A Self-Regulated Learning in Fully Complex-valued Radial Basis Function Networks

R. Savitha, *Student Member, IEEE*, S. Suresh, *Member, IEEE* and N. Sundararajan, *Fellow, IEEE*

Abstract— **In this paper, we present an efficient learning algorithm for a Fully Complex-valued Radial Basis Function (FC-RBF) Network using a self-regulatory system. One of the important issues in gradient descent learning algorithm for complex-valued network is the proper selection of training data sequence. In general, it is assumed that the training data is uniformly distributed in the input space with non-recurrent training samples. For most real-world problems, this assumption may not be valid. Hence, one needs to develop a learning algorithm which can select proper samples for learning. This paper presents a self-regulatory system that selects samples for learning in each epoch of the batch learning scheme. The algorithm focuses on learning samples with higher errors in the same epoch, deleting samples with smaller errors from the training data set. If the samples do not satisfy both these conditions, they are neither learnt nor deleted but will be used in the next epoch for learning. As this system avoids repeated learning of similar samples, it improves the generalization performance of the FC-RBF network with a lesser computational effort. Performance studies on benchmark problems clearly show the superiority of the proposed algorithm.**

I. Introduction

Complex-valued neural networks are being increasingly employed in adaptive signal processing, communication, medical imaging, which involves complex-valued signals, and requires adaptive algorithms. A few applications using complex-valued neural networks, available in literature, are adaptive antenna arrays [1], Doppler Signals classification [2] etc. Conventionally, split-complex networks [3], which are actually real-valued neural networks, were used in these applications. Such a network considered the complex-valued signal as two real-valued entities - the signals were split into magnitude-phase components or real-imaginary components. For example, Uncini et al.[4], presented a neural network based on adaptive activation functions, for split complex-valued multi-layer perceptron. But, from the sensitivity study on the split complex-valued MLP presented by Yang et al. [5] , it is observed that the process of splitting the complex-valued signals into real and imaginary components introduces phase distortions in complex-valued approximations. Also, it is shown by Kim and Adali [6] that such a network fails to approximate phase accurately. This is due to the use of real-valued gradient, which do not represent the true complex-valued gradient. As accurate phase estimation is very important in many communication application problems [7], a fully complex-valued network becomes very essential. The critical issue in a fully complex-valued neural network is the selection of an appropriate activation function, as Liouville's theorem suggests that an entire and bounded function is a constant function in the Complex domain. However, a constant function is not acceptable as an activation function as they cannot project the input space to a non-linear higher dimensional space. As neither analyticity and boundedness can be compromised, nor is a constant function acceptable, choices of activation functions for the complex-valued neural network are limited. Kim and Adali relaxed the desired properties for a complex-valued activation function presented by Georgiou and Koutsougeras [8] and proposed Elementary Transcendental Functions (ETF), which are almost everywhere *(a.e)* bounded functions as activation functions for a Fully complex-valued Multi-Layer Perceptron (FC-MLP) [6]. However, these activation functions have finite singular points, as also their derivatives. Hence, the network has to be guarded against the singularities of the activation functions and their derivatives. A brief overview of the complex-valued networks, available in literature, has been presented by Savitha et al. [9]. They also showed that the performance of the FC-MLP network depends on the initialization and learning rate parameter.

On the other hand, RBF networks use Gaussian function as the basis function and several real-valued RBF networks and their learning algorithms have been extended to the Complex domain. Complex-valued RBF network (CRBF), first introduced by Chen et al. [10], is a batch learning algorithm, with complex-valued weights and centers, but employing Gaussian activation function as the basis function. Similarly, the Complex-valued Minimal Resource Allocation Network (CMRAN) presented by Jianping et al. [11] and the Complex-valued Extreme Learning Machines (CELM) presented by Li et al. [12] are direct extensions of the real-valued MRAN and the ELM algorithms, respectively. However, all these algorithms use the Gaussian function as the basis of the activation function. Hence, in these algorithms, despite the weights and centers being complex-valued, the hidden layer response remains real-valued. Recently, Savitha et al. [13], have proposed a Fully complex-valued RBF (FC-RBF) network with a fully complex-valued *sech* activation function, and also presented its complex-valued gradient descent based learning algorithm. The approximation ability of the algorithm was shown through a number of simulation studies. It should be noted that all the complex-valued learning algorithms in the literature are based on the assumption that the training data set is well-distributed in the input-space with non-recurring data samples. Satisfying

S. Suresh is with the School of Computer Engineering, Nanyang Technological University, Singapore. email: ssundaram@ntu.edu.sg

R. Savitha and N. Sundararajan are with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore.

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

such an assumption in real world problems is difficult. In addition, presenting similar samples again and again in the training sequence leads to poor generalization performance. In real-valued problems, many research articles have been published on global selective attention for efficient learning of gradient/stochastic algorithm [15], [16]). Chariatis [16] carried out the adaptive selection of training samples using an exponential trace of the average error. Here, samples enter and leave the training set automatically, with a tendency to train on samples with high error. In this paper, we present a similar framework for the complex-valued batch learning FC-RBF network. The fully complex-valued radial basis function network proposed in [13] is used as the basic framework to study the self-regulatory system. The performance of the proposed Self-Regulatory system for the FC-RBF network (SR-FCRBF) is evaluated using benchmark function approximation problems defined in [9] and the highly non-linear two-spiral classification problem [18].

