Third International Workshop on Advanced Computational Intelligence
August 25-27, 2010 - Suzhou, Jiangsu, China

Complex Neuro-Fuzzy Intelligent Approach to Function Approximation

Chunshien Li, Tai-Wei Chiang, Jhao-Wun Hu and Tsunghan Wu

***Abstract*—A complex neuro-fuzzy self-learning approach using complex fuzzy sets to the problem of function approximation is proposed in this paper. The concept of complex fuzzy sets (CFSs) is an extension of traditional fuzzy set whose membership degrees are within a unit disk in the complex plane. The Particle Swarm Optimization (PSO) algorithm and the recursive least square estimator (RLSE) algorithm are used in hybrid way to train the proposed complex neuro-fuzzy system (CNFS). The PSO is used to adjust the premise parameters of the CNFS, and the RLSE is used to update the consequent parameters. With the experimental results, the CNFS shows better performance than the traditional neuro-fuzzy system (NFS) that is designed with regular fuzzy sets. Moreover, the PSO-RLSE hybrid learning method for the CNFS improves the rate of learning convergence and shows better performance in accuracy. In order to test the feasibility and approximation performance of the proposed approach, two benchmark functions are used for the proposed approach. The results by the proposed approach compared to other approaches. Excellent performance by the proposed approach has been observed.**

I. Introduction

THEORIES of fuzzy logic and neural networks for system identification or system modelling have been widely investigated for applications [1]-[2], such as communication, pattern classification, expert system, medical engineering, and many others. A reliable model can be used for the purpose of performance monitoring, fault detection, or performance optimization. By observing input-output data pairs for an unknown system of interest, a model can be set up for the system, using an intelligent modelling approach. With the model, the relationship of input-output behaviour can be approximated. This process can be viewed as system identification, which is also known as system modelling or function approximation. Thus, system or can be viewed as the problem of function approximation, for which an optimization process is usually involved to search for the optimal solution to the problem. However, due to the complexity and nonlinearity in real-world application problems, mathematical approaches for system identification are usually laborious and difficult. Since neural networks and fuzzy inference systems are universal approximator [3], neuro-fuzzy systems, which incorporate the advantages of fuzzy inference, neural structure and learning flexibility, have become popular and fundamental issues in modelling problems. Fuzzy sets are simple extension of crisp sets. For fuzzy sets, the belongingship is expanded from Boolean-valued state to real-valued state. Each element of fuzzy set can be given a membership value between 0 and 1. Fuzzy sets can be used to reflect human concepts and thoughts, which tend to be imprecise, incomplete and vague. Complex fuzzy set (CFS) [4]-[5] is a new development in the theory of fuzzy systems. The concept of CFS is an extension of fuzzy set, by which the membership for each element of a complex fuzzy set is extended to complex-valued state. In a complex fuzzy set, membership values are complex numbers in the unit disc of the complex plane [4]-[5]. The CFS membership degrees consist of an amplitude and a phase in the complex plane [4]. Although the introductory theory of the CFS has been presented [5], the research on complex fuzzy system designs and applications using the concept of CFS is found rare. The design of a complex fuzzy inference system is complicated, and it is hard to construct intuitively understandable complex fuzzy sets.

In this paper, a complex neuro-fuzzy system with a hybrid learning method is proposed to the problem of function approximation. The hybrid learning method includes the well-known particle swarm optimization (PSO) algorithm and the recursive least square estimator (RLSE) algorithm to train the proposed the complex neuro-fuzzy systems (CNFS). PSO is a population based heuristic method and it is used to update the premise parameters of the proposed CNFS in the learning process, and the RLSE is used to update the consequent parameters so that the CNFS parameters can be converged in fast way. Two benchmark functions are used in experiments to test the proposed approach for the ability in function approximation. The main contribution of the study is that a complex neuro-fuzzy approach is proposed to provide excellent flexibility in input-output mapping capability for function approximation. The proposed approach shows better adaptability in approximating capability than a traditional neuro-fuzzy system, in terms of approximation accuracy and learning convergence rate.

In section II, the proposed complex neuro-fuzzy approach is specified. In section III, the PSO-RLSE hybrid learning method is given. In section IV, experimental results for

Chunshien Li is with the Laboratory of Intelligent System and Applications, Department of Information Management, National Central University, Taiwan, ROC (phone: 03-422-7151; fax: 03-425-4604; e-mail: jamesli@mgt.ncu.edu.tw).

