Our reference: INS 9307 P-authorquery-v10

AUTHOR QUERY FORM

ELSEVIER	**Journal: INS** **Article Number: 9307**	**Please e-mail or fax your responses and any corrections to:** **E-mail: corrections.esch@elsevier.sps.co.in** **Fax: +31 2048 52799**

Dear Author,

Please check your proof carefully and mark all corrections at the appropriate place in the proof (e.g., by using on-screen annotation in the PDF file) or compile them in a separate list. Note: if you opt to annotate the file with software other than Adobe Reader then please also highlight the appropriate place in the PDF file. To ensure fast publication of your paper please return your corrections within 48 hours.

For correction or revision of any artwork, please consult http://www.elsevier.com/artworkinstructions.

Any queries or remarks that have arisen during the processing of your manuscript are listed below and highlighted by flags in the proof. Click on the 'Q' link to go to the location in the proof.

Location in article	**Query / Remark: click on the Q link to go** **Please insert your reply or correction at the corresponding line in the proof**
Q1	Please confirm that given names and surnames have been identified correctly.
Q2	Please provide the significance of bold entities in Tables 2–5.

Thank you for your assistance.

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Fast learning Circular Complex-valued Extreme Learning Machine (CC-ELM) for real-valued classification problems

R. Savitha [a], S. Suresh [b,*], N. Sundararajan [a]

[a] School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore
[b] School of Computer Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Article history:
Received 9 November 2010
Received in revised form 14 September 2011
Accepted 6 November 2011
Available online xxxx

Keywords:
Complex-valued ELM
Orthogonal decision boundaries
Circular function
Acoustic emission and mammogram classification

ABSTRACT

In this paper, we present a fast learning fully complex-valued extreme learning machine classifier, referred to as ‘Circular Complex-valued Extreme Learning Machine (CC-ELM)’ for handling real-valued classification problems. CC-ELM is a single hidden layer network with non-linear input and hidden layers and a linear output layer. A circular transformation with a translational/rotational bias term that performs a one-to-one transformation of real-valued features to the complex plane is used as an activation function for the input neurons. The neurons in the hidden layer employ a fully complex-valued Gaussian-like (‘*sech*’) activation function. The input parameters of CC-ELM are chosen randomly and the output weights are computed analytically. This paper also presents an analytical proof to show that the decision boundaries of a single complex-valued neuron at the hidden and output layers of CC-ELM consist of two hyper-surfaces that intersect orthogonally. These orthogonal boundaries and the input circular transformation help CC-ELM to perform real-valued classification tasks efficiently.

Performance of CC-ELM is evaluated using a set of benchmark real-valued classification problems from the University of California, Irvine machine learning repository. Finally, the performance of CC-ELM is compared with existing methods on two practical problems, viz., the acoustic emission signal classification problem and a mammogram classification problem. These study results show that CC-ELM performs better than other existing (both) real-valued and complex-valued classifiers, especially when the data sets are highly unbalanced.

1. Introduction

Complex-valued neural networks were originally developed for solving problems involving complex-valued signals. According to Liouville’s theorem, an analytic and bounded function is a constant in the complex plane. To overcome this restriction, non-linear, analytic and *almost everywhere* bounded fully complex-valued activation functions [14,25–27,42] have been used as activation functions in the literature. Several applications of complex-valued neural networks operating on complex-valued signals have been reported in the literature (e.g. communication channel equalization [26,7] adaptive array signal processing [6,31,28], image reconstruction [32], etc.). Recently, complex-valued neural networks have been shown to have better computational power than real-valued neural networks [16] and also that they are better in performing real-valued classification tasks because of their inherent orthogonal decision boundaries [18,19].

[*] Corresponding author. Tel.: +65 6790 6185; fax: +65 6792 6559.
E-mail address: ssundaram@ntu.edu.sg (S. Suresh).

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2011.11.003

In [17], Nitta showed that a single complex-valued neuron with an orthogonal decision boundary can be used to solve the XOR problem and the detection of symmetry problem, both of which cannot be solved using a single real-valued neuron. Further, he also showed that the decision boundary of a single complex-valued neuron consists of two hyper surfaces intersecting orthogonally and this improves the generalization ability of a single hidden layered complex-valued network [19].

Classification is one of the most frequently encountered decision making problems in practical applications. The input features in most of these problems are real-valued and there are several real-valued classifiers available in the literature [38,41,44,23]. In addition, entropy minimization/maximization based neural/fuzzy classifiers have also been developed [43,45]. All these real-valued classifiers operate on real-valued features to approximate the decision function defined by the training data set. However, while developing complex-valued classifiers, it is essential to convert real-valued features to the complex domain to exploit the computational ability and the orthogonal decision boundaries of the complex-valued neural networks. This motivated researchers to develop new transformations for the input features resulting in new complex-valued classifiers to solve real-valued classification problems.

Aizenberg et al. [1,2], was the first one to suggest a complex-valued multi-valued neuron to solve real-valued classification problems. For a data set with C classes, a multi-valued neuron maps the complex-valued input to C discrete class labels (multiple-valued threshold logic) using a piecewise continuous activation function. The complex-valued Multi-Layer Multi Valued Network (MLMVN) that employs multi-valued neurons, uses a derivative free global error correcting learning rule to update the network parameters. In MLMVN, the normalized real-valued input features ($x \in [0,1]$) are mapped to a full unit circle using the transformation $exp(i2\pi x)$ ($i = \sqrt{-1}$ is the complex operator) where the class labels are encoded by the roots of unity in the complex plane. However, as the input features are mapped to a full unit circle, this mapping results in the same complex-valued features for real-valued features with values 0 and 1. Hence, this transformation is not unique. In addition, the multi-valued neurons map the complex-valued inputs to C discrete outputs in the unit circle. As number of classes (C) increases, the region of sectors per class within the unit circle decreases, which increases misclassification rate. Furthermore, the output neurons of MLMVN have multiple discrete values and hence, MLMVN classifier lacks the orthogonal decision boundaries that are characteristic of complex-valued neural classifiers.

Recently, Amin et al. [3,4] have developed a Phase Encoded Complex-Valued Neural Network (PE-CVNN) to solve real-valued classification problems. In PE-CVNN, the complex-valued input features are obtained by phase encoding the normalized real-valued input features (x) between $[0,\pi]$, using the transformation '$exp(i\pi x)$'. PE-CVNN uses a gradient-descent based batch learning algorithm that requires a significant computational effort to approximate the decision surface. In addition, the activation functions used in PE-CVNN are similar to those used in the split-type ($f(z) = f(z^R) + if(z^I)$, where z^R and z^I are the real and imaginary parts of z) complex-valued neural networks. Here, the gradients used are not fully complex-valued [14] thereby resulting in a significant loss of complex-valued information. Hence, there is a need to develop a fast learning, fully complex-valued classifier with a fully complex-valued activation function, which exploits the advantages of the orthogonal decision boundaries more efficiently to solve real-valued classification problems. Recently, a fast learning phase encoded complex-valued extreme learning machine has been developed using the phase encoded transformation at the input layer and a fully complex-valued activation function at the hidden layer [29]. However, it was observed from the studies that the phase encoded transformation considers only the first two quadrants of the complex plane and does not fully exploit the advantages of the orthogonal decision boundaries offered by the complex-valued neural networks.

In this paper, we develop such a fast learning fully complex-valued neural classifier referred to as 'Circular Complex-valued Extreme Learning Machine' (CC-ELM). CC-ELM uses a non-linear transformation with a translational/rotational bias term in the input layer which results in a unique transformation (one-to-one mapping from real to complex domain) from the real-valued to the complex-valued feature space. The transformation is called as 'circular transformation'. It uses the '*sine*' function given by $z = \sin(ax + i\,bx + \alpha)$, where α is the non-zero translational/rotational bias term, a and b are real-valued non-zero transformation constants. The α shifts the origin of the transformation and also rotates the resultant complex-valued inputs to any of the four quadrants in the complex plane.