The paper is organized as follows: Section II gives a brief summary of the Fully Complex-valued Radial Basis Function Network presented in [13]. This network is the basic framework on which the proposed algorithm is studied. The self- regulatory system for the FC-RBF network is also proposed in section II. In section III, the performance of the proposed algorithm is studied on two benchmark complex-valued function approximation problems and a two spiral classification problem. Section IV gives the conclusions from the study.

II. SELF-REGULATORY SYSTEM FOR FC-RBF NETWORK

A. A Brief Review on FC-RBF

In this paper, we use the FC-RBF network as the basic building block. The architecture of the FC-RBF network, presented in Fig. 1 has one input layer, one hidden layer and one output layer. The output layer has linear activation function and hidden layer uses '$sech(.)$' function, proposed in [13] as the basis of the activation function.

Let K be the number of complex-valued hidden neurons of the network. Here, each neuron uses '$sech$' function as the basis of the activation function. The predicted complex-valued output of the network ($\widehat{\mathbf{y}}_t \in \mathbb{C}^n$) for a complex-valued input $\mathbf{x}_t \in \mathbb{C}^m$ is given by eq. (1)

$$\widehat{y}_t^i = \sum_{j=1}^{K} w_{ij}\ sech[\mathbf{v}_j^T(\mathbf{x}_t - \mathbf{c}_j)], i = 1,2,....n \quad (1)$$

where $\mathbf{v}_j \in \mathbb{C}^m$ is the complex-valued scaling factor, $\mathbf{c}_j \in \mathbb{C}^m$ is the center of the j^{th} neuron and w_{ij} be the complex-valued output weight connecting the j^{th} hidden neuron and the i^{th} output neuron. The detailed gradient descent based learning algorithm for this network has been presented in [13]. The following updates were derived for $\mathbf{c}_j$, $\mathbf{v}_j$ and w_{ij}.

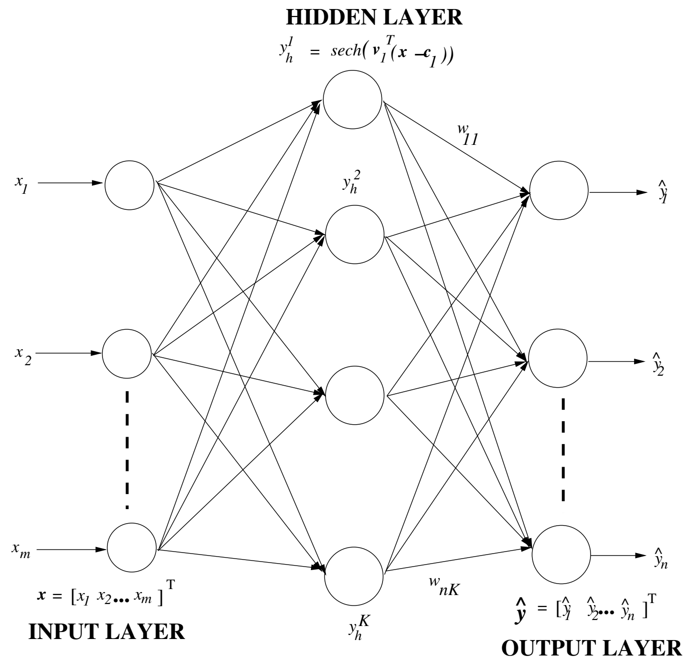


Fig. 1: The Fully Complex-valued RBF Network

$$\Delta w_{ij} = \eta_w \overline{\mathbf{y}}_h^j\ (y_t^i - \widehat{y}_t^i); \quad i = 1,2,..n; \quad j = 1,2,...K \quad (2)$$

$$\Delta \mathbf{v}_j = \eta_v (y_t^i - \widehat{y}_t^i)\ \overline{w_{ij}}\ \overline{f}'(\mathbf{x}_t - \mathbf{c}_j)\overline{(\mathbf{x}_t - \mathbf{c}_j)} \quad (3)$$

$$\Delta \mathbf{c}_j = \eta_c (y_t^i - \widehat{y}_t^i)\ \overline{w_{ij}}\ \overline{f}'(\mathbf{v}_j^T(\mathbf{x}_t - \mathbf{c}_j))\overline{\mathbf{v}}_i \quad (4)$$

where, η_c, η_v and η_w are the learning rates for center, scaling and weights parameters respectively.

B. Self-Regulatory System

In this section, a self-regulatory system, that chooses samples to participate in learning in each epoch, is presented.