Tai-Wei Chiang is with the Laboratory of Intelligent System and Applications, Department of Information Management, National Central University, Taiwan, ROC (e-mail: taiwei.chiang@gmail.com).

Jhao-Wun Hu is with the Laboratory of Intelligent System and Applications, Department of Information Management, National Central University, Taiwan, ROC (e-mail: cwhterry@gmail.com).

Tsunghan Wu is with the Laboratory of Intelligent System and Applications, Department of Information Management, National Central University, Taiwan, ROC (e-mail: chonghang28@hotmail.com).

Acknowledgment:

This research work is supported by the National Science Council, Taiwan, ROC, under the Grant contract no. NSC98-2221-E-008-086.

function approximation are given. Finally, the paper is discussed and concluded.

II. Methodology for Complex Neuro-Fuzzy System

Neuro-fuzzy system (NFS) is a universal approximator that can approximate highly nonlinear functions with excellent accuracy, and it can deal with nonlinear estimation [6]. NFS is a multilayer feed-forward neural fuzzy network which combines neural network together with fuzzy reasoning to have excellently flexible mapping ability from input into output. There are two frequently used fuzzy inference systems (FISs). The first is Mandani type FIS and the other is Takagi-Sugeno (T-S) type FIS. The difference between them lies in the consequents of fuzzy rules. For Mamdani fuzzy model, the consequents are specified with linguistic terms, which can be defined with fuzzy sets [7]. For T-S fuzzy model [8], the consequents are expressed as polynomial functions of the input variables. In this paper, the design of the proposed CNFS in extended from the concept of traditional NFS, and the fuzzy T-S model is used in the proposed CNFS. For the proposed CNFS, complex fuzzy sets (CFSs) are used to have better adaptability in mapping ability.

A. Complex fuzzy set

The fuzzy theory can be used to represent uncertain or imprecise data, information and concept. The values of applying fuzzy sets for modelling uncertainty, for representing subjective human knowledge, and for emulating human reasoning processes, have been validated [4]. Mathematically, the concept of fuzzy set is an extension of classical set, and each element of a fuzzy set is not Boolean-valued state. The concept of fuzzy sets is again extended to complex fuzzy set (CFS), which expands the range of membership from the unit interval [0, 1] to the unit disc in the complex plane. In other words, membership degrees for the elements in a CFS are no longer real-valued state between 0 and 1, but they are extended to the complex-valued state in the unit complex disk.

Assume there is a complex fuzzy set S whose membership function is given as follows.

$$\begin{aligned}\mu_s(h) &= r_s(h)\exp(j\omega_s(h)) \\ &= \mathrm{Re}(\mu_s(h)) + j\,\mathrm{Im}(\mu_s(h)) \\ &= r_s(h)\cos(\omega_s(h)) + jr_s(h)\sin(\omega_s(h))\end{aligned} \tag{1}$$

where h is the base variable for the complex fuzzy set, rs(h) is the amplitude function of the complex membership, ωs(h) is the phase function. The complex fuzzy set S is expressed as follows.

$$S = \{(h, \mu_s(h)) \mid h \in U\} \tag{2}$$

The property of the waves appears obviously in the definition of complex fuzzy set, and the membership function is found with sinusoidal waves. In the case that $\omega_s(h)$ equals to 0, a traditional fuzzy set is regarded as a special case of a complex fuzzy set.

Assume there is a fuzzy rule with the form of " If (x_A=A and x_B=B) Then…", where A and B represent two different conditions in the rule. The two conditions can be described using two complex fuzzy sets, given as follows.

$$\mu_A(h_A) = r_A(h_A)\exp(j\omega_A(h_A)) \tag{3}$$

$$\mu_B(h_B) = r_B(h_B)\exp(j\omega_B(h_B)) \tag{4}$$

where h_A and h_B are the base variables; x_A and x_B are the linguistic variables for h_A and h_B, respectively.

Intersection of the complex fuzzy sets A and B is expressed as $A \cap B$, which is still a complex fuzzy set, defined as follows.

$$\mu_{A\cap B} \equiv [r_A(h_A) * r_B(h_B)]\exp(j\omega_{A\cap B}) \tag{5}$$

where * is for t-norm operator (intersection operator), $\omega_{A\cap B} = \wedge(\omega_A(h_A), \omega_B(h_B))$ is the intersection, and $\wedge(.,.)$ denotes the phase intersection operation.

