitable learning strategies for each sample (control signal) in the training data set.

The cognitive component *monitors* the meta-cognitive component and updates the meta-cognitive component's dynamic model of the cognitive component using the instantaneous magnitude and phase errors based on the residual error of FCRN ($\mathbf{e}^t = [e_1^t, \dots, e_k^t, \dots, e_n^t]^T$) defined as:

$$e_k^t = y_k^t - \hat{y}_k^t; \quad k = 1, \dots, n; \quad t = 1, \dots, N. \quad (17)$$

Based on the residual error, the monitory signals of McFCRN are defined by:

- The instantaneous magnitude error:

$$M_t^e = \frac{1}{n} \sqrt{\mathbf{e}^{tH} \cdot \mathbf{e}^t}. \quad (18)$$

- The instantaneous phase error:

$$\phi_t^e = \frac{1}{n} \sum_{k=1}^n \left| \arg(y_k^t \cdot \overline{\hat{y}_k^t}) \right| \quad (19)$$

where, $\overline{\hat{y}_k^t}$ refers to the conjugate of the predicted output \hat{y}_k^t and the function $\arg(\cdot)$ returns the phase of a complex-valued number in $[-\pi, \pi]$, and is given by:

$$\arg(z) = \text{atan} \left(\frac{\text{imag}(z)}{\text{real}(z)} \right). \quad (20)$$

The self-regulatory learning mechanism uses these errors to measure the relative knowledge of the cognitive component (FCRN) in comparison to the knowledge contained in the training data set. Based on these measures of relative knowledge, the meta-cognitive component *controls* the learning process of FCRN by selecting one of the following strategies:

- Strategy (a) Sample Deletion:** Samples that contain knowledge already learnt by FCRN are deleted.
- Strategy (b) Sample Learning:** Samples with relatively new knowledge are used to add a new hidden neuron or update the output weights of existing neurons
- Strategy (c) Sample Reserve:** Samples that satisfy neither of the above conditions are pushed to the rear end of the stack for future use.

We explain each of these strategies in detail next:

- **Strategy (a) Sample Deletion:** If the sample delete criteria given by:

$$M_t^e < E_d^M \quad \text{AND} \quad \phi_t^e < E_d^\phi \quad (21)$$

is satisfied, then the sample is deleted from the training data set. Here, E_d^M and E_d^ϕ are the magnitude and phase delete thresholds, respectively. They are usually set based on the desired accuracy.

- **Strategy (b) Sample Learning:** Samples are used to either add a new hidden neuron or to update the parameters of the network. Thus, sample learning strategy comprises of the neuron addition strategy and the parameter update strategy.
 - **Neuron addition strategy:** As the training samples arrive sequentially, some of the selected samples will be used to 'add' new hidden neurons based on the following criterion

$$M_t^e \geq E_a^M \quad \text{OR} \quad \phi_t^e \geq E_a^\phi \quad (22)$$

where E_a^M is the neuron growing magnitude threshold and E_a^ϕ is the neuron growing phase threshold. These thresholds are not constants, but are self-regulated such that samples with higher error are learnt first, followed by samples with lower error. The self-regulation occurs according to:

$$\begin{aligned} \text{If } M_t^e \geq E_a^M, \quad E_a^M &= \delta E_a^M - (1 - \delta) M_t^e \\ \text{If } \phi_t^e \geq E_a^\phi, \quad E_a^\phi &= \delta E_a^\phi - (1 - \delta) \phi_t^e \end{aligned} \quad (23)$$

where δ is the slope at which the thresholds are self-regulated. Larger value of δ results in a slow decay of the thresholds from their initial values. This helps fewer samples with significant information to be learnt first, and samples containing less significant information to be learnt last. Thus, larger values of δ ensures that the meta-cognitive principles are emulated efficiently. Usually, δ is set close to 1. When a new neuron is added to FCRN, the input parameters (\mathbf{u}_K and \mathbf{v}_K) of the new neuron are initialized randomly and the optimal output weights are computed using the projection based learning algorithm. Accordingly, the following sequence of operations are carried out:

$$\mathbf{A}_{(K+1) \times (K+1)} = \left[\begin{array}{c|c} \mathbf{A}_{K \times K} & \overline{\mathbf{a}_{K+1}} \\ \hline \mathbf{a}_{K+1}^T & a_{K+1, K+1} \end{array} \right] \quad (24)$$

where matrix $\mathbf{A}_{K \times K} \in \mathbb{C}^{K \times K}$ is updated as

$$A_{kp} = A_{kp} + \overline{h_k^t} h_p^t; \quad k, p = 1, \dots, K \quad (25)$$

where h_k^t is the response of the k -th hidden neuron to the t -th sample being used to add the neuron.