C. Brief description of the algorithm:

The system chooses samples to be learnt in each epoch, alongwith deleting samples with the least error in each epoch. The control parameters that aid in selection of proper training samples for learning are self-regulatory in that, their values decline as learning progresses. The more the number of samples that participate in learning in each epoch, the lesser are the errors. Hence, the behavior of the control parameters are such as to accommodate this error behavior. Besides, samples with information similar to the knowledge already acquired by the network are deleted during the training. However, the deleting control parameters are fixed, and are not self-regulatory. As function approximation using complex-valued neural network includes accurate approximation of both magnitude and phase, the instantaneous magnitude and phase error defined below are used for regulating the learning process.

$$M_t^e = \max_{i=1,2,\cdots,n} \sqrt{\overline{(y_t^i - \widehat{y}_t^i)} * (y_t^i - \widehat{y}_t^i)}; \quad (5)$$

$$\phi_t^e = \max_{i=1,2,\cdots,n} |(\angle y_t^i - \angle \widehat{y}_t^i)|/(\pi). \quad (6)$$

where $\widehat{y}_t^i$ is as defined in eq. (1).

As each sample is presented to the network in each epoch, the samples are either

- Used in learning to update the network parameters. OR
- Deleted from the training dataset OR
- Skipped in the present epoch. These samples continue to remain in the training dataset for presentation to the network in the succeeding epochs.

Next, we present each of these conditions in detail:

- **Network Learning:** If either the magnitude error or the phase error of a sample presented to the FC-RBF network are greater than their respective control parameters, the sample participates in learning, i.e., the free parameters of the network are updated if the following condition is satisfied:

$$\textbf{IF} \quad M_t^e \geq E_L^M \quad OR \quad \phi_t^e \geq E_L^\phi \quad (7)$$

where E_L^M and E_L^ϕ are the self-regulated learning control parameters.

Learning control parameters: The self-regulating learning control parameters for the magnitude and phase reduces from maximum to minimum based on the instantaneous sample error in each epoch. The initial value of the learning control parameters are set at a higher value to facilitate participation of more samples in the initial epochs. The learning control parameters gradually reduces with learning. Thus, as learning progresses, more knowledge is acquired by the network, and hence, fewer samples participate in the learning process. To ensure this, the self- regulation of the magnitude and phase learning control parameters occur based on:

$$IF \ M_t^e \geq E_L^M, \ E_L^M = \delta E_L^M - (1-\delta) M_t^e \quad (8)$$

$$IF \ \phi_t^e \geq E_L^\phi, \ E_L^\phi = \delta E_L^\phi - (1-\delta) \phi_t^e \quad (9)$$

where δ is the slope at which the control parameters decrease with respect to the instantaneous error.

Thus the algorithm evaluates the error of all samples but only trains samples whose magnitude or phase error is greater than the magnitude or phase learning control parameters, respectively. Initially, training is performed on all samples, but gradually it concentrates on samples with high error or boundary samples. During learning, the network parameters are updated based on the gradient descent based learning algorithm for the FC-RBF network presented in [13] and summarized in eqs. (3), (4) and (4).

- **Deletion of Samples:** Samples whose root mean square magnitude error (eq. (5) and the average phase error (eq. (6)) are lesser than their respective deleting control parameters are deleted from the training dataset in each epoch, i. e.,

$$\textbf{IF} \quad M_t^e < E_D^M \ \& \ \phi_t^e < E_D^\phi \quad (10)$$

where E_D^M and E_D^ϕ are the magnitude and phase deleting control parameters, respectively. These deleting control parameters are fixed and are not self-regulated. The deletion of similar samples ensures that the network is not over-trained. Since the algorithm uses reduced training set in every epoch, the computational time to approximate the desired function is also less.

- **Sample Skipping:** A few samples that satisfy neither the learning condition, nor the deletion condition are skipped from learning/deletion in each epoch. However, they continue to remain in the training data set. When the magnitude and phase learning control parameters are adapted, the samples skipped in earlier epochs might enter into learning process or might be deleted, based on their errors, in the subsequent epochs.

The algorithm is briefly described in Listing I.

In the next section, the performance of the proposed algorithm is evaluated using two benchmark complex-valued function approximation problems and a two-spiral classification problem.

III. Performance Analysis

In this section, the performance of the proposed self-regulatory framework for the FC-RBF network is evaluated on two benchmark complex-valued function approximation problems and the two spiral classification problem. In the study, the mean square error for magnitude and the average phase are defined in eq. (11).

$$J_{Me} = \sqrt{\frac{1}{N_t}\sum_{j=1}^{N_t}\left[\frac{1}{n}\sum_{k=1}^{n}(e_j^k.\bar{e}_j^k)\right]}$$

$$\Phi_e = \frac{1}{N_t}\sum_{t=1}^{N_t}\left[\frac{1}{n}\left[\sum_{k=1}^{n} |\,[arg(y_t^k) - arg(\widehat{y}_t^k)]\,|\right]\right] \quad (11)$$

where e_i is the error of the i-th output neuron and N_t is the total number of training samples.