CC-ELM is a single hidden layer neural network with the circular transformation as an activation function in the input layer, a fully complex-valued '*sech*' activation function (presented in [26,30]) in the hidden layer and a linear activation function in the output layer. Similar to the Complex-valued Extreme Learning Machines (C-ELM) [15], in CC-ELM, the transformation constants, translational/rotational bias terms and the parameters of the input and hidden layer are chosen randomly and the output weights are calculated analytically. We also provide an analytical proof for the existence of the orthogonal decision boundaries in CC-ELM. These boundaries and the unique input transformation help CC-ELM to provide a better classification performance. In addition, as the output weights are calculated analytically, CC-ELM requires significantly lower computational effort to approximate the decision function.

As the nonlinear transformation at the input layer maps real-valued input features to all the four quadrants of the complex plane, this transformation fully exploits the advantages of the orthogonal decision boundaries offered by complex-valued neural networks in classification problems. Also, the nonlinear circular transformation and the *sech* activation function in the hidden layer help in exploiting the advantages of the orthogonal decision boundaries of complex-valued neural networks to provide a better generalization performance. Moreover, as the output weights are computed analytically, CC-ELM network learns the decision function very quickly.

Performance of CC-ELM classifier has been evaluated using a set of benchmark classification problems from the University of California, Irvine (UCI) Machine learning repository [5]. In this performance study, the effects of the orthogonal decision boundaries of the complex-valued classifiers are studied by comparing the performances of CC-ELM classifier with the best

Please cite this article in press as: R. Savitha et al., Fast learning Circular Complex-valued Extreme Learning Machine (CC-ELM) for real-valued classification problems, Inform. Sci. (2011), doi:10.1016/j.ins.2011.11.003

results obtained using the existing real-valued classifiers. Next, to evaluate the benefits of the circular transformation, the performance of CC-ELM is also compared against other complex-valued classifiers, viz., the phase encoded complex-valued neural network (PE-CVNN) [3,?] and the Multi-Layer Multi Valued Network (MLMVN) [1]. The study results clearly show that CC-ELM classifier outperforms both the real-valued and complex-valued classifiers, reported in the literature for these problems. Also, it is seen that CC-ELM exhibits a significant performance improvement when the data set is highly unbalanced.

Finally, the performance of CC-ELM classifier has also been evaluated using two practical classification problems viz., the acoustic emission signal classification problem [20] and the mammogram classification problem for breast cancer detection [34]. The results clearly highlight that CC-ELM classifier provides a better generalization performance than other existing (both) real-valued and complex-valued classifiers.

The paper is organized as follows: Section 2 presents the details of the circular complex-valued extreme learning machine. It also highlights the advantages of the circular input transformation over other input transformations. In Section 3, the proof for the orthogonality of the decision boundaries at the hidden and output layers of CC-ELM network is presented. In Section 4, an elaborate performance comparison of CC-ELM network on a number of benchmark and practical classification problems is presented along with other existing results in the literature. Finally, Section 5 summarizes the main conclusions from this study.

2. Circular Complex-valued Extreme Learning Machine Classifier (CC-ELM)

In this section, a fast learning fully-complex valued classifier called 'circular complex-valued extreme learning machine' is developed to solve real-valued classification tasks. First, we present the formulation of a real-valued classification problem in the complex domain followed by a detailed description of CC-ELM algorithm.

2.1. Real-valued classification problem definition

Suppose we have N observations $\{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_t, c_t), \ldots, (\mathbf{x}_N, c_N)\}$, where $\mathbf{x}_t \in \Re^m$ be the m-dimensional input features of tth observation, $c_t \in [1, 2, \ldots, C]$ is its class label, and C is the number of distinct classes. The observation data $(\mathbf{x}_t, c_t)$ are random in nature and the observation $\mathbf{x}_t$ provides some useful information on the probability distribution over the observation data to predict the corresponding class label (c_t) with certain accuracy.

To solve real-valued classification problems using complex-valued neural networks, the coded class label $(\mathbf{y}^t = [y_1^t \cdots y_k^t \cdots y_C^t]^T)$ is defined in the complex domain as:

$$y_k^t = \begin{cases} 1 + i1, & \text{if } c_t = k, \\ -1 - i1, & \text{otherwise,} \end{cases} \quad k = 1, 2, \ldots, C. \tag{1}$$

The real-valued classification problem using complex-valued neural networks can be viewed as finding the decision function F that maps the real-valued input features to the complex-valued coded class labels, i.e., $F : \Re^m \to \mathbb{C}^C$. For notational convenience, the superscript t will be dropped in the rest of the paper.

In the next section, we present a fast learning CC-ELM classifier to estimate the decision function (F). The detailed description of the algorithm is also presented.

2.2. Circular complex-valued extreme learning machine

The circular complex-valued extreme learning machine classifier is a single hidden layer network with m input neurons, K hidden neurons and C output neurons, as shown in Fig. 1. A non-linear circular transformation as shown in the inset of Fig. 1 is used as the activation function at the input layer.

The circular transformation that transforms the real-valued input features to the Complex domain $(\Re \to \mathbb{C} : \mathbf{x} = [x_1, \ldots, x_m] \to \mathbf{z} = [z_1, \ldots, z_m])$ is given by

$$z_l = \sin(ax_l + ibx_l + \alpha_l), \quad l = 1, 2, \ldots, m \tag{2}$$

where $a, b, \alpha_l \in \Re^+$ are non-zero constants and x_l is the input feature normalized in [0, 1]. The scaling factors a, and b, and the translational, rotational bias term α_l are randomly chosen such that $0 < a, b \leqslant 1$, and $0 < \alpha_l < 2\pi$. The translational, rotational bias term α_l shifts the origin of the resultant complex-valued feature (z_l) and rotates it to any quadrant of the complex domain.

The effect of the circular transformation for different values of the translational, rotational bias term (α_l) is shown in Fig. 2. It can be observed from this figure that the randomly chosen bias terms perform the translation/rotation of the feature vector in different quadrants of the complex-valued feature space. Therefore, the bias term $(\alpha_l, l = 1, \ldots, m)$ associated with each input feature ensures that all the input features are not mapped onto the same quadrant of the complex plane. Thus, as the input features are well distributed in the complex plane, CC-ELM exploits the orthogonal decision boundaries of the fully complex-valued neural networks more efficiently.

The essential properties of a fully complex-valued activation function given in [14] states that for a complex-valued non-linear function to be used as an activation function, the function has to be analytic and bounded *almost everywhere*. As the

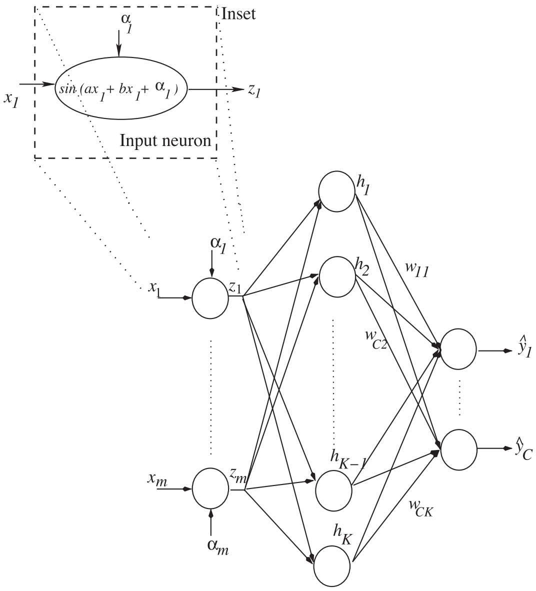

Fig. 1. The architecture of a circular complex-valued extreme learning machine. The expanded box (inset figure) shows the actual circular transformation function which maps real-valued input feature to complex-valued feature.