The $\mathbb{C}^{K \times 1}$ vector $\mathbf{a}_{K+1} = [a_{1, K+1}, \dots, a_{p, K+1}, \dots, a_{K, K+1}]$ is given by

$$a_{p, K+1} = \sum_{l=1}^t h_{K+1}^l \overline{h_p^l}; \quad p = 1, \dots, K \quad (26)$$

and the complex-valued scalar $a_{(K+1)(K+1)}$ is given by

$$a_{(K+1)(K+1)} = \sum_{l=1}^t h_{K+1}^l \overline{h_{K+1}^l}. \quad (27)$$

Similarly, the dimensionality of the \mathbf{B} matrix is also increased from $K \times n$ to $(K + 1) \times n$ according to

$$\mathbf{B} = \left[\begin{array}{c} \mathbf{B}_{K \times n} \\ \mathbf{b}_{1 \times n} \end{array} \right] \quad (28)$$

where the $K \times n$ matrix is updated with the current sample t according to:

$$\mathbf{B} = \sum_{l=1}^t \ln(\overline{y_k^t}) h_j^t; \quad j = 1, \dots, K, k = 1, \dots, n \quad (29)$$

and the output matrix \mathbf{B} is appended with $1 \times n$ elements corresponding to the $K + 1$ -th neuron as:

$$\mathbf{b} = \sum_{l=1}^t \ln(\overline{y_k^t}) h_{K+1}^l; \quad k = 1, \dots, n. \quad (30)$$

Finally, the output weights are estimated as

$$\left[\begin{array}{c} \mathbf{w}_K \\ \mathbf{w}_{K+1} \end{array} \right] = \left[\begin{array}{c} \mathbf{B}_{K \times n} \\ \mathbf{b}_{1 \times n} \end{array} \right]^{-1} \left[\begin{array}{c|c} \mathbf{A}_{K \times K} & \overline{\mathbf{a}_{K+1}} \\ \hline \mathbf{a}_{K+1}^T & a_{K+1, K+1} \end{array} \right]^{-1}. \quad (31)$$

- **Parameter update strategy:** When a sample contains significant information that is not novel, but is less familiar to FCRN, the output weights of the network are updated according to Eq. (16). The parameter update criterion is given by:

$$M_t^e \geq E_l^M \quad \text{OR} \quad \phi_t^e \geq E_l^\phi \quad (32)$$

where E_l^M and E_l^ϕ are the parameter update magnitude and phase thresholds, respectively. Similar to the neuron addition thresholds, the parameter update thresholds are also self-regulated according to:

$$\begin{aligned} \text{If } M_t^e \geq E_l^M, \quad E_l^M &= \delta E_l^M - (1 - \delta) M_t^e \\ \text{If } \phi_t^e \geq E_l^\phi, \quad E_l^\phi &= \delta E_l^\phi - (1 - \delta) \phi_t^e. \end{aligned} \quad (33)$$

It is intuitive that the initial values of the self-regulating parameter update thresholds are lesser than their respective neuron add thresholds. It is usually set at a value smaller than the smallest value the neuron addition thresholds can achieve after self-regulation. For a complete guideline on the initialization of the various self-regulating thresholds of the meta-cognitive component, one must refer to (Suresh et al., 2011).

- **Sample Reserve Strategy:** If the current observation ($\mathbf{z}^t, \mathbf{y}^t$) does not satisfy the sample deletion criterion or the neuron growing criterion or the parameter update criterion, then the sample is pushed to the rear end of the data stream. Due to the self-adaptive nature of the thresholds, these reserve samples may also contain some useful information and will be used later in the learning process.

These three control strategies of the meta-cognitive component are repeated for all the samples in the training process and help to improve the generalization performance of FCRN as will be shown in Section 3.

The learning algorithm of McFCRN is summarized as:

Pseudocode 1 Pseudo code for McFCRN algorithm

Given the training data set:
 $(\mathbf{z}^1, \mathbf{y}^1), \dots, (\mathbf{z}^t, \mathbf{y}^t), \dots, (\mathbf{z}^N, \mathbf{y}^N)$

START

For each input,
 Compute the network output $\hat{\mathbf{y}}^t$ using Eq. (2).
 Compute M_t^e and ϕ_t^e using Eqs. (18) and (19), respectively.
 If $M_t^e < E_d^M$ AND $\phi_t^e < E_d^\phi$
 Delete the sample.
 Elseif $M_t^e \geq E_a^M$ OR $\phi_t^e \geq E_a^\phi$
 Add a neuron. $K = K+1$;
 Choose the center and scaling factor of the K -th neuron (\mathbf{v}_K and \mathbf{u}_K) randomly.
 Compute the hidden layer responses h_j^t using Eq. (1).
 Compute the square matrix A according to Eq. (14).
 Compute the matrix B using Eq. (15).
 Estimate the optimum output weights (\mathbf{w}_K) using Eq. (16).
 Elseif $M_t^e \geq E_l^M$ OR $\phi_t^e \geq E_l^\phi$
 Update the optimum output weight of the network using Eq. (16).
 Else
 Reserve the sample for future use.

END

3. Performance evaluation of McFCRN: a complex-valued function approximation problem

In this section, we evaluate the complex-valued function approximation performance of McFCRN in comparison to other

Table 1

Performance comparison for the function approximation problem.