A. Function Approximation: Synthetic Examples

Two synthetic Complex-valued Function Approximation Problems (CFAP-I and CFAP-II) defined in [9] are first considered for the study. The control parameters for the self-regulation system, used in the study are as shown below:

- E_L^M = [0.6;0.0001]; $E_L^\phi = [0.4; 0.0001]$.
- E_D^M = 0.0001; E_D^ϕ = 0.005.
- α = 0.999995

1) CFAP-I:: The performance of the proposed self-regulated system was studied on the synthetic complex-valued function approximation problem II defined by Eq. ((12)) [9].

$$f_2(\mathbf{z}) = Z_3 + 10 Z_1 Z_4 + \frac{Z_2^2}{Z_1} \quad (12)$$

Listing 1 Self-Regulatory System for Fully-Complex Radial Function Network.

```
Input: The data set{(x_1,y_1),(x_2,y_2),..,(x_N,y_N)}
       of the function to be approximated.
Output:Parameters of the network.
START
  Initialization:
     a. Choose the number of hidden neurons.
     b. Initialize c_j, v_j and w_ij.
     c. Initialize the number of epochs OR
        specify the error based stopping
                                criterion.
  While STOPPING CRITERION(Epoch)
    FOR S=1,2,..,N(Sample)
       Compute the network output (eq. (1)).
       Compute the instantaneous Magnitude
          and Phase errors (eqs. (5), (6)).
          IF M_t^e < E_D^M & φ_t^e < E_D^φ THEN
            Delete the sample from the
            training dataset.
          ELSEIF M_t^e ≥ E_L^M OR φ_t^e ≥ E_L^φ THEN
            Use the sample in learning.
            Update the network parameters
            using eqs. (3), (4) and (4).
            The control parameters E_L^M and
            E_L^φ are self-regulated
            according to eq. (8, 9).
          ELSE
            The current sample continues to
            remain in the training dataset,
            unaffected. They can be used in
            learning/ deleted in subsequent
            epochs.
          ENDIF
    END FOR(Sample)
  END(Epoch)
STOP
```

where $Z_i \in \mathbb{C}$ and are initialized within ball-of-radius 2.5. The function $f_2(\mathbf{z})$ is normalized to ball-of-radius 1. Training set of 3000 randomly chosen samples and a testing set of 1000 randomly chosen samples were used for the study. The number of hidden neurons are chosen using a heuristic procedure similar to that presented in [17] for real-valued networks.

In Fig. 2, the instantaneous magnitude and phase error, the magnitude and phase learning control parameters, the magnitude and phase deleting control parameters are shown over a window of 50 samples (Sample instants 1050 -1100) from the learning during epoch 50. Fig. 2(a) gives a snapshot of instantaneous magnitude error, the magnitude learning and deleting control parameters. The effect of the self-regulation on the samples can be clearly seen from the figure. The following observations can be made:

- A few samples whose instantaneous magnitude error is greater than the learning control parameter for magnitude (E_L^M) participated in learning. Eg. sample instant 1067.
- A few samples whose instantaneous magnitude error is lesser than the deleting control parameter for magnitude (E_D^M) are deleted from the training dataset. Eg. Sample instant 1069.
- There are a few samples whose instantaneous magnitude error is greater than E_D^M and lesser than E_L^M (Eg. Sample instant 1081). These samples neither took part in learning, nor were deleted in the current epoch. Instead, they were skipped in the current epoch, to be presented to the network in future.

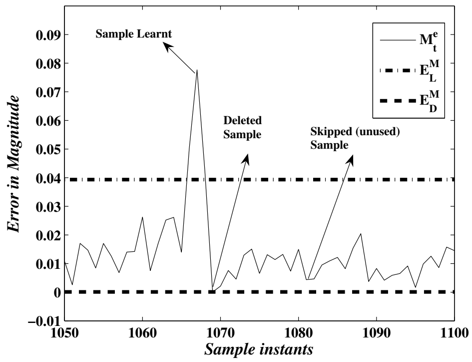
Similarly, Fig. 2(b) gives the snapshot of instantaneous phase error, the learning control parameter for phase (E_L^ϕ) and the deleting control parameter (E_D^ϕ) over 50 samples (sample instant 1050-1100) in learning during epoch 50. The effect of self-regulation using the instantaneous phase error of the samples is clearly seen from this plot.

- A few samples whose instantaneous magnitude error is greater than the learning control parameter for magnitude (E_L^ϕ) participated in learning. Eg. sample instant 1065
- A few samples whose instantaneous magnitude error is lesser than the deleting control parameter for magnitude (E_D^ϕ) are deleted from the training dataset. Eg. Sample instant 1069.
- There are a few samples whose instantaneous magnitude error is greater than E_D^ϕ and lesser than E_L^ϕ (Eg. Sample instant 1068). These samples neither took part in learning, nor were deleted in the current epoch. Instead, they were skipped in the current epoch, to be presented to the network in future.