Union of the complex fuzzy sets A and B is expressed as $A \cup B$, which is still a complex fuzzy set, given as follows.

$$\mu_{A\cup B} \equiv [r_A(h_A) \oplus r_B(h_B)]\exp(j\omega_{A\cup B}) \tag{6}$$

where $\oplus$ is for s-norm operator (union operator), $\omega_{A\cup B} = \vee(\omega_A(h_A), \omega_B(h_B))$ is the union of phase, and $\vee(.,.)$ is represented the union operator of phase.

B. Complex fuzzy reasoning for CNFS

Assume we have a complex fuzzy system with T-S fuzzy rules, given as follows.

Rule i: IF (x_1 is $A_1^i(h_1(t))$) and (x_2 is $A_2^i(h_2(t))$) …

and (x_M is $A_M^i(h_M(t))$) Then $z^i = a_0^i + \sum_{j=1}^{M} a_j^i h_j$ (7)

i =1,2,...,K

where x_j is the j-th input linguistic variable, h_j is the j-th input of base variables, $A^i_j(h_j)$ is the complex fuzzy set for the j-th condition in the i-th rule, z^i is the output of the i-th rule, and a^i_j, i=1,2...K and j=0,1,...M are the consequent parameters.

For the proposed CNFS, the complex fuzzy system is cast into the framework with six layered neuro-fuzzy network. The complex fuzzy reasoning for the CNFS from input to output is explained as follows.

Layer 0: The layer is called the input layer, which receives the inputs and transmits them to the next layer directly. The input vector is given as follows.

$$\boldsymbol{H}(t) = [h_1(t), h_2(t), \ldots, h_M(t)]^T \tag{8}$$

Layer 1: The layer is called the fuzzy-set layer. Nodes in the layer are used to represent the complex fuzzy sets for the premise part of the CNFS and to calculate the membership degrees.

Layer 2: This layer is for the firing-strengths. The firing strength of the i-th rule is calculated as follows.

$$\beta^i(t)=\mu_1^i(h_1(t))*\mu_2^i(h_2(t))*\cdots*\mu_M^i(h_M(t))$$
$$=\prod_{j=1}^{M} r_j^i(h_j(t))\exp(j\omega_{A_1^i\cap..\cap A_M^i}) \quad (9)$$
$$i=1,2,\ldots,K$$

where min operator is used for the t-norm calculation of the firing strength. r^i_j is the amplitude of complex membership degree for the j-th fuzzy set of the i-th rule.

Layer 3: This layer is for the normalization of the firing strengths. The normalized firing strength for the i-th rule is represented as follows.

$$\lambda^i(t)=\frac{\beta^i(t)}{\sum_{i=1}^{K}\beta^i(t)}=\frac{(\prod_{j=1}^{M} r_j^i(h_j(t)))\exp(j\omega_{A_1^i\cap..\cap A_M^i})}{\sum_{i=1}^{K}(\prod_{j=1}^{M} r_j^i(h_j(t)))\exp(j\omega_{A_1^i\cap..\cap A_M^i})} \quad (10)$$

Layer 4: The layer is for normalized consequents. The normalized consequent of the i-th rule is represented as follows.

$$\xi^i(t)=\lambda^i(t)\times z^i(t)$$
$$=\lambda^i(t)\times\left(a_0^i+\sum_{j=1}^{M}a_j^i h_j(t)\right) \quad (11)$$
$$=\frac{(\prod_{j=1}^{M} r_j^i(h_j(t)))\exp(j\omega_{A_1^i\cap..\cap A_M^i})}{\sum_{i=1}^{K}(\prod_{j=1}^{M} r_j^i(h_j(t)))\exp(j\omega_{A_1^i\cap..\cap A_M^i})}\times\left(a_0^i+\sum_{j=1}^{M}a_j^i h_j(t)\right)$$

Layer 5: This layer is called the output layer. The normalized consequents from Layer 4 are congregated in the layer to produce the CNFS output, given as follows.

$$\xi(t)=\sum_{i=1}^{K}\xi^i(t)=\sum_{i=1}^{K}\lambda^i(t)\times z^i(t)$$
$$=\sum_{i=1}^{K}\frac{(\prod_{j=1}^{M} r_j^i(h_j(t)))\exp(j\omega_{A_1^i\cap...\cap A_M^i})}{\sum_{i=1}^{K}(\prod_{j=1}^{M} r_j^i(h_j(t)))\exp(j\omega_{A_1^i\cap..\cap A_M^i})}\times\left(a_0^i+\sum_{j=1}^{M}a_j^i h_j(t)\right) \quad (12)$$