Algorithm	K	Train time (s)	Training error		Testing error	
			J_{Me}	Φ_e	J_{Me}	Φ_e
FC-MLP	15	1857	0.029	15.74	0.054	15.6
C-ELM	15	0.2	0.192	90	0.23	88.2
CMRAN	14	52	0.026	2.23	0.48	18.7
C-RBF	15	9686	0.15	51	0.18	52
FC-RBF	20	1910	0.02	15.9	0.05	15.8
FCRN	10	0.42	0.03	1.38	0.06	3.22
McFCRN	10	10.5	0.04	1.5	0.041	2.98

complex-valued networks available in the literature. The approximation ability of McFCRN is studied using a synthetic complex-valued function approximation problem (Savitha et al., 2009). The synthetic complex-valued function approximation problem is defined in Savitha et al. (2009) as:

$$f(\mathbf{z}) = \frac{1}{1.5} \left(z_3 + 10z_1z_4 + \frac{z_2^2}{z_1} \right) \quad (34)$$

where \mathbf{z} is a 4-dimensional complex-valued vector, z_1, z_2, z_3 and z_4 are complex-valued variables of magnitude less than 2. The study was conducted with a training set with 3000 samples and a testing set with 1000 samples. The root mean squared magnitude error (J_{Me}) (Eq. (35)) and the average absolute phase error (Φ_e) (Eq. (36)), as defined in Savitha et al. (2009), are used as the performance measures for the complex-valued function approximation problems.

$$J_{Me} = \sqrt{\frac{1}{N \times n} \sum_{t=1}^N \left[\sum_{l=1}^n (e_l^t \cdot \bar{e}_l^t) \right]} \quad (35)$$

$$\Phi_e = \frac{1}{N \times n} \sum_{t=1}^N \left[\sum_{k=1}^n \left| \arg \left(y_k^t \bar{\hat{y}}_k^t \right) \right| \right] \times \frac{180}{\pi}. \quad (36)$$

Performance of McFCRN is compared against the best performing results of other complex-valued learning algorithms available in the literature. The algorithms used for comparison are: Fully Complex-valued Multi-layer Perceptron (FC-MLP) (Kim & Adali, 2002), Complex-valued Extreme Learning Machine (C-ELM) (Li et al., 2005), Complex-valued Radial Basis Function network (CRBF) (Chen et al., 1994), the Complex-valued Minimal Resource Allocation Network (Deng, Sundararajan, & Saratchandran, 2002), Fully Complex-valued Radial Basis Function network (FC-RBF) (Savitha et al., 2009) and Fully Complex-valued Relaxation Network (FCRN) (Suresh et al., 2011). For FC-MLP network, *asinh* activation function is used at the hidden layer. The Gaussian activation function is used at the hidden layer of C-ELM, CRBF and CMRAN learning algorithms. For FC-RBF, FCRN, and McFCRN the 'sech' activation function is used at the hidden layer. The number of neurons used for approximation, the training time, the root mean squared magnitude error and the average absolute phase error of McFCRN in comparison with the other learning algorithms is presented in Table 1.

Comparing the training and testing performances of the various complex-valued learning algorithms presented in Table 1, McFCRN has the lowest generalization magnitude and phase errors. It can also be observed that McFCRN requires only 10 neurons to approximate the function. Although FCRN also requires only 10 neurons, its testing magnitude and phase errors are slightly greater than those of McFCRN. This is because the meta-cognitive component of McFCRN helps to delete similar samples, thereby, improving its generalization performance.

4. Performance study: real-valued classification problems

Recent research studies have shown that the complex-valued neurons have better computational power than the real-valued neurons (Nitta, 2003) and the complex-valued neural networks are better decision makers than the real-valued networks. The better decision making ability of complex-valued neural networks is attributed to the presence of two decision boundaries that are orthogonal to each other as shown by Nitta (2004). In this section, we evaluate the classification performance of McFCRN using a set of benchmark real-valued classification problems from the UCI machine learning repository (Blake & Merz, 1998) and a practical mammogram classification for breast cancer detection (Suckling et al., 1994).

4.1. The real-valued classification problem in the Complex domain

Consider $\{(\mathbf{x}^1, c^1), \dots, (\mathbf{x}^t, c^t), \dots, (\mathbf{x}^N, c^N)\}$, where $\mathbf{x}^t = [x_1^t \dots x_j^t \dots x_m^t]^T \in \mathbb{R}^m$ are a set of N observations belonging to n distinct classes, where \mathbf{x}^t is the m -dimensional real-valued input features of t -th observation and $c^t \in \{1, 2, \dots, n\}$ is its class label.