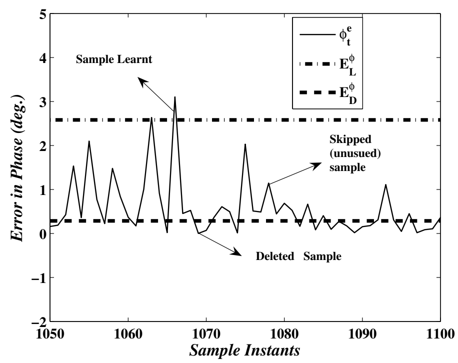
Figures 3 give the sample history for the number of samples that participated in learning and those that were not used, at an interval of 100 epochs, over 5000 epochs. From Fig. 3, it can be observed that, on an average, only 600 samples participated in learning in each epoch. Hence, it can be inferred that only 600 samples contained vital information about the function to be approximated (CFAP-I).

In Fig. 3, the total number of samples deleted at the end of each epoch and the number of unused samples in each epoch is also presented. It can be observed that though only a few samples were deleted in each epoch, most samples were neither deleted, nor learnt. These samples that did not satisfy both the learning and the delete criteria, were skipped during training in each epoch.

The performance study results on CFAP-I are presented in Table I. The performance of the proposed algorithm has been compared against other batch learning complex-valued RBF learning algorithms available in literature. It was also observed that during training, on an average only **540** samples participated in learning in each epoch. The remaining samples were either deleted during training or were skipped in learning, as they contained similar information already acquired by the network. The performance of the FC-RBF



(a) A snap shot of instantaneous magnitude error from 1050-1100 sample instants



(b) A snap shot of instantaneous phase error from 1050-1100 sample instants

Fig. 2: CFAP-I: Self-Regulation

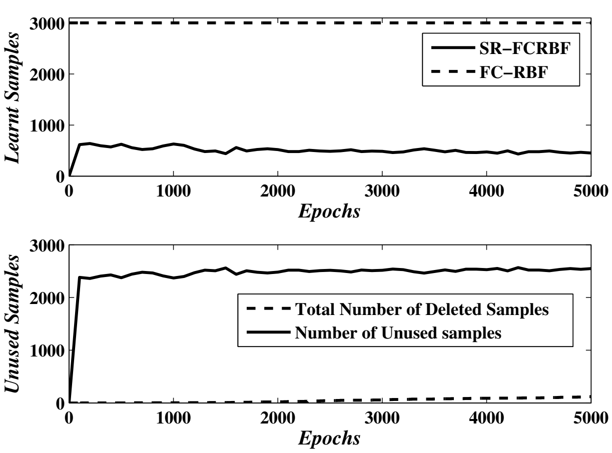


Fig. 3: CFAP-I: Sample History- Used and Unused Samples

algorithm is better than other complex-valued RBF learning algorithms available in literature. Comparing its performance with the performance of the FC-RBF network with the self-regulation system, it can be observed that the self-regulatory system that operates on the FC-RBF network improves its magnitude and phase approximation performance at least by one order. The approximation ability of the FC-RBF network has been improved by almost 90% in the CFAP-I. Also, it has reduced the computational time of the FC-RBF network at least by 400 seconds. It is notable that, though only 540 samples participated in learning, on an average, the reduction in computational time is not significant. This is so because the network performs forward computation on all the samples present in the training dataset. The unused samples only do not go through the backward computation.

TABLE I: Performance Comparison for CFAP-I

Algorithm	Time (sec)	K	Training		Testing	
			J_{Me}	ϕ_e (deg.)	J_{Me}	ϕ_e (deg.)
CRBF	9233	15	0.15	51	0.18	52
CELM	0.2	15	0.19	90	0.23	88
CMRAN	52	14	0.026	2	0.48	19
FC-RBF	6013	20	0.019	16	0.048	16
SRFCRBF	5620	15	0.0009	0.5	0.0009	0.6

Fig. 4 gives the magnitude and phase error convergence plots of the FC-RBF network with and without the self-regulation, over a window of 1000 epochs, for the CFAP-I. It can be observed from the plots that both the magnitude and phase errors of the FC-RBF network converged faster with the self-regulation system. This verifies the performance results tabulated in I, where the errors of the FC-RBF network is lesser at least by an order, when samples are selected for learning using the self-regulation system.