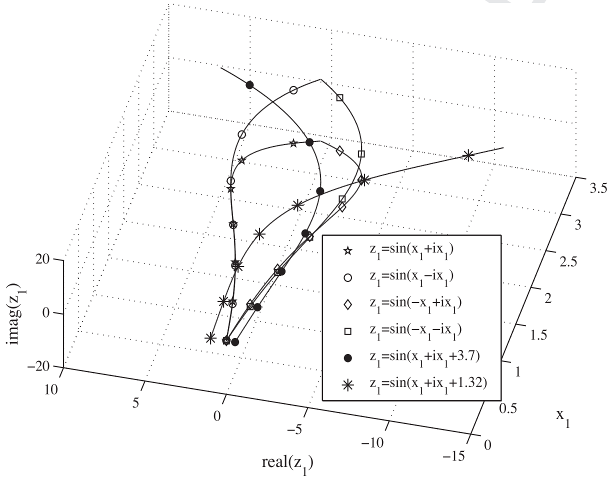

Fig. 2. Significance of translational/rotational bias term (α) in the circular transformation.

circular transformation is used as an activation function in the input layer, we need to ensure that the transformation satisfies the essential properties of a fully complex-valued activation function. The '*sine*' activation function is an analytic function with an essential singularity at ∞. The input to the '*sine*' function

$$ax_l + ibx_l + \alpha_l = \infty \quad \text{if } ax_l + \alpha_l = \infty \quad \text{or } bx_l = \infty. \tag{3}$$

Since the transformation constants (a,b) and the translational/rotational bias (α_l) are restricted between $(0,1]$ and $(0,2\pi)$ respectively, the transformation becomes unbounded only when the input feature is ∞. Since, we use the real-valued features that are normalized between $[0,1]$, the circular transformation is analytic and bounded *almost everywhere*. Hence, the circular transformation used in the input layer of CC-ELM is a valid activation function.

The neurons in the hidden layer of CC-ELM employ a fully-complex valued '*sech*' activation function (Gaussian like) as developed in [26]. The response of the jth hidden neuron (h_j) of CC-ELM is given by

$$h_j = sech\left[\mathbf{u}_j^T(\mathbf{z} - \mathbf{v}_j)\right], \quad j = 1, 2, \ldots, K, \tag{4}$$

where K is the number of hidden neurons, $\mathbf{u}_j \in \mathbb{C}^m$ is the complex-valued scaling factor of the j-th hidden neuron, $\mathbf{v}_j \in \mathbb{C}^m$ is the complex-valued center of the jth hidden neuron and the superscript T represents the matrix transpose operator.

Please cite this article in press as: R. Savitha et al., Fast learning Circular Complex-valued Extreme Learning Machine (CC-ELM) for real-valued classification problems, Inform. Sci. (2011), doi:10.1016/j.ins.2011.11.003

The neurons in the output layer employ a linear activation function. The output ($\hat{\mathbf{y}} = [\hat{y}_1 \cdots \hat{y}_n \cdots \hat{y}_C]^T$) of CC-ELM network with K hidden neurons is

$$\hat{y}_n = \sum_{j=1}^{K} w_{nj} h_j, \quad n = 1, 2, \ldots, C, \tag{5}$$

where w_{ij} is the output weight connecting the jth hidden neuron and the ith output neuron.

The estimated class label ($\hat{c}$) is then obtained using

$$\hat{c} = arg \max_{n=1,2,\ldots,C} \text{ real-part of } \hat{y}_n. \tag{6}$$

The output of CC-ELM given in Eq. (5) can be written in matrix form as,

$$\widehat{Y} = WH, \tag{7}$$

where W is the matrix of all output weights connecting hidden and output neurons. The H is the response of hidden neurons for all training samples and is given as

$$H(V, U, Z) = \begin{bmatrix} sech\left(\mathbf{u}_1^T \|\mathbf{z}_1 - \mathbf{v}_1\|\right) & \cdots & sech\left(\mathbf{u}_1^T \|\mathbf{z}_N - \mathbf{v}_1\|\right) \\ \vdots & \vdots & \vdots \\ sech\left(\mathbf{u}_j^T \|\mathbf{z}_1 - \mathbf{v}_j\|\right) & \cdots & sech\left(\mathbf{u}_j^T \|\mathbf{z}_N - \mathbf{v}_j\|\right) \\ \vdots & \vdots & \vdots \\ sech\left(\mathbf{u}_K^T \|\mathbf{z}_1 - \mathbf{v}_K\|\right) & \cdots & sech\left(\mathbf{u}_K^T \|\mathbf{z}_N - \mathbf{v}_K\|\right) \end{bmatrix}. \tag{8}$$

Note that H is $K \times N$ hidden layer output matrix. The jth row of the H matrix represents the hidden neuron response (h_j) with respect to the inputs ($\mathbf{z}_1, \ldots, \mathbf{z}_N$).

In CC-ELM, the transformation constants (a,b), the translational/rotation bias term ($\boldsymbol{\alpha}$), center (V) and scaling factor (U) are chosen randomly and the output weights W are estimated by the least squares method according to:

$$W = YH^{\dagger}, \tag{9}$$

where $H^{\dagger}$ is the generalized Moore–Penrose pseudo-inverse [22] of the hidden layer output matrix and Y is the complex-valued coded class label.

In short, CC-ELM algorithm can be summarized as:

- For a given training set (X,Y), select the appropriate number of hidden neurons K.
- Randomly select $0 < a,b < 1, 0 < \alpha_l < 2\pi$, the neuron scaling factor (U) and the neuron centers (V).
- Then calculate the output weights W analytically: $W = Y H^{\dagger}$.

Performance of CC-ELM is influenced by the selection of appropriate number of hidden neurons. Recently, an incremental constructive method to determine the appropriate number of hidden neurons for the C-ELM has been presented in [11]. A complete survey of the extreme learning machine algorithms is presented in [13]. In this paper, we use a simple neuron incremental-decremental strategy, similar to the one presented in [35] for real-valued networks. The following steps are followed to select the appropriate number of hidden neurons for CC-ELM network:

Step 1. Select a network with a minimum configuration ($K = m + C$).
Step 2. Select the input weights randomly and compute the output weights analytically.
Step 3. Use leave-one cross-validation to determine training/validation accuracy from the training data.
Step 4. Increase K until the validation accuracy improves and return to Step 2.
Step 5. If the validation accuracy decreases as K increases, then stop.

From this section, it can be observed that the two important properties of CC-ELM classifier are that:

- CC-ELM classifier uses a unique circular transformation that makes a one-to-one mapping while transforming the real-valued input features to the Complex domain ($\Re \rightarrow \mathbb{C}$).
- CC-ELM classifier requires lesser computational effort than other complex-valued classifiers as the weights are computed analytically.

3. Orthogonal decision boundaries in CC-ELM

In this section, we show that the three layered CC-ELM with the fully complex-valued *sech* activation function at the hidden layer exhibits orthogonal decision boundaries. Since CC-ELM maps the complex-valued input features to the higher

Please cite this article in press as: R. Savitha et al., Fast learning Circular Complex-valued Extreme Learning Machine (CC-ELM) for real-valued classification problems, Inform. Sci. (2011), doi:10.1016/j.ins.2011.11.003

dimensional Complex plane in the hidden layer randomly, we prove the existence of orthogonal decision boundaries in the output neuron with respect to the hidden layer output. Next, we also show that the decision boundaries formed by the real and imaginary parts of the hidden layer response with respect to the input are orthogonal to each other.

3.1. Case (i): Orthogonality of decision boundaries in the output layer

The responses of the nth output neuron can be written as:

$$\hat{y}_n = \sum_{j=1}^{K} w_{nj} h_j; \quad n = 1, \ldots, C, \tag{10}$$

$$= \sum_{j=1}^{K} \left(w^R_{nj} + i w^I_{nj}\right)\left(h^R_j + i h^I_j\right), \tag{11}$$

where the superscripts R and I represents the real and imaginary parts of the complex-valued signals, respectively. Therefore,

$$\hat{y}_n = \hat{y}^R_n + i\hat{y}^I_n = \sum_{j=1}^{K} \left(w^R_{nj} h^R_j - w^I_{nj} h^I_j\right) + i\left(w^R_{nj} h^I_j + w^I_{nj} h^R_j\right) \tag{12}$$

From the above equation, we can see that the real-part and the imaginary-part of the nth output neuron forms two decision boundaries with respect to the hidden layer output ($\mathbf{h}$). The two decision boundaries are

$$\hat{y}^R_n = \sum_{j=1}^{K} \left(w^R_{nj} h^R_j - w^I_{nj} h^I_j\right) \to \mathbf{S}^R, \tag{13}$$

$$\text{and } \hat{y}^I_n = \sum_{j=1}^{K} \left(w^R_{nj} h^I_j + w^I_{nj} h^R_j\right) \to \mathbf{S}^I. \tag{14}$$