Solving the real-valued classification problem in the Complex domain requires that the real-valued input features be mapped onto the Complex space ($\mathbb{R}^m \rightarrow \mathbb{C}^m$) and the class labels are coded into the Complex domain. In a Multi-Layered network with Multi-Valued Neuron (MLMVN) (Aizenberg & Moraga, 2007), a multiple-valued threshold logic to map the complex-valued input to n discrete outputs using a piecewise continuous activation function (n is the total number of classes). Thus, the transformation does not perform one-to-one mapping of the real-valued input features to the Complex domain, and might cause misclassification. In a Phase Encoded Complex Valued Neural Network (PE-CVNN) (Amin & Murase, 2009), the complex-valued input features are obtained by phase encoding the real-valued input features between $[0, \pi]$ using the transformation $z^t = \exp(i\pi x^t)$, where x^t are the real-valued input features normalized in $[0, 1]$. Recently, a Fully Complex-valued Radial Basis Function (FC-RBF) classifier (Savitha, Suresh, Sundararajan, & Kim, 2011) and a fast learning Phase Encoded Complex-valued Extreme Learning Machine (PE-CELM) classifier (Savitha, Suresh, Sundararajan, & Kim, 2011) have been developed using the phase encoded transformation to convert the real-valued input features to the Complex domain. However, the phase encoded transformation maps the real-valued input features onto the unit circle in the I and II quadrants of the Complex plane, completely ignoring the other two quadrants and other regions in the I and II quadrants. Therefore, the transformation used in the PE-CVNN does not completely exploit the advantages of the orthogonal decision boundaries.

To overcome the issues due to the transformation, a nonlinear transformation using *asinh* function has been proposed in Savitha, Suresh, and Sundararajan (2011). Although this transformation also maps the real-valued features to the I and II quadrants, the region of the complex-valued input space is not restricted to the unit circle. However, this transformation also ignores the other two quadrants of the Complex plane and does not effectively use the advantages of the orthogonal decision boundaries. In this paper, the complex-valued input features ($\mathbf{z}^t = [z_1^t \dots z_j^t \dots z_m^t]^T$) are obtained by using a circular transformation (Savitha, Suresh, & Sundararajan, 2012). The circular transformation for the j -th feature of the t -th sample is given by:

$$z_j^t = \sin(ax_j^t + bx_j^t + \alpha_j); \quad j = 1, \dots, m \quad (37)$$

where $a, b \in [0, 1]$ are randomly chosen scaling constants and $\alpha_j \in [0, 2\pi]$ is used to shift the origin to enable effective usage of the four quadrants of the Complex plane. As the circular

Table 2

Description of the various real-valued classification problems used in the performance study.

Type of data set	Prob.	No. of features	No. of classes	No. of samples		I.F.	
				Train	Test	Train	Test
Multi-Categ.	Image seg.	19	7	210	2100	0	0
	Vehicle class.	18	4	424	422	0.1	0.12
	Glass iden.	9	6	109	105	0.68	0.73
Binary	Liver dis.	6	2	200	145	0.17	0.145
	PIMA data	8	2	400	368	0.225	0.39
	Breast cancer	9	2	300	383	0.26	0.33
	Heart dis.	14	2	70	200	0.14	0.1

transformation performs one-to-one mapping of the real-valued input features to the Complex domain, it uses all the four quadrants of the Complex domain effectively and overcomes the issues due to transformation in the existing complex-valued classifiers.

The coded class label in the Complex domain $\mathbf{y}^t = [y_1^t \cdots y_l^t \cdots y_n^t]^T \in \mathbb{C}^n$ is given by

$$y_l^t = \begin{cases} 1 + 1i & \text{if } c^t = l \\ -1 - 1i & \text{otherwise} \end{cases} \quad l = 1, \dots, n. \quad (38)$$

The classification problem in the Complex domain is defined as: Given the training data set, $\{(\mathbf{z}^1, \mathbf{y}^1), \dots, (\mathbf{z}^t, \mathbf{y}^t), \dots, (\mathbf{z}^N, \mathbf{y}^N)\}$, estimate the decision function ($F: \mathbb{C}^m \rightarrow \mathbb{C}^n$) to enable an accurate prediction of the class labels of unseen samples. The predicted class label of the t -th sample (\hat{c}^t) is obtained from the predicted output of the network ($\hat{\mathbf{y}}^t$) as

$$\hat{c}^t = \max_{l=1,2,\dots,n} \text{Re}(\hat{y}_l^t). \quad (39)$$

4.2. Data sets

The decision making performance of McFCRN is evaluated using a set of benchmark classification problems from the UCI machine learning repository (Blake & Merz, 1998). Table 2 presents the details of the various benchmark data sets including the number of features, and number of samples used in training/testing data sets used in the study. The availability of small number of samples, the sampling bias, and the overlap between classes introduce additional complexity in the classification and may affect the classification performance of the classifier (Suresh, Sundararajan, & Saratchandran, 2008). To study the effect of these factors, we also consider the Imbalance Factor (I.F.) of the training and testing data sets, defined as:

$$I.F. = 1 - \frac{n}{N} \min_{l=1,\dots,n} N_l \quad (40)$$

where N_l is the number of samples belonging to a class l . Note that $N = \sum_{l=1}^n N_l$. The imbalance factor gives a measure of the sample imbalance in the various classes of the training data set, and the imbalance factor of each data set considered in this study is also presented in Table 2. The training data set of the image segmentation problem is a balanced data set, while the remaining data sets are unbalanced in nature. As it can be observed from the table, the imbalance factors of the training data sets vary widely in the range from 0.1 to 0.68.