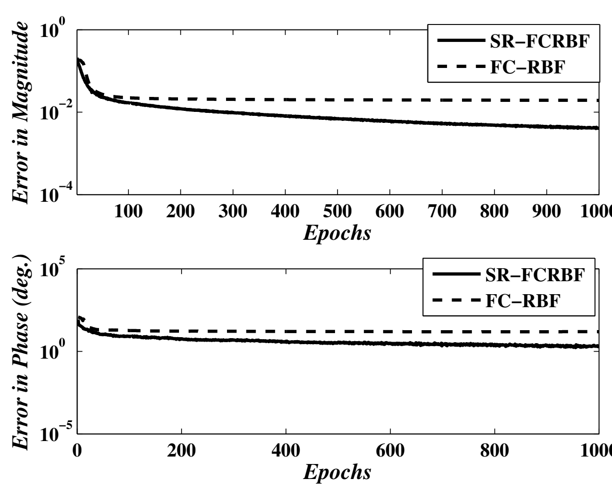


Fig. 4: CFAP-I: Error convergence plots

2) CFAP-II: The other synthetic complex-valued function approximation problem (CFAP-II) defined in [9] as

$$f_1(\mathbf{z}) = \frac{1}{6.25}\left(Z_1^2 + Z_2^2\right) \tag{13}$$

where $Z_i \in \mathbb{C}$. Here, 3000 randomly generated training samples and 1000 randomly generated testing samples were used to study the approximation ability of the algorithm.

In Figure 5 the history of used samples (samples that took part in learning) over 5000 epochs has been presented, in an interval of 100 epochs. From Fig. 5, it can be seen that the number of samples that took part in the network learning decreases gradually, as learning progresses. On an average, **520** samples participated in learning in all the epochs. This selective participation of samples is aided by the self-regulated magnitude and phase learning control parameters.

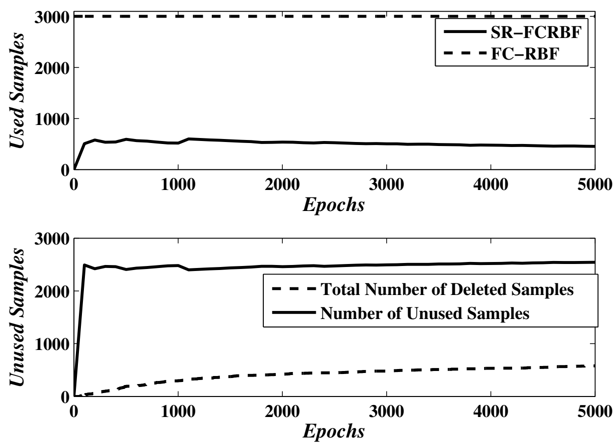


Fig. 5: CFAP-II:Sample History-Used and Unused Samples

The figure also gives the total number of samples deleted at the end of each epoch and the number of samples unused in each of the 5000 epochs. It can be observed from the figure, that though no samples were deleted in the initial epochs, as the network parameters were updated based on gradient descent algorithm of the FC-RBF, the self-regulatory system deletes samples which contain similar information already learnt by the network. In all, a total of 580 samples were deleted at the end of training over 5000 epochs. Also, similar to the CFAP-I, a few samples neither took part in learning, nor were deleted in each epoch. These samples just remained in the training dataset, to be learnt in the future epochs.

Figure 6 gives the magnitude and error convergence plot of FC-RBF with and without the self-regulation scheme for CFAP-II, over a window of the initial 1000 epochs. It can be observed that the self-regulation system increased the speed of convergence of both the magnitude and phase errors of the FC-RBF network, though by a very small interval.

Table II gives the performance of FC-RBF network with the self-regulation system, compared against other complex-valued RBF learning algorithms. It can be observed from the Table that of the complex-valued RBF learning algorithms available in literature, the FC-RBF network performed better.

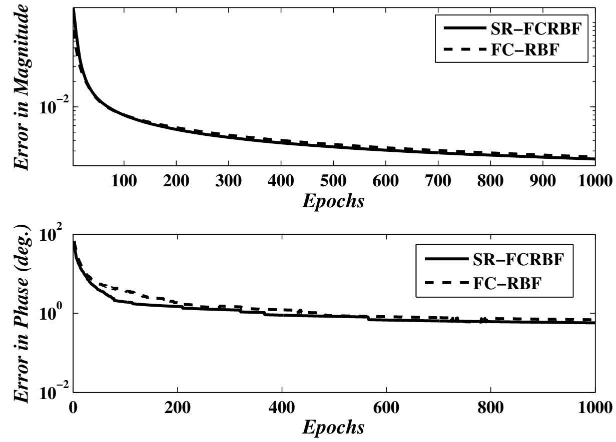


Fig. 6: CFAP-II:Error Convergence Plots

It is also observable that the self-regulation system has improved the training/generalization performance of the FC-RBF network for the CFAP-II. The self-regulation system improved the generalized phase approximation performance of the FC-RBF network, at least by 40%. Also, as a few samples are skipped in learning in each epoch, the computational time required in training the FC-RBF network is reduced at least by 200 seconds.