The real-part decision boundary $\mathbf{S}^R$ classifies the hidden layer output $\mathbf{h} \in \mathbb{C}^K$ into two decision regions

$$\left\{\mathbf{h} \in \mathbb{C}^K, |\hat{y}^R_n \geqslant \mathbf{S}^R\right\} \quad \text{and} \quad \left\{\mathbf{h} \in \mathbb{C}^K, |\hat{y}^R_n < \mathbf{S}^R\right\}. \tag{15}$$

Similarly, the imaginary-part decision boundary $\mathbf{S}^I$ classifies the hidden layer output $\mathbf{h} \in \mathbb{C}^K$ into two decision regions

$$\left\{\mathbf{h} \in \mathbb{C}^K, |\hat{y}^I_n \geqslant \mathbf{S}^I\right\} \quad \text{and} \quad \left\{\mathbf{h} \in \mathbb{C}^K, |\hat{y}^I_n < \mathbf{S}^I\right\}. \tag{16}$$

The normal vector ($Q^R(\mathbf{h}^R, \mathbf{h}^I)$) to the real-part of the decision boundary ($\mathbf{S}^R$) and the normal vector ($Q^I(\mathbf{h}^R, \mathbf{h}^I)$) to the imaginary-part of the decision boundary ($\mathbf{S}^I$) are:

$$Q^R(\mathbf{h}^R, \mathbf{h}^I) = \left(\frac{\partial \hat{y}^R_n}{\partial h^R_1} \cdots \frac{\partial \hat{y}^R_n}{\partial h^R_K} \frac{\partial \hat{y}^R_n}{\partial h^I_1} \cdots \frac{\partial \hat{y}^R_n}{\partial h^I_K}\right) = (w^R_{n1} \cdots w^R_{nK} - w^I_{n1} \cdots - w^I_{nK}), \tag{17}$$

$$Q^I(\mathbf{h}^R, \mathbf{h}^I) = \left(\frac{\partial \hat{y}^I_n}{\partial h^R_1} \cdots \frac{\partial \hat{y}^I_n}{\partial h^R_K} \frac{\partial \hat{y}^I_n}{\partial h^I_1} \cdots \frac{\partial \hat{y}^I_n}{\partial h^I_K}\right) = (w^I_{n1} \cdots w^I_{nK} w^R_{n1} \cdots w^R_{nK}). \tag{18}$$

The decision boundaries ($\mathbf{S}^R$ and $\mathbf{S}^I$) of the nth output neuron are orthogonal *iff* the dot product of their normal vectors is zero.

$$Q^R(\mathbf{h}^R, \mathbf{h}^I).Q^I(\mathbf{h}^R, \mathbf{h}^I) = \begin{pmatrix} w^R_{n1}.w^I_{n1} + \cdots + w^R_{nK}.w^I_{nK} \\ -w^I_{n1}.w^R_{n1} + \cdots - w^I_{nK}.w^R_{nK} \end{pmatrix} \tag{19}$$

$$= 0. \tag{20}$$

From the above results, we can say that any output neuron in CC-ELM has two decision boundaries with respect to the hidden neuron output and they are orthogonal to each other.

3.2. Case (ii): Orthogonality of decision boundaries in the hidden layer

The response of the jth neuron in the hidden layer can be written as

$$h_j = h^R_j + i h^I_j = sech(O_j),$$

$$\text{where } O_j = \mathbf{u}^T_j(\mathbf{z} - \mathbf{v}_j), \quad j = 1, 2, \ldots, K, \tag{21}$$

$$O_j = O^R_j + i O^I_j = \left(\mathbf{u}^R_j + i\mathbf{u}^I_j\right)^T \left[\left(\mathbf{z}^R - \mathbf{v}^R_j\right) + i\left(\mathbf{z}^I - \mathbf{v}^I_j\right)\right] \tag{22}$$

$$= \left[\mathbf{u}^R_j\left(\mathbf{z}^R - \mathbf{v}^R_j\right) - \mathbf{u}^I_j\left(\mathbf{z}^I - \mathbf{v}^I_j\right)\right] + i\left[\mathbf{u}^R_j\left(\mathbf{z}^I - \mathbf{v}^I_j\right) + \mathbf{u}^I_j\left(\mathbf{z}^R - \mathbf{v}^R_j\right)\right]. \tag{23}$$

Please cite this article in press as: R. Savitha et al., Fast learning Circular Complex-valued Extreme Learning Machine (CC-ELM) for real-valued classification problems, Inform. Sci. (2011), doi:10.1016/j.ins.2011.11.003

Using trigonometric and hyperbolic trigonometric definitions, the *sech* function can be written as:

$$sech(O_j^R + iO_j^I) = \frac{2(\cos(O_j^I \cosh(O_j^R) - i\sin(O_j^I)\sinh(O_j^R))}{\cos(2O_j^I) + \cosh(2O_j^I)}. \tag{24}$$

The two decision boundaries formed by the real and imaginary part of the *j*th hidden neuron response with respect to the inputs are:

$$h_j^R = \frac{2(\cos(O_j^I \cosh(O_j^R)))}{\cos(2O_j^I) + \cosh(2O_j^I)} \to \mathbf{S}^R, \tag{26}$$

$$h_j^I = \frac{2(-i\sin(O_j^I)\sinh(O_j^R))}{\cos(2O_j^I) + \cosh(2O_j^I)} \to \mathbf{S}^I. \tag{27}$$

Note that here the decision boundaries are shown with respect to the net input to the hidden neuron O_j, which is a linear function of actual input **z**.

Hence, the complex-valued input $\mathbf{z} \in \mathbb{C}^m$ are classified into two decision regions with respect to the real and imaginary part ($\mathbf{S}^R$ and $\mathbf{S}^I$)

$$\left\{\mathbf{z} \in \mathbb{C}^m, |h_j^R \geqslant \mathbf{S}^R\right\} \quad \text{and} \quad \left\{\mathbf{z} \in \mathbb{C}^m, |h_j^R < \mathbf{S}^R\right\}, \tag{28}$$

$$\left\{\mathbf{z} \in \mathbb{C}^m, |h_j^I \geqslant \mathbf{S}^I\right\} \quad \text{and} \quad \left\{\mathbf{z} \in \mathbb{C}^m, |h_j^I < \mathbf{S}^I\right\}. \tag{29}$$

The normal vectors to the decision boundaries formed by the hidden neuron (given by Eqs. (26) and (27)) with respect to the input (**z**) are given by:

$$Q_h^R(\mathbf{z}^R, \mathbf{z}^I) = \left(\frac{\partial h_j^R}{\partial z_1^R} \cdots \frac{\partial h_j^R}{\partial z_k^R} \cdots \frac{\partial h_j^R}{\partial z_m^R}\, \frac{\partial h_j^R}{\partial z_1^I} \cdots \frac{\partial h_j^R}{\partial z_k^I} \cdots \frac{\partial h_j^R}{\partial z_m^I}\right), \tag{30}$$

$$Q_h^I(\mathbf{z}^R, \mathbf{z}^I) = \left(\frac{\partial h_j^I}{\partial z_1^R} \cdots \frac{\partial h_j^I}{\partial z_k^R} \cdots \frac{\partial h_j^I}{\partial z_m^R}\, \frac{\partial h_j^I}{\partial z_1^I} \cdots \frac{\partial h_j^I}{\partial z_k^I} \cdots \frac{\partial h_j^I}{\partial z_m^I}\right). \tag{31}$$