4.3. Modifications in McFCRN learning algorithm to solve real-valued classification problems

The learning algorithm of McFCRN described in Section 2 has been developed to solve complex-valued function approximation problems. Although they can also be used to approximate the decision surface to solve real-valued classification problems, the classification performance is affected by the definition of error and

the sample learn criteria. Hence, we modify the meta-cognitive component to improve the classification performance of McFCRN. In this respect, the predicted output in Eq. (2) is replaced to accommodate the hinge loss function and the criteria for neuron addition/parameter update is modified to incorporate a classification measure also.

Hinge loss error function: Recently, it was shown in Zhang (2003) and Suresh et al. (2008) that in real-valued classifiers, the hinge loss function helps the classifier to estimate the posterior probability more accurately than the mean squared error function. Hence, in this paper, we modify the error function defined in Eq. (6) as:

$$J = \begin{cases} 0, & \text{if } \text{Real}(y_l^t \hat{y}_l^t) \geq 1 \\ \ln\left(\frac{y_l^t}{\hat{y}_l^t}\right) \ln\left(\frac{\bar{y}_l^t}{\bar{\hat{y}}_l^t}\right), & \text{otherwise} \end{cases} \quad l = 1, 2, \dots, n. \quad (41)$$

Criteria for learning: While solving real-valued classification problems in the Complex domain, it is mandatory that the class labels are predicted accurately. Hence, we have modified the neuron addition and parameter update criteria are modified to ensure accurate prediction of the class labels. Accordingly, the neuron addition criteria is modified as:

$$\text{If } \hat{c}^t \neq c^t \quad \text{OR} \quad (M_t^e \geq E_a^M \text{ AND } \phi_t^e \geq E_a^\phi). \quad (42)$$

Add a neuron to the network. Choose the neuron center (\mathbf{u}_K) and the scaling factor (\mathbf{v}_K) randomly and compute the optimum output weights according to Eq. (16). Here, the predicted class labels are estimated using Eq. (39).

The parameter update criteria is modified as

$$\text{If } \hat{c}^t \neq c^t \quad \text{OR} \quad (M_t^e \geq E_t^M \text{ AND } \phi_t^e \geq E_t^\phi). \quad (43)$$

Then, update the output weight according to Eq. (16).

4.4. Performance measures

The following performance measures are used to evaluate the classification performance of FCRN in comparison to other complex-valued learning algorithms on the problems presented in Table 2.

Average classification efficiency: The average classification efficiency (η_a) is defined as the average ratio of number of correctly classified samples in each class, to the total number of samples in each class.

$$\eta_a = \frac{1}{n} \sum_{l=1}^n \frac{q_{ll}}{N_l} \times 100\% \quad (44)$$

where q_{ll} is the total number of correctly classified samples in the training/testing data set.

Overall classification efficiency: The overall classification efficiency (η_o) is defined as the ratio of total number of correctly classified

Table 3

Performance comparison for the multi-category classification problems.

Problem	Classifier domain	Learning model	No. of neurons	Training time (sec.)	Testing η	
					η_o	η_a
Image segmentation	Real-valued	SVM	127	721	91.38	91.38
		ELM	49	0.25	90.23	90.23
		SRAN	48	22	93	93
	Complex-valued	FC-RBF	38	421	92.33	92.33
		BB-CELM	65	0.03	92.5	92.5
		PE-CELM	75	0.03	92.1	92.1
		FCRN	70	0.4	93.3	93.3
		McFCRN	60	14.23	93.8	93.8
Vehicle classification	Real valued	SVM	340	550	70.62	68.51
		ELM	150	0.4	77.01	77.59
		SRAN	113	55	75.12	76.86
	Complex valued	FC-RBF	70	678	77.01	77.46
		BB-CELM	100	0.11	80.3	80.4
		PE-CELM	100	0.11	80.8	81.1
		FCRN	90	0.8	82.62	82.46
		McFCRN	90	19.8	83.2	83.4
Glass identification	Real valued	SVM	183	320	70.4	75.61
		ELM	80	0.05	81.31	87.43
		SRAN	59	28	86.2	80.95
	Complex valued	FC-RBF	90	452	83.76	80.95
		BB-CELM	70	0.08	88.16	81
		PE-CELM	70	0.08	86.35	80
		FCRN	80	0.25	94.5	88.3
		McFCRN	80	16.22	97.7	92.2

samples to the total number of samples available in the training/testing data set.

$$\eta_o = \frac{\sum_{l=1}^n q_{ll}}{N} \times 100\%. \quad (45)$$

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

4.5. Multi-category benchmark classification problems

First, to study the effect of meta-cognition on a simple problem, we studied the performance of McFCRN on the IRIS classification problem. In the IRIS classification problem, 4 input features are used to classify the samples into one of the 3 classes. McFCRN achieves an overall testing classification efficiency of 98.1% with 7 hidden neurons. Also, it is observed during the training process that of the total 45 training samples, 7 samples are used to add a neuron and the remaining 38 samples are deleted during the training process. On the other hand, SVM classifier uses all the training samples and 25 neurons to achieve an overall testing efficiency of 96.19%. Thus, the effect of meta-cognition is clearly evident in the IRIS classification problem.