Generally the output of the CNFS is represented as follows.

$$\xi(t)=\xi_{\mathrm{Re}}(t)+j\xi_{\mathrm{Im}}(t)$$
$$=|\xi(t)|\times\exp(j\omega_\xi) \quad (13)$$
$$=|\xi(t)|\times\cos(\omega_\xi)+j\,|\xi(t)|\times\sin(\omega_\xi)$$

where $\xi_{\mathrm{Re}}(t)$ is the real part of the output for the CNFS, $\xi_{\mathrm{Im}}(t)$ is the imaginary part, the magnitude of the complex output is given in (14), and the phase of the complex output is expressed in (15).

$$\xi(t)=\sqrt{(\xi_{\mathrm{Re}}(t))^2+(\xi_{\mathrm{Im}}(t))^2} \quad (14)$$

$$\omega_\xi=\tan^{-1}(\frac{\xi_{\mathrm{Im}}}{\xi_{\mathrm{Re}}}) \quad (15)$$

Based on (12), the complex inference system can be viewed as a complex function system, expressed as follows.

$$\xi(t)=F(H(t),W)=F_{\mathrm{Re}}(H(t),W)+jF_{\mathrm{Im}}(H(t),W) \quad (16)$$

where $F_{\mathrm{Re}}(.)$ is the real part of the CNFS output, $F_{\mathrm{Im}}(.)$ is the imaginary part of the output, $H(t)$ is the input vector to the CNFS, W denotes the parameter set of the CNFS. The parameter set W can be divided into two subsets, which are the premise-part subset and the consequent-part subset, denoted as W_{If} and W_{Then}, respectively.

III. Hybrid PSO-RLSE Learning for CNFS

A. Particle swarm optimization

Particle swarm optimization (PSO) was developed by Eberhart and Kennedy [9] . PSO is a population-based optimization method, which is motivated by the food searching behaviour of bird flocking or fish schooling. Each bird in the swarm in viewed as a particle. Assume the location of food is viewed as the optimal solution in the problem space. Each particle is viewed as a potential solution in the search space. All particles in the swarm have their locations and velocities. Each particle location can be mapped to a fitness (or called a cost) with some given fitness function (or called cost function). The particles in the swarm complete to each other to become the winner, which is known as gbest. The best location of a particle during the evolution process is called pbest. The location and the velocity of a particle in the swarm are updated using the information of gbest and its pbest. Assume the problem space is with Q dimensions. The method of PSO is expressed as follows.

$$\boldsymbol{V}_i(k+1)=\boldsymbol{V}_i(k)+c_1\cdot\xi_1\cdot(\boldsymbol{pbest}_i(k)-\boldsymbol{L}_i(k))+c_2\cdot\xi_2\cdot(\boldsymbol{gbest}(k)-\boldsymbol{L}_i(k)) \quad (17a)$$

$$\boldsymbol{L}_i(k+1)=\boldsymbol{L}_i(k)+\boldsymbol{V}_i(k+1) \quad (17b)$$

$$\boldsymbol{V}_i(k)=[v_{i,1}(k),v_{i,2}(k),\ldots,v_{i,Q}(k)]^{\mathrm{T}} \quad (18)$$

$$\boldsymbol{L}_i(k)=[l_{i,1}(k),l_{i,2}(k),\ldots,l_{i,Q}(k)]^{\mathrm{T}} \quad (19)$$

where $\boldsymbol{V}_i(k)$ is the velocity for the i-th particle on k-th iteration, $\boldsymbol{L}_i(k)$ is the location for the i-th particle, $\{c_1, c_2\}$ are the parameters for PSO, and $\{\xi_1, \xi_2\}$ are random numbers in [0,1].