The decision boundaries are orthogonal to each other *iff* the dot product of these normal vectors is zero. The dot product of the normal vectors is given by:

$$Q_h^R(\mathbf{z}^R, \mathbf{z}^I).Q_h^I(\mathbf{z}^R, \mathbf{z}^I) = \sum_{k=1}^{m} \left(\frac{\partial h_j^R}{\partial z_k^R} \cdot \frac{\partial h_j^I}{\partial z_k^R}\right) + \left(\frac{\partial h_j^R}{\partial z_k^I} \cdot \frac{\partial h_j^I}{\partial z_k^I}\right), \tag{32}$$

$$\frac{\partial h_j^R}{\partial z_k^R} = \frac{\partial h_j^R}{\partial O_j^R}\frac{\partial O_j^R}{\partial z_k^R} + \frac{\partial h_j^R}{\partial O_j^I}\frac{\partial O_j^I}{\partial z_k^R} = \frac{\partial h_j^R}{\partial O_j^R} u_{jk}^R + \frac{\partial h_j^R}{\partial O_j^I} u_{jk}^I, \tag{33}$$

$$\frac{\partial h_j^I}{\partial z_k^R} = \frac{\partial h_j^I}{\partial O_j^R}\frac{\partial O_j^R}{\partial z_k^R} + \frac{\partial h_j^I}{\partial O_j^I}\frac{\partial O_j^I}{\partial z_k^R} = \frac{\partial h_j^I}{\partial O_j^R} u_{jk}^R + \frac{\partial h_j^I}{\partial O_j^I} u_{jk}^I, \tag{34}$$

$$\frac{\partial h_j^R}{\partial z_k^I} = \frac{\partial h_j^R}{\partial O_j^R}\frac{\partial O_j^R}{\partial z_k^I} + \frac{\partial h_j^R}{\partial O_j^I}\frac{\partial O_j^I}{\partial z_k^I} = \frac{\partial h_j^R}{\partial O_j^R}\left(-u_{jk}^I\right) + \frac{\partial h_j^R}{\partial O_j^I} u_{jk}^R, \tag{35}$$

$$\frac{\partial h_j^I}{\partial z_k^I} = \frac{\partial h_j^I}{\partial O_j^R}\frac{\partial O_j^R}{\partial z_k^I} + \frac{\partial h_j^I}{\partial O_j^I}\frac{\partial O_j^I}{\partial z_k^I} = \frac{\partial h_j^I}{\partial O_j^R}\left(-u_{jk}^I\right) + \frac{\partial h_j^I}{\partial O_j^I} u_{jk}^R. \tag{36}$$

Using the laws of differentiation, the derivative of the *j*th hidden layer response with respect to its net input

$$\frac{\partial h_j^R}{\partial O_j^R} = \frac{\sinh(O_j^R)\cos(3O_j^I) - \sinh(3O_j^R)\cos(O_j^I)}{\left(\cos(2O_j^I) + \cosh(2O_j^R)\right)^2}, \tag{37}$$

$$\frac{\partial h_j^I}{\partial O_j^I} = \frac{\sinh(O_j^R)\cos(3O_j^I) - \sinh(3O_j^R)\cos(O_j^I)}{\left(\cos(2O_j^I) + \cosh(2O_j^R)\right)^2}, \tag{38}$$

$$\frac{\partial h_j^R}{\partial O_j^I} = \frac{\cosh(O_j^R)\sin(3O_j^I) - \cosh(3O_j^R)\sin(O_j^I)}{\left(\cos(2O_j^I) + \cosh(2O_j^R)\right)^2}, \tag{39}$$

Please cite this article in press as: R. Savitha et al., Fast learning Circular Complex-valued Extreme Learning Machine (CC-ELM) for real-valued classification problems, Inform. Sci. (2011), doi:10.1016/j.ins.2011.11.003

$$\frac{\partial h_j^I}{\partial O_j^R} = \frac{\cosh\left(3O_j^R\right)\sin\left(O_j^I\right) - \cosh\left(O_j^R\right)\sin\left(3O_j^I\right)}{\left(\cos\left(2O_j^I\right) + \cosh\left(2O_j^R\right)\right)^2}. \tag{40}$$

From Eqs. (37)–(40), it can be observed that

$$\frac{\partial h_j^R}{\partial O_j^R} = \frac{\partial h_j^I}{\partial O_j^I}, \tag{41}$$

$$\text{and } \frac{\partial h_j^R}{\partial O_j^I} = -\frac{\partial h_j^I}{\partial O_j^R}. \tag{42}$$

Hence, it is evident that the *sech* activation function satisfies the Cauchy Riemann equations.

Substituting the Cauchy Riemann Equations in Eqs. (33)–(36), and obtaining the dot product of the normal vectors, we have,

$$Q_h^R(\mathbf{z}^R, \mathbf{z}^I).Q_h^I(\mathbf{z}^R, \mathbf{z}^I) = \sum_{k=1}^{m} \begin{bmatrix} -\frac{\partial h_j^R}{\partial O_j^R}\frac{\partial h_j^R}{\partial O_j^I}\left(u_{jk}^R\right)^2 - \left(\frac{\partial h_j^R}{\partial O_j^I}\right)^2 u_{jk}^R u_{jk}^I \\ +\left(\frac{\partial h_j^R}{\partial O_j^R}\right)^2 u_{jk}^R u_{jk}^I + \frac{\partial h_j^R}{\partial O_j^R}\frac{\partial h_j^R}{\partial O_j^I}\left(u_{jk}^I\right)^2 \\ -\frac{\partial h_j^R}{\partial O_j^R}\frac{\partial h_j^R}{\partial O_j^I}\left(u_{jk}^I\right)^2 + \left(\frac{\partial h_j^R}{\partial O_j^I}\right)^2 u_{jk}^R u_{jk}^I \\ -\left(\frac{\partial h_j^R}{\partial O_j^R}\right)^2 u_{jk}^R u_{jk}^I + \frac{\partial h_j^R}{\partial O_j^R}\frac{\partial h_j^R}{\partial O_j^I}\left(u_{jk}^R\right)^2 \end{bmatrix}, \tag{43}$$

$$= 0. \tag{44}$$

Thus, the decision boundaries formed by the real and imaginary parts of the hidden layer output are orthogonal. It can further be inferred that any fully complex-valued activation function that satisfies the Cauchy Riemann equations has two decision boundaries (formed by the real and imaginary parts of the hidden layer) and they are orthogonal to each other.

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

Lemma 3.1. *The decision boundaries formed by the real and imaginary parts of an output/hidden neuron in a fully complex-valued network with any fully complex-valued activation function that satisfies the Cauchy Riemann conditions are orthogonal to each other.*

4. Performance evaluation of CC-ELM classifier

In this section, we present the performance results of CC-ELM classifier in comparison with other existing complex-valued and real-valued classifiers. For performance evaluation, we consider both multi-category/binary classification problems from the UCI machine learning repository [5]. Based on a wide range of the Imbalance Factor (I.F.) (as defined in [37]) of the data sets, three multi-category and four binary data sets are chosen for the study. The imbalance factor is defined as

$$(\text{I.F.}) = 1 - \left(\frac{\min_{j=1\cdots C} N_j}{\sum_{j=1}^{C} N_j}\right) * C, \tag{45}$$

where N_j is the total number of samples belonging to the class j.

The description of these data sets including the number of classes, the number of input features, the number of samples in the training/testing and the imbalance factor are presented in Table 1. From the table, it can be observed that the problems chosen for the study have both balanced and unbalanced data set and the imbalance factors of the data sets vary widely.

Table 1
Description of benchmark data sets selected from [5] for performance study.

Type of data set	Problem	No. of features	No. of classes	No. of samples		I.F.
				Training	Testing	
Multi-category	Image Segmentation (IS)	19	7	210	2100	0
	Vehicle Classification (VC)	18	4	424	422	0.1
	Glass Identification (GI)	9	7	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

Finally, CC-ELM is used to solve two practical classification problems: the acoustic emission signal classification problem [20] and the mammogram classification for breast cancer detection [34].

The classification/confusion matrix Q is used to obtain the statistical measures for both the class-level and global performance of the various classifiers. Class-level performance is measured by the percentage classification (η_j) which is defined as:

$$\eta_j = \frac{q_{jj}}{N_j} \times 100\%, \tag{46}$$

where q_{jj} is the total number of correctly classified samples in the class c_j.

The global measures used in the evaluation are the average per-class classification accuracy (η_a) and the over-all classification accuracy (η_o) defined as:

$$\eta_a = \frac{1}{C}\sum_{j=1}^{C}\eta_j,$$
$$\eta_o = \frac{\sum_{j=1}^{C} q_{jj}}{\sum_{j=1}^{C} N_j} \times 100\%. \tag{47}$$

The performance of the classifiers are compared using these class-level and global performance measures.