Next, the performance results of McFCRN is studied in comparison with other complex-valued classifiers and a few best performing real-valued classifiers on the three multi-category benchmark classification problems. Support Vector Machines and Self-adaptive Resource Allocation Network (SRAN) (Suresh et al., 2010) are the two real-valued classifiers used for comparison. The complex-valued classifiers used in comparison are Phase Encoded Complex-valued Extreme Learning Machine (PE-CELM), Bilinear Branch-cut Complex-valued Extreme Learning Machine (BB-CELM) (Savitha et al., 2011), FC-RBF (Savitha et al., 2011), CSRAN (Suresh et al., 2011) and FCRN (Suresh et al., 2011). The

results for SVM and SRAN classifiers are reproduced from (Suresh et al., 2010). The performance results for PE-CELM, and BB-CELM are reproduced from (Savitha et al., 2011) and those of FC-RBF classifier from (Savitha et al., 2011). The number of neurons used in the classification, and the testing classification accuracies of McFCRN in comparison with the aforementioned classifiers for the multi-category benchmark classification problems are presented in Table 3.

From the table, it can be observed that the generalized performance of McFCRN is better than other real and complex-valued classifiers available in the literature. Also, the computational effort required to train McFCRN is significantly less. McFCRN evidently outperforms the real-valued classifiers used in comparison, especially in the unbalanced vehicle classification and the glass identification data sets. The following observations are notable from the performance results on the multi-category benchmark data sets presented in Table 3:

- *Balanced data set—image segmentation problem:* McFCRN uses only 194 samples of the total 210 samples from the training data set to learn the decision surface described by the image segmentation problem. Moreover, it requires only a fewer neurons and a slightly improved generalization performance.
- *Unbalanced data set—vehicle classification problem:* The meta-cognitive component of McFCRN uses only 572 of the total 620 samples to approximate the decision surface represented by the vehicle classification data set. It is observable from the table that the meta-cognitive component improves the generalization ability of FCRN by at least 1%.
- *Unbalanced data set—glass identification problem:* In the glass identification problem with highly unbalanced data set, McFCRN deleted 18 samples of the total 336 training samples during the training process. Of the remaining 318 samples, 80 samples are used to add neurons, and 238 samples are used in parameter update. It can also be observed from Table 3 that the meta-cognitive component improves the generalization ability of FCRN at least by 3%.

Table 4

Performance comparison on benchmark binary classification problems.

Problem	Classifier domain	Classifier	K	Training time (s)	Testing efficiency (η_o)
Liver disorders	Real-valued	SVM	141	0.0972	71.03
		ELM	100	0.1685	72.41
		SRAN	91	3.38	66.9
	Complex-valued	FC-RBF	20	133	74.46
		BB-CELM	15	0.06	75.17
		PE-CELM	10	0.05	75.86
		FCRN	10	0.05	75.86
		McFCRN	10	1.3	76.55
	Real-valued	SVM	221	0.205	77.45
		ELM	100	0.2942	76.63
		SRAN	97	12.24	78.53
PIMA data	Real-valued	FC-RBF	20	130.3	78.53
		BB-CELM	10	0.15	78.8
		PE-CELM	5	0.08	78.53
	Complex-valued	FCRN	15	0.125	80.71
		McFCRN	15	1.46	81.82
	Real-valued	SVM	24	0.1118	96.6
		ELM	66	0.1442	96.35
		SRAN	7	0.17	96.87
Breast cancer	Real-valued	FC-RBF	10	158.3	97.12
		BB-CELM	15	0.06	92.69
		PE-CELM	15	0.09	97.13
	Complex-valued	FCRN	15	0.16	97.4
		McFCRN	15	1.23	97.65
	Real-valued	SVM	42	0.038	75.5
		ELM	36	0.15	76.5
		SRAN	28	0.534	78.5
Heart Disease	Real-valued	FC-RBF	20	45.6	78
		BB-CELM	5	0.03	83
		PE-CELM	5	0.02	83.5
	Complex-valued	FCRN	10	0.03	84.5
		McFCRN	10	0.16	85.5

4.6. Binary classification problems

We study the classification performance of McFCRN classifier using the benchmark binary classification problems described in Table 2. The performance of McFCRN is compared with the real-valued Support Vector Machines (SVM), Extreme Learning Machine (ELM), SRAN classifiers and the complex-valued FC-RBF classifier (Savitha et al., 2011). The performance results of McFCRN classifier on the binary classification problems in comparison to these classifiers are presented in Table 4.