TABLE II: Performance Comparison for CFAP-II

Network	Time	K (sec.)	Training		Testing	
			J_{Me}	ϕ_e	J_{Me} (deg.)	ϕ_e (deg.)
CRBF	5020	20	0.592	45	0.6	47
CELM	0.22	20	0.688	35	0.7	36
CMRAN	84	27	0.068	13	0.07	16
FC-RBF	1341	15	0.002	0.34	0.003	0.34
SR-FCRBF	1152	15	0.005	0.3	0.002	0.18

* Number of samples used.

B. Two spiral classification problem

The two-spiral problem is a famous benchmark problem, because it is an extremely difficult nonlinear classification problem available in literature. Though several variations of the two-spiral problem have been briefly discussed [18], the original two-spiral classification problem has been considered in this study.

The original two-spiral data can be generated using a C-routine, available in the CMU repository. The number of data points and the range of values generated are determined by the parameters, density and radius respectively. Here, 194 training points are generated with the default parameters of density 1 and radius 6.5. The 194 data points are equally shared between the two spirals (97 data points each). Each spiral has three turns of 32 points per turn, and one end point. Both the spirals have the same origin and orientation, but are displaced by 180 degrees to each other. The first spiral is generated by the set of equations in eq. (14)

$$\begin{aligned} \phi &= \frac{i}{16}.\pi \\ r &= \frac{d.(104-i)}{104]} \\ x = r\ \cos\phi; y &= r\ \sin\phi \\ z &= x+iy \end{aligned} \tag{14}$$

The second spiral can be obtained by plotting $(-x,-y)$ and hence $z1 = -x - iy$. The problem at hand is to classify the points into their respective spirals, which makes up a highly non-linear two class classification problem of complex-valued data, with one input neuron and one output neuron. Rigorous study was done to identify the number of neurons required to achieve 100% classification accuracy. The number of neurons and number of epochs required to achieve the classification accuracy of 100% was observed. In Table III, the number of neurons and number of epochs required by each complex-valued RBF network to achieve a classification accuracy of 100% is tabulated. It can be observed that with the self-regulatory system, the performance of the FC-RBF network has improved in that, it required at least 10 lesser neurons to achieve 100% classification accuracy, in lesser time, compared to other batch learning complex-valued RBF learning algorithms, except for C-ELM. C-ELM being an analytical learning algorithm, requires the least time for classification. Figs. 7(a) and 7(b) gives the initial spiral points before classification and the final spiral points after classification into their respective spirals. It can be seen that at 100% classification accuracy, all the points are distinctly classified in their respective spirals.

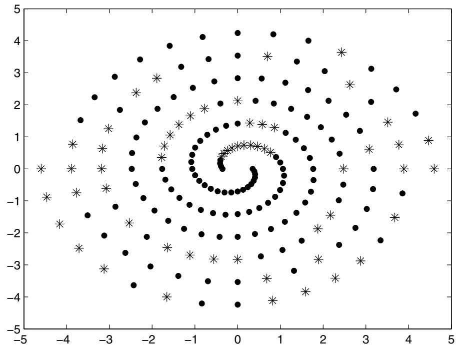
TABLE III: Performance Comparison for Two-Spiral Classification Problem

Network	h	epochs	Training Time
CRBF	100	7226	10865.21
CMRAN	130	—	12.766
CELM	120	—	0.3281
FC-RBF	80	7829	3183.6
SR-FCRBF	70	6333	3133

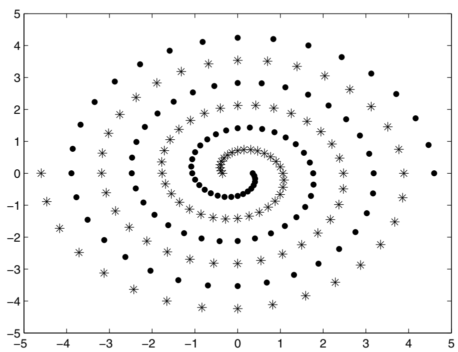
In Fig. 8, the sample history of used and unused samples during the entire training is presented. It can be observed that no samples were deleted during training, although, there were few samples that satisfied neither the learn criteria nor the deletion criteria, and hence were neither deleted nor learnt in each epoch.

IV. Conclusions

The paper presents a self-regulatory system for the Fully Complex-valued RBF network. The algorithm chooses the samples for learning, based on the instantaneous magnitude



(a) Initial Spiral



(b) Final Spiral with 100% Classification Accuracy

Fig. 7: Two Spirals (a) Initial Spiral points (b) Final Spirals after classification

and phase error of the sample, in each epoch. If the magnitude or phase error are greater than their respective self-regulated learning threshold parameters, the sample is learnt. Samples with errors lesser than the deleting threshold are deleted from the training dataset. Samples which do not satisfy both the learn and delete criterion are skipped in learning, in each epoch. Thus, the algorithm deletes similar samples and hence, prevents overtraining. Performance of the proposed algorithm is compared with other complex-valued RBF network on two benchmark function approximation problems and two-spiral classification problem. Based on the study, it can be inferred that the self-regulation system improves the generalization performance of the FC-RBF network and also reduces the computational time and effort in training.