4.1. Performance evaluation using benchmark classification problems

First, the performance of CC-ELM is evaluated using the three multi-category benchmark classification problems in Table 1, i.e., the Image Segmentation (IS), the Vehicle Classification (VC) and the Glass Identification (GI) data sets. The main aim here is to highlight the advantages of CC-ELM classifier with respect to the following two factors, viz.,

- The orthogonal decision boundaries that are characteristic of the complex-valued neural networks and
- The unique, non-linear circular transformation used to convert the real-valued features to the Complex domain.

To study the effect of the orthogonal decision boundaries, the performance of CC-ELM is compared against the best performing results of the existing real-valued classifiers for this problem, viz., Support Vector Machine (SVM) [8] and Self-adaptive Resource Allocation Network (SRAN) [41]. As CC-ELM classifier is developed in the framework of C-ELM, its performance is also compared against the best results of real-valued ELM [10] and its variant, the Real-Coded Genetic Algorithm for input weight selection (RCGA-ELM) [39,40]). The results for SVM and SRAN are reproduced from [41,**?**]. The results for RCGA-ELM is reproduced from [39]. Note that SVM is implemented in C language [41] and the other algorithms are implemented in MATLAB environment on a Pentium 4 processor with 4 GB RAM.

To study the effect of the circular transformation, the performance of CC-ELM classifier is compared against other existing complex-valued classifiers *viz.*, PE-CVNN [3,**?**] and MLMVN [1] classifiers. The results for the single layered PE-CVNN are reproduced from [4] and the results for MLMVN classifier are obtained using the software simulator available in the author's web site.[1]

4.1.1. Balanced data set: Image Segmentation (IS) problem

The image segmentation data set available in the UCI machine learning repository [5] is a multi-category well-balanced data set with 7 classes. The data set description for this problem is presented in Table 1. The data set for the IS problem has 30 samples in each class for training and 300 samples in each class for testing. The structure of CC-ELM and parameters of the network are selected using the procedure described in Section 2.2. The effect of the random selection of the transformation constants (a, b and α_l) and the hidden layer parameters ($\mathbf{u}_j$ and $\mathbf{v}_j$) of CC-ELM is studied over 50 independent runs, each with different random initializations. The mean (μ) and the standard deviation (σ) over the 50 independent runs are 92.18 and 0.32, respectively. Thus, it can be inferred that the random selection of parameters does not affect the classification performance of CC-ELM significantly.

The testing performance, the number of hidden neurons (K) and the training time for the IS problem are presented in Table 2. The best results obtained from 50 independent runs of CC-ELM is presented in the table. From the table, one can see that CC-ELM classifier performs better than other classifiers. It can also be observed that CC-ELM classifier requires lower computational effort, compared to all the other classifiers. The orthogonal decision boundary and the unique circular transformation of CC-ELM classifier helps it to outperform all the real-valued classifiers in the classification of this well-balanced data set. Also, even without any self-regulation and sequence altering, CC-ELM performs better than SRAN classifier, with reduced computational effort.

It can also be observed that CC-ELM classifier is better than the other complex-valued classifiers available in the literature. The performance of PE-CVNN classifier is almost equal to that of CC-ELM classifier. However, PE-CVNN classifier is

[1] http://www.eagle.tamut.edu/faculty/igor/Downloads.htm.

Table 2

Q2 Performance comparison on well-balanced multi-category image segmentation problem.

Type	Algorithm	K	Training time (s)	Testing efficiency (η_o)
Real-valued	ELM	100	0.03	90.67
	RCGA-ELM	50	–	91.00
	SRAN	47	22	92.29
	SVM	96[a]	721	90.62
Complex-valued	MLMVN	80	1384	83
	PE-CVNN	–	–	93.2[b]
	CC-ELM	**60**	**0.03**	**93.52**

[a] Number of support vectors.

[b] A single layer network was used in [4]. Also, in [4], a 10-fold validation has been done using 90% of the total samples in training and the remaining 10% for testing in each validation. In our work, we use only 10% of the samples in training.

trained using a 10-fold validation with 90% of the total samples used in training (in each validation) and 10% of the samples in testing [4], while CC-ELM classifier uses only 10% of the samples for training. Also, PE-CVNN classifier uses a computationally intensive gradient descent based learning algorithm as compared to CC-ELM that is computationally less intensive.

4.1.2. Unbalanced data sets: Vehicle Classification (VC) and Glass Identification (GI) problems

Next, we consider two unbalanced multi-category data sets with varying degrees of imbalance factors for the performance study, viz., the Vehicle Classification problem with a lower imbalance factor (0.1) and the Glass Identification problem with a higher imbalance factor (0.68).

The detailed description of these data sets including the number of input features, the number of samples in the training and testing data set and the number of classes are presented in Table 1.

As the input and hidden neuron parameters of CC-ELM (a, b, $\boldsymbol{\alpha}, V$ and U) are chosen randomly and the output weights W are computed analytically using the training samples, the effect of these random initializations is studied over 50 independent runs of CC-ELM. The mean (μ) and standard deviation (σ) of the average efficiencies for the VC problem are 79.48 and 0.75, respectively. Similarly, the mean (μ) and standard deviation (σ) of the average efficiencies for the GI problem are 88.31 and 1.85, respectively. Thus, it can be inferred that the random initialization of the input/hidden layer parameters does not affect the classification performance significantly, even in unbalanced data sets.

The classification performance of CC-ELM algorithm in comparison with the other real and complex-valued classifiers for these problems is presented in Table 3. From the table, it can be observed that the average and over-all classification accuracies of CC-ELM classifier is higher than those of the real-valued classifiers for the VC and GI problems.

The advantage in the classification performance of CC-ELM classifier in the highly unbalanced glass identification data set can be clearly seen from the Table 3. Also, with no self-regulation, CC-ELM classifier outperforms the best results in real-valued (SRAN algorithm) classifier by nearly 6% and 4% in the vehicle classification and the glass identification problems, respectively. Moreover, CC-ELM performs significantly better than the other existing complex-valued classifiers. It can be clearly seen that PE-CVNN, whose performance is comparable to that of CC-ELM in the well-balanced IS data set does not perform well in classification of the unbalanced VC and GI data sets. Another interesting observation that can be made from the table is that the improvement in classification accuracy of CC-ELM is higher in the highly unbalanced GI data set than that in the VC data set.

Comparing the performance of CC-ELM classifier against other complex-valued classifiers on multi-category classification problems, it can be seen that the performance of CC-ELM classifier is better than the other complex-valued classifiers, especially in unbalanced data sets. Though both PE-CVNN and MLMVN also use orthogonal decision boundaries, their performances may be affected by the transformation, learning algorithm and activation function.

4.2. Binary classification data sets

Recently, in [12], an optimization based real-valued Extreme Learning Machine (O-ELM) is presented for solving classification problems and the performance of O-ELM has been reported on a set of binary classification data sets. As CC-ELM is also developed in the framework of extreme learning machines, we also compare the performance of CC-ELM here in comparison with support vector machines, SRAN and O-ELM [12] on the 4 binary benchmark data sets from the UCI machine learning repository. The details of the problems considered for the study are presented in Table 1. The performance comparison of the results are tabulated in Table 4.

From the table, it is clearly evident that even without any optimization tools CC-ELM outperforms O-ELM classifier [12]. When compared to O-ELM, the network used in the classification is also compact.

Also, the performance of CC-ELM has been studied on the noisy synthetic two class classification problem [24]. With five hidden neurons, CC-ELM achieves 92.3% classification accuracy on 1000 testing samples, which is 1.5% more than the result reported in the literature for real-valued multi-layer perceptron network with 6 hidden neurons.

Please cite this article in press as: R. Savitha et al., Fast learning Circular Complex-valued Extreme Learning Machine (CC-ELM) for real-valued classification problems, Inform. Sci. (2011), doi:10.1016/j.ins.2011.11.003

Table 3
Performance comparison on unbalanced multi-category classification problems.