From the table, it can be observed that McFCRN classifier outperforms all the other real-valued/complex-valued classifiers used in this study. Although FC-RBF is also a fully complex-valued classifier with the *sech* activation function in the hidden layer, the energy function used in McFCRN helps it to perform better than FC-RBF classifier. Moreover, the meta-cognitive component of McFCRN improves the generalization ability of FCRN classifier.

4.7. Performance study on practical real-valued classification problems

In this section, we study the decision making ability of McFCRN on a practical mammogram classification problem for breast cancer detection. Mammogram is a non-invasive procedure that is preferred for early diagnosis of breast cancer as tumors and abnormalities show up in mammogram much before they can be detected through physical examinations (Gamigami, 1996). Clinically, detection of malignant tissues involves identifying the abnormal masses or tumors, if any, and then classifying the mass as either malignant or benign (Aize, Song, Yang, Liu, & Guo, 2004). Classification of the abnormal mass involves the removal of cells

Table 5

Performance comparison results for the mammogram problem.

Classifier domain	Classifier	K	Testing	
			η_o	η_a
Real-valued	ELM	30	91	91
	SVM (Subashini et al., 2010)	–	–	95.44
Complex-valued	FCRN	40	100	100
	McFCRN	30	100	100

or tissue from patients. A machine learning based non-invasive method of identifying the abnormalities in a mammogram can reduce the number of unnecessary biopsies, thus sparing the patients of inconvenience and saving medical costs.

We use the mammogram database available in Suckling et al. (1994) in this study. Classification of an abnormal mass as malignant or benign is *d* based on 9 input features extracted from the mammogram of the identified abnormal mass. McFCRN classifier is trained with 97 samples and tested using remaining 11 samples. For further details on the input features and the data set, one should refer to (Suckling et al., 1994).

The performance results of McFCRN classifier is studied in comparison with other results available in the literature for this problem in Table 5. From the table, it is seen that McFCRN classifier outperforms the other classifiers with an overall classification accuracy of 100%. Although FCRN also presents a classification accuracy of 100%, it requires at least 10 extra neurons to learn the decision surface, while McFCRN has achieves a similar classification performance with fewer neurons.

5. Conclusions

We have presented a Meta-cognitive Fully Complex-valued Relaxation Network (McFCRN) that has two components, namely, the cognitive and meta-cognitive component. FCRN is the cognitive component of McFCRN and a self-regulatory learning mechanism is the meta-cognitive component of McFCRN. FCRN that is the cognitive component of McFCRN uses a logarithmic error function that is an explicit representation of both magnitude and phase of the complex-valued errors to enable accurate approximations. For random constant input and hidden layer parameters, FCRN estimates the unique and optimum output weights corresponding to the minimum energy point of the logarithmic error function using a projection based batch learning algorithm. The projection based learning algorithm of FCRN begins with zero hidden neurons and builds a minimal network architecture to approximate the function defined by the training data set. Thus FCRN requires lesser computational effort as the weights are learnt directly by inversion of a nonsingular matrix. The self-regulatory learning mechanism of McFCRN decides *what-to-learn*, *when-to-learn* and *how-to-learn* in a meta-cognitive framework by choosing suitable learning strategies for each sample in the training data set. The two components of McFCRN and their learning algorithm are explained in detail. Performance study on a function approximation problem and a set of real-valued classification problems show the superior performance and computational abilities of FCRN.

Acknowledgments

The authors would like to thank the reviewers for their comments and suggestions to improve the quality of the manuscript. The first and second authors would like to thank the Nanyang Technological University—Ministry of Defence (NTU-MINDEF), Singapore, for the financial support (Grant number: MINDEF-NTU-JPP/11/02/05) to conduct this study.