References

[1] J. C. Bregains and F. Ares, "Analysis, synthesis, and diagnostics of antenna arrays through complex-valued neural networks,", *Microwave and Optical Technology Letters*, vol. 48, no. 8, pp. 1512-1515, 2006.

[2] M. Ceylan, R. Ceylan, Y. Ozbay, S. Kara, "Application of complex discrete wavelet transform in classification of Doppler signals using complex-valued artificial neural network,", *Artificial Intelligence in Medicine*, vol. 44, no. 1, pp. 65-76, 2008.

[3] H. Leung and S. Haykin, "Complex backpropagation algorithm,", *IEEE Trans. on Signal Processing*, vol. 39, no. 9, pp. 2101-2104, 1991.

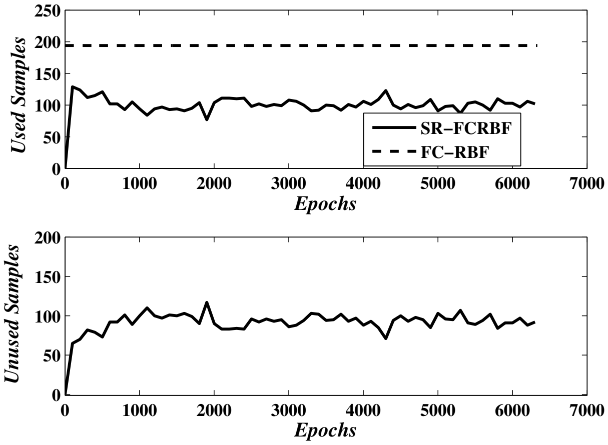


Fig. 8: Two-Spiral: Sample History- Used and Unused Samples

[4] A. Uncini, L. Vecci, P. Campolucci and F. Piazza, "Complex-valued neural networks with adaptive spline activation function for digital radio links nonlinear equalization,", *IEEE Trans. Signal Process.*, vol. 47, no. 2, pp. 505-514, 1999.

[5] S. S. Yang, C. L. Ho, and S. Siu, "Sensitivity analysis of the split-complex valued multilayer perceptron due to the errors of the *i.i.d.* Inputs and weights," *IEEE Trans. on Neural Networks*, vol. 18, no. 5, pp. 1280-1293, 2007.

[6] T. Kim, and T. Adali, "Fully-complex multilayer perceptron network for nonlinear signal processing,", *Journal of VLSI, Signal Processing*, vol. 32, no. 1-2, pp. 29-43, 2002.

[7] D. V. Loss, M. C. F. DeCastro, P. R. G. Franco and F. C. C. DeCastro, "Phase transmittance RBF neural networks," *Electronics Letters* , vol. 43, no. 16, pp. 882-884, 2007.

[8] G. Georgiou and C. Koutsougeras, "Complex backpropagation,", *IEEE Trans. on Circuits and Systems II*, vol. 39, no.5, pp. 330-334, 1992.

[9] R. Savitha, S. Suresh, N. Sundararajan and P. Saratchandran, "A new learning algorithm with logarithmic performance index for complex-valued neural networks,", *Neurocomputing*, vol. 72, no. 16-18, pp. 3771-3781, 2009.

[10] S. Chen, S. McLaughlin, and N. Mulgrew, "Complex-valued radial basis function network, Part I: Network architecture and learning algorithms,", *EURASIP Signal Processing Journal*, vol. 35, no. 1, pp. 19-31, 1994.

[11] D. Jianping, N. Sundararajan, and P. Saratchandran, "Complex-valued minimal resource allocation network for nonlinear signal processing,", *International Journal of Neural Systems*, vol. 10, no. 2, pp. 95-106, 2000.

[12] M. B. Li, G. B. Huang, P. Saratchandran, and N. Sundararajan, "Fully complex extreme learning machines,", *Neurocomputing*, vol. 68, pp. 306-314, 2005.

[13] R. Savitha, S. Suresh and N. Sundararajan, "A fully complex-valued radial basis function network and its learning algorithm,", *International Journal of Neural Systems*, vol. 19, no. 4, pp. 253 - 267, 2009.

[14] R. E. Schapire, "A brief introduction to boosting,", *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, Morgan Kaufmann, pp. 1401-1406, 1999.

[15] S. Zhong and J. Ghosh, "Decision boundary focused neural network classifier,", *Intelligent Engineering Systems Through Artificial Neural Networks (ANNIE)*, ASME Press,2000.

[16] Aggelos Chariatis, "Very fast online learning of highly non-linear problems,", *Journal of Machine Learning Research*, vol.8, Sep., pp. 2017-2045, 2007.

[17] 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*, vol. 7, no. 8, pp. 595-602, 2003.

[18] Stephan K. Chalup and Lukasz Wiklendt, "Variations of the two-spiral task," *Connection Science* , vol. 19, No. 2, pp. 183-199, 2007.