Problem	Type	Algorithm	K	Training time (s)	Testing efficiency	
					(η_o)	(η_a)
Vehicle classification problem		ELM	300	0.81	78.17	78
	Real-valued	RCGA-ELM	75	-	74.2	74.4
		SRAN	55	113	75.12	76.86
		SVM	234[a]	550	68.72	67.99
		PE-CVNN	–	–	78.7[b]	–
	Complex-valued	MLMVN	90	1396	78	77.25
		CC-ELM	**85**	**0.1084**	**82.23**	**82.52**
Glass identification problem		ELM	60	0.21	73	65.46
	Real-valued	RCGA-ELM	60	–	78.1	–
		SRAN	59	28	86.21	80.95
		SVM	102	320	64.23	60.01
		PE-CVNN	–	–	65.5b	–
	Complex-valued	MLMVN	85	1421	73.24	66.83
		CC-ELM	**100**	**0.08**	**94.44**	**84.52**

[a] Number of support vectors.
[b] A single layer network was used in [4]. Also, in [4], a 10-fold validation has been done using 90% of the total samples in training and the remaining 10% for testing in each validation. In our work, we use only 50% of the samples in training.

Table 4
Performance comparison on benchmark binary classification problems.

Problem	Type	Algorithm	K	Training time (s)	Testing efficiency (η_o)
Liver disorders		SVM	158	0.0972	68.24
	Real-valued	O-ELM	131	0.1734	72.34
		SRAN	91	3.38	66.9
	Complex-valued	**CC-ELM**	**10**	**0.059**	**75.5**
PIMA data		SVM	209	0.205	76.43
	Real-valued	O-ELM	217	0.2867	77.27
		SRAN	97	12.24	78.53
	Complex-valued	**CC-ELM**	**20**	**0.073**	**81.25**
Breast cancer		SVM	190	0.1118	94.20
	Real-valued	O-ELM	66	0.1423	96.32
		SRAN	7	0.17	96.87
	Complex-valued	**CC-ELM**	**15**	**0.0811**	**97.39**
Iono-sphere		SVM	30	0.0218	90.18
	Real-valued	O-ELM	32	0.0359	89.48
		SRAN	21	3.7	90.84
	Complex-valued	**CC-ELM**	**15**	**0.0312**	**92.43**

Thus, from the study on both the multi-category and binary benchmark data sets considered, it may be noted that CC-ELM classifier outperforms the existing complex-valued and real-valued classifiers. It can also be inferred that the improvement in the classification performance of CC-ELM classifier is better in the highly unbalanced data sets. The better classification performance of CC-ELM classifier can be associated to the orthogonal decision boundaries of the complex-valued classifiers and the unique circular transformation used as the activation function in the input layer.

Next, we study the performance of CC-ELM classifier on two practical classification problems, viz., an acoustic emission signal classification problem [20] and the mammogram classification problem for breast cancer detection [34].

4.3. Acoustic emission signal classification problem

Acoustic emission signals are electrical version of the stress or pressure waves, produced by sensitive transducers. These waves are produced due to the transient energy release caused by irreversible deformation processes in the material [20]. There are various sources of acoustic emission, and these sources can be characterized by the acoustic signals. The classification of acoustic emission signals based on their sources is a very difficult problem, especially in the real world, where ambient noise and pseudo acoustic emission signals exist. Even in a noise free environment, superficial similarities exist between acoustic emission signals produced by different sources, making the classification task cumbersome.

In the study conducted in [20], noise free burst type acoustic emission signal from a metal surface is assumed. The data set presented in [20] uses five input features to classify the acoustic signals to one of the four sources, i.e., the pencil source, the pulse source, the spark source and the noise. A training data set with 62 samples and testing data set with 137 samples

Table 5
Performance comparison results for the acoustic emission problem.

Type	Classifier	Testing	
		η_o	η_{av}
Real-valued	Genetic programming [36]	90.58	–
	Ant colony optimization [21]	90.51	89.27
	Fuzzy C-means clustering [20]	–	93.34
Complex-valued	**CC-ELM**	**99.27**	**99.17**

Table 6
Performance comparison results for the mammogram problem.

Classifier domain	Classifier	Train time	K	Testing	
				η_o	η_a
Real-valued	ELM	0.032	30	91	91
	SVM [33]	–	–	–	95.44
Complex-valued	**CC-ELM**	**0.047**	**40**	**100**	**100**

are used for the acoustic emission signal classification problem. For details of the input features and the experimental set up used in the data collection, one should refer to [20].

The performance study results of CC-ELM classifier in comparison to best results available in the literature for the acoustic emission signal classification problem, is presented in Table 5. The results of CC-ELM classifier are compared against a Fuzzy K-means clustering algorithm [20], the ant colony optimization algorithm [21] and genetic programming [36]. It is observed that CC-ELM algorithm requires only 10 neurons to achieve an over-all testing efficiency of 99.27%, which is about 6% better than the best results reported in the literature for this problem. Thus, CC-ELM performs an efficient classification of the acoustic emission signals using a compact network.

4.4. Mammogram classification for breast cancer detection problem

Mammogram is a better means for early diagnosis of breast cancer, as tumors and abnormalities show up in mammogram much before they can be detected through physical examinations [9]. Clinically, identification of malignant tissues involves identifying the abnormal masses or tumors, if any, and then classifying the mass as either malignant or benign. However, once a tumor is detected, the only method of determining whether it is benign or malignant is by conducting a biopsy, which is an invasive procedure that involves the removal of the cells or tissue from a patient. A 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.

In this study, the mammogram database available in [34] has been used. The nine input features, extracted from the mammogram of the identified abnormal mass, are used to classify the tumor as either malignant or benign. Here, 97 samples are used to develop CC-ELM classifier and the performance of CC-ELM classifier is evaluated using remaining 11 samples. For further details on the input features and the data set, one should refer to [34].

Performance results of CC-ELM classifier, in comparison with ELM and SVM [33] are presented in Table 6. From the table, it is seen that CC-ELM classifier performs efficiently with 100% classification accuracy. This is higher than the best results available in the literature for this problem by approximately 4%. Comparing the classification performance of CC-ELM classifier to that of the real-valued ELM classifier, the classification performance is improved by nearly 9%.

Thus, from the performance study conducted with different real-valued and complex-valued classifiers for the chosen benchmark data sets and practical classification problems, it can be observed that CC-ELM classifier performs better than other existing classifiers.

5. Conclusions

In this paper, we have presented the circular complex-valued extreme learning machine classifier for performing real-valued classification tasks in the complex domain. CC-ELM classifier uses a single hidden layer network with a non-linear input/hidden layer and a linear output layer. At the input layer, a unique nonlinear circular transformation is used as the activation function to make a one-to-one mapping of the real-valued input features to the Complex domain. At the hidden layer, the complex-valued input features are mapped onto a higher dimensional complex plane, using the '*sech*' activation function. In CC-ELM, the input parameters and the parameters of the hidden layer are chosen randomly and the output weights are calculated analytically, requiring lesser computational effort to perform classification tasks. Existence of orthogonal decision boundaries in the hidden and output layers of CC-ELM has been proven analytically. The advantages of these orthogonal

Please cite this article in press as: R. Savitha et al., Fast learning Circular Complex-valued Extreme Learning Machine (CC-ELM) for real-valued classification problems, Inform. Sci. (2011), doi:10.1016/j.ins.2011.11.003

decision boundaries and the activation function used in the input layer are studied by comparing the performance of CC-ELM classifier with other real-valued and complex-valued classifiers using benchmark classification problems from the UCI machine learning repository and two practical classification problems. Performance of CC-ELM on these data sets clearly show that it performs better than the existing (well-known) real-valued and other complex-valued classifiers, especially for unbalanced data sets.

Acknowledgement

The authors thank the reviewers for their critical comments which helped to improve the quality of the paper significantly. The first and second authors also wish to thank the Ministry of Education (MoE), Singapore for the financial support through TIER-I funding to conduct this study.