References

- Aize, C., Song, Q., Yang, X., Liu, S., & Guo, C. (2004). Mammographic mass detection by vicinal support vector machine. In *Proc. of ICNN'04, international conference on neural networks*. Vol. 3 (pp. 1953–1958).
- Aizenberg, I., & Moraga, C. (2007). Multilayer feedforward neural network based on multi-valued neurons (MLMVN) and a backpropagation learning algorithm. *Soft Computing*, 11(2), 169–183.
- Amin, M. F., & Murase, K. (2009). Single-layered complex-valued neural network for real-valued classification problems. *Neurocomputing*, 72(4–6), 945–955.
- Babu, G. S., & Suresh, S. (2012). Meta-cognitive neural network for classification problems in a sequential learning framework. *Neurocomputing*, (12), 2621–2638. doi:10.1016/j.neucom.2011.02.001.
- Blake, C., & Merz, C. (1998). UCI repository of machine learning databases. Department of Information and Computer Sciences. University of California, Irvine [URL: <http://archive.ics.uci.edu/ml/>].
- Chen, S., McLaughlin, S., & Mulgrew, B. (1994). Complex valued radial basis function network, part I: network architecture and learning algorithms. *EURASIP Signal Processing Journal*, 35(1), 19–31.
- Deng, J. P., Sundararajan, N., & Saratchandran, P. (2002). Communication channel equalization using complex-valued minimal radial basis function neural networks. *IEEE Transactions on Neural Networks*, 13(3), 687–696.
- Gamigami, P. (1996). *Atlas of mammography: new early signs in breast cancer*. Cambridge, MA, USA: Blackwell Science.
- Haykin, S. (1998). *Neural networks: a comprehensive foundation*. New Jersey, USA: 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.
- Kim, T., & Adali, T. (2002). Fully complex multi-layer perceptron network for nonlinear signal processing. *Journal of VLSI Signal Processing*, 32(1–2), 29–43.
- Li, M. B., Huang, G.-B., Saratchandran, P., & Sundararajan, N. (2005). Fully complex extreme learning machine. *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 equalisation. *IEEE Proceedings—Vision, Image and Signal Processing*, 153(4), 411–418.
- Nelson, T. O., & Narens, L. (1980). Metamemory: a theoretical framework and new findings. In T. O. Nelson (Ed.), *Metacognition: core readings* (pp. 9–24). Boston: Allyn and Bacon (Chapter).
- Nitta, T. (2003). The computational power of complex-valued neuron. In *Lecture notes in computer science: Vol. 2714. Artificial neural networks and neural information processing ICANN/ICONIP* (pp. 993–1000).
- Nitta, T. (2004). Orthogonality of decision boundaries of complex-valued neural networks. *Neural Computation*, 16(1), 73–97.
- Remmert, R. (1991). *Theory of complex functions*. New York, USA: Springer Verlag.
- Rivers, W. P. (2001). Autonomy at all costs: an ethnography of meta-cognitive self-assessment and self-management among experienced language learners. *The 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 *IEEE international joint conference on neural networks, 2010, IJCNN 2010* (pp. 1–8).
- Savitha, R., Suresh, S., & Sundararajan, N. (2011). A fast learning complex-valued neural classifier for real-valued classification problems. In *International joint conference on neural networks. IJCNN 2011* (pp. 2243–2249).
- Savitha, R., Suresh, S., & Sundararajan, N. (2011). Meta-cognitive learning in fully complex-valued radial basis function network. *Neural Computation*, in press (doi:10.1162/NECO_a_00254).
- Savitha, R., Suresh, S., & Sundararajan, N. (2012). Fast learning circular complex-valued extreme learning machine (CC-ELM) for real-valued classification problems. *Information Sciences*, 187(1), 277–290.
- Savitha, R., Suresh, S., Sundararajan, N., & Kim, H. J. (2011). A fully complex-valued radial basis function classifier for real-valued classification. *Neurocomputing*, 78(1), 104–110.
- Savitha, R., Suresh, S., Sundararajan, N., & Kim, H. J. (2011). In D. Liu, et al. (Eds.), *Lecture Notes in Computer Science (LNCS): Vol. 6675. Fast learning fully complex-valued classifiers for real-valued classification problems* (pp. 602–609). ISSN 2011 (Part I).
- 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.
- Subashini, T. S., Ramalingam, V., & Palanivel, S. (2010). Automated assessment of breast tissue density in digital mammograms. *Computer Vision and Image Understanding*, 114(1), 33–43.
- Suckling, J., et al. (1994). The mammographic image analysis society digital mammogram database. *Excerpta Medica International Congress Series*, 1069, 375–378.
- Suresh, S., Dong, K., & Kim, H. J. (2010). A sequential learning algorithm for self-adaptive resource allocation network classifier. *Neurocomputing*, 73(16–18), 3012–3019.
- Suresh, S., Savitha, R., & Sundararajan, N. (2011). A fast learning fully complex-valued relaxation network (FCRN). In *IEEE international joint conference on neural networks, 2011. IJCNN 2011* (pp. 1372–1377).
- Suresh, S., Savitha, R., & Sundararajan, N. (2011). A sequential learning algorithm for complex-valued self-regulating resource allocation network-CSRN. *IEEE Transactions on Neural Networks*, 22(7), 1061–1072.
- Suresh, S., & Subramanian, K. (2011). A sequential learning algorithm for meta-cognitive neuro-fuzzy inference system for classification problems. In *IEEE international joint conference on neural networks, 2011. IJCNN 2011* (pp. 2507–2512).
- Suresh, S., Sundararajan, N., & Saratchandran, P. (2008). Risk-sensitive loss functions for sparse multi-category classification problems. *Information Sciences*, 178(12), 2621–2638.
- Wenden, A. L. (1998). Meta-cognitive 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, 357–375.
- Yu, S.-S., & Tsai, W.-H. (1992). Relaxation by the Hopfield neural network. *Pattern Recognition*, 25(2), 197–209.
- Zhang, T. (2003). Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32(1), 56–85.