References

[1] I. Aizenberg, C. Moraga, Multilayer feedforward neural network based on multi-valued neurons (MLMVN) and a backpropagation learning algorithm, Soft Computing 11 (2) (2007) 169–183.
[2] I. Aizenberg, D.V. Paliy, J.M. Zurada, J.T. Astola, Blur identification by multilayer neural network based on multivalued neurons, IEEE Transactions on Neural Networks 19 (5) (2008) 883–898.
[3] M.F. Amin, K. Murase, Single-layered complex-valued neural network for real-valued classification problems, Neurocomputing 72 (4–6) (2009) 945–955.
[4] M.F. Amin, M.M. Islam, K. Murase, Ensemble of single-layered complex-valued neural networks for classification tasks, Neurocomputing 72 (10–12) (2009) 2227–2234.
[5] C. Blake, C. Merz, UCI Repository of Machine Learning Databases, Department of Information and Computer Sciences, University of California, Irvine, 1998. URL: <http://www.archive.ics.uci.edu/ml/>.
[6] J.C. Bregains, F. Ares, Analysis, synthesis, and diagnostics of antenna arrays through complex-valued neural networks, Microwave and Optical Technology Letters 48 (8) (2006) 1512–1515.
[7] K. Burse, R.N. Yadav, S.C. Shrivastava, Channel equalization using neural networks: a review, IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews 40 (3) (2010) 352–357.
[8] N. Cristianini, J.S. Taylor, An Introduction to Support Vector Machines, Cambridge University Press, Cambridge, UK, 2000.
[9] P. Gamigami, Atlas of Mammography: New Early Signs in Breast Cancer, Blackwell Science, Cambridge, MA, USA, 1996.
[10] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.
[11] G.B. Huang, M. Li, L. Chen, C.K. Siew, Incremental extreme learning machine with fully complex hidden nodes, Neurocomputing 71 (4–6) (2008) 576–583.
[12] G.B. Huang, X. Ding, H. Zhou, Optimization method based extreme learning machine for classification, Neurocomputing 74 (1–3) (2010) 155–163.
[13] G.-B. Huang, D.H. Wang, Y. Lan, Extreme learning machines: a survey, International Journal of Machine Learning and Cybernetics 2 (2) (2011) 107–122.
[14] T. Kim, T. Adali, Fully-complex multilayer perceptron network for nonlinear signal processing, Journal of VLSI Signal Processing 32 (1–2) (2002) 29–43.
[15] M.B. Li, G.B. Huang, P. Saratchandran, N. Sundararajan, Fully complex extreme learning machine, Neurocomputing 68 (1–4) (2005) 306–314.
[16] T. Nitta, The computational power of complex-valued neuron, Artificial Neural Networks and Neural Information Processing ICANN/ICONIP. Lecture Notes in Computer Science 2714 (2003) 993–1000.
[17] T. Nitta, Solving the XOR problem and the detection of symmetry using a single complex-valued neuron, Neural Networks 16 (8) (2003) 1101–1105.
[18] T. Nitta, On the inherent property of the decision boundary in complex-valued neural networks, Neurocomputing 50 (2003) 291–303.
[19] T. Nitta, Orthogonality of decision boundaries of complex-valued neural networks, Neural Computation 16 (1) (2004) 73–97.
[20] S.N. Omkar, S. Suresh, T.R. Raghavendra, V. Mani, Acoustic emission signal classification using fuzzy C-means clustering, in: Proceedings of the ICONIP'02, 9th International Conference on Neural Information Processing, vol. 4, 2002, pp. 1827–1831.
[21] S.N. Omkar, U. Raghavendra Karanth, Rule extraction for classification of acoustic emission signals using ant colony optimisation, Engineering Applications of Artificial Intelligence 21 (8) (2008) 1381–1388.
[22] J.M. Ortega, Matrix Theory, Plenum Press, New York, USA, 1986.
[23] L. Peng, B. Yang, Y. Chen, A. Abraham, Data gravitation based classification, Information Sciences 179 (6) (2009) 809–819.
[24] B.D. Ripley, Neural networks and related methods for classification, Journal of the Royal Statistical Society. Series B (Methodological) 56 (3) (1994) 409–456.
[25] R. Savitha, S. Suresh, N. Sundararajan, P. Saratchandran, A new learning algorithm with logarithmic performance index for complex-valued neural networks, Neurocomputing 72 (16–18) (2009) 3771–3781.
[26] R. Savitha, S. Suresh, N. Sundararajan, A fully complex-valued radial basis function network and its learning algorithm, International Journal of Neural Systems 19 (4) (2009) 253–267.
[27] R. Savitha, S. Suresh, N. Sundararajan, A self-regulated learning in fully complex-valued radial basis function networks, in: Proceedings of International Joint Conference on Neural Networks (IJCNN 2010), ~~2010~~ (doi: doi:10.1109/IJCNN.2010.5596781).
[28] R. Savitha, S. Vigneshwaran, S. Suresh, N. Sundararajan, Adaptive beamforming using complex-valued radial basis function neural networks, in: Proceedings of TENCON'09, IEEE Region 10 Conference, Singapore, Nov. 23–26, 2009, pp. 1–6.
[29] R. Savitha, S. Suresh, N. Sundararajan, H.J. Kim, Fast learning fully complex-valued classifiers for real-valued classification problems, Lecture Notes in Computer Science 6675 (LNCS Part I) (2011) 602–609.
[30] R. Savitha, S. Suresh, N. Sundararajan, H.J. Kim, A fully complex-valued radial basis function classifier for real-valued classification problems, Neurocomputing (2011), ~~doi:10.1016/j.neucom.2011.05.036;~~
[31] C. Shen, H. Lajos, S. Tan, Symmetric complex-valued RBF receiver for multiple-antenna-aided wireless systems, IEEE Transactions on Neural Networks 19 (9) (2008) 1659–1665.
[32] N. Sinha, M. Saranathan, K.R. Ramakrishna, S. Suresh, Parallel magnetic resonance imaging using neural networks, in: Proceedings of ICIP'07, IEEE International Conference on Image Processing, vol. 3, 2007, pp. 149–152.
[33] T.S. Subashini, V. Ramalingam, S. Palanivel, Automated assessment of breast tissue density in digital mammograms, Computer Vision and Image Understanding 114 (1) (2010) 33–43.
[34] J. Suckling et al., The mammographic image analysis society digital mammogram database, Exerpta Medica International Congress Series 1069 (1994) 375–378.
[35] S. Suresh, S.N. Omkar, V. Mani, T.N. Guru Prakash, Lift coefficient prediction at high angle of attack using recurrent neural network, Aerospace Science and Technology 7 (8) (2003) 595–602.
[36] S. Suresh, S.N. Omkar, V. Mani, C. Menaka, Classification of acoustic emission signal using genetic programming, Journal of Aerospace Science and Technology 56 (1) (2004) 26–41.

Please cite this article in press as: R. Savitha et al., Fast learning Circular Complex-valued Extreme Learning Machine (CC-ELM) for real-valued classification problems, Inform. Sci. (2011), doi:10.1016/j.ins.2011.11.003

[37] S. Suresh, N. Sundararajan, P. Saratchandran, Risk-sensitive loss functions for sparse multi-category classification problems, Information Sciences 178 (12) (2008) 2621–2638.

[38] S. Suresh, N. Sundararajan, P. Saratchandran, A sequential multi-category classifier using radial basis function networks, Neurocomputing 71 (7–9) (2008) 1345–1358.

[39] S. Suresh, R. Venkatesh Babu, H.J. Kim, No-reference image quality assessment using modified extreme learning machine classifier, Applied Soft Computing 9 (2) (2009) 541–552.

[40] S. Suresh, S. Saraswathi, N. Sundararajan, Performance enhancement of extreme learning machine for multi-category sparse data classification problems, Engineering Applications of Artificial Intelligence 23 (7) (2010) 1149–1157.

[41] S. Suresh, Keming Dong, H.J. Kim, A sequential learning algorithm for self-adaptive resource allocation network classifier, Neurocomputing 73 (16–18) (2010) 3012–3019.

[42] S. Suresh, R. Savitha, N. Sundararajan, A sequential learning algorithm for complex-valued self-regulating resource allocation network-CSRAN, IEEE Transactions on Neural Networks 22 (7) (2011) 1061–1072.

[43] A. Veloso, W. Meira, M. Goncalves, H.M. Almeida, M. Zaki, Calibrated lazy associative classification, Information Sciences 181 (13) (2011) 2656–2670.

[44] X.Z. Wang, J.H. Zhai, S.X. Lu, Induction of multiple fuzzy decision trees based on rough set technique, Information Sciences 178 (16) (2008) 3188–3202.

[45] X.Z. Wang, C.R. Dong, Improving generalization of fuzzy if-then rules by maximizing fuzzy entropy, IEEE Transactions on Fuzzy Systems 17 (3) (2009) 556–567.