B. Recursive Least Square Estimator

In this study, the RLSE [10]-[11] is used for the identification of the parameters of consequent part. For a general least-squares estimation problem, the output of a linear model, y, is specified by the linearly parameterized expression, given as follows.

$$y=\theta_1 f_1(u)+\theta_2 f_2(u)+\cdots+\theta_m f_m(u), \quad (20)$$

where u is the model's input, $f_i(.)$ is known function of u and θ_i , i=1,2,…,m represents unknown parameters to be estimated. Here θ_i can be viewed as the consequent parameters of the proposed T-S fuzzy approximator. To estimate the unknown parameters $\{\theta_i, i=1,2,\ldots,m\}$ of a unknown target system (or function), first have to sample (observe) the input-output behaviour of the target system to collect a set of input-output data pairs, which will be used as training data, denoted as follows.

$$TD = \{(u_i, y_i), i = 1,2,\ldots,N\} \tag{21}$$

Substituting data pairs into (20) yield a set of N linear equations, given as follows.

$$\begin{aligned} f_1(u_1)\theta_1 + f_2(u_1)\theta_2 + \cdots + f_m(u_1)\theta_m &= y_1 \\ f_1(u_2)\theta_1 + f_2(u_2)\theta_2 + \cdots + f_m(u_2)\theta_m &= y_2 \\ \vdots \qquad\quad \vdots \qquad\qquad\quad \vdots \quad &\quad \vdots \\ f_1(u_N)\theta_1 + f_2(u_N)\theta_2 + \cdots + f_m(u_N)\theta_m &= y_N \end{aligned} \tag{22}$$

In a matrix form (22), can be given as follows.

$$\boldsymbol{A\theta} = \boldsymbol{y} \tag{23}$$

where $\boldsymbol{A}$, $\boldsymbol{\theta}$, and $\boldsymbol{y}$ are expressed as follows.

$$\boldsymbol{A} = \begin{bmatrix} f_1(u_1) & f_2(u_2) & \cdots & f_m(u_1) \\ f_1(u_2) & f_2(u_2) & \cdots & f_m(u_2) \\ \vdots & \vdots & \cdots & \vdots \\ f_1(u_N) & f_2(u_N) & & f_m(u_N) \end{bmatrix} \tag{24}$$

$$\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_m]^{\mathrm{T}} \tag{25}$$

$$\boldsymbol{y} = [y_1, y_2, \ldots, y_N]^{\mathrm{T}} \tag{26}$$

The optimal estimation for $\boldsymbol{\theta}$ in (23), can be calculated using the following RLSE equations.

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_k - \frac{\boldsymbol{P}_k \boldsymbol{b}_{k+1} \boldsymbol{b}_{k+1}{}^{\mathrm{T}} \boldsymbol{P}_k}{1 + \boldsymbol{b}_{k+1}{}^{\mathrm{T}} \boldsymbol{P}_k \boldsymbol{b}_{k+1}}, \tag{27a}$$

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{P}_{k+1} \boldsymbol{b}_{k+1} (\boldsymbol{y}_{k+1} - \boldsymbol{b}_{k+1}{}^{\mathrm{T}} \boldsymbol{\theta}_k) \tag{27b}$$

$$k = 0,1,\ldots,N-1$$

where $[\boldsymbol{b}^{\mathrm{T}}{}_k, \boldsymbol{y}_k]$ in the k-th row of $[\boldsymbol{A}, \boldsymbol{y}]$. To start the RLSE algorithm in (27), we need to select the initial values for $\boldsymbol{\theta}_0$ and $\boldsymbol{P}_0$ is given as follows.

$$\boldsymbol{P}_0 = \alpha \boldsymbol{I} \tag{28}$$

where α is a large value and $\boldsymbol{I}$ is the identity matrix, and θ_0 is initially set to zeros.

C. Hybrid Learning for CNFS

For the training of the proposed CNFS, the hybrid PSO-RLSE learning method is applied to update the premise parameters and the consequent parameters. In hybrid way, the PSO is used with the RLSE for fast learning convergence. The premise parameters and the consequent parameters of the CNFS are updated by the PSO given in (17) and the RLSE given in (27), respectively.

The PSO in (17) is used to update the premise parameters of the CNFS. The PSO is a heuristic method retaining characteristics of evolutionary search algorithms. Each location by ***gbest*** of the PSO provides a potential premise solution. Based on the premise parameters, the normalized firing strengths in layer 3 are calculated. After the premise structure is determined, next task is to identify the parameters of the consequent part. The RLSE is used to update the consequent parameters, with the normalized firing strengths. With (27), to estimate the consequent parameters, the row vector $\boldsymbol{b}$ and the vector $\boldsymbol{\theta}$ are arranged as follows.

$$\boldsymbol{b}_{k+1} = [\boldsymbol{bb}^1(k+1) \quad \boldsymbol{bb}^2(k+1) \quad \cdots \quad \boldsymbol{bb}^K(k+1)] \tag{29}$$

$$\boldsymbol{bb}^i(k+1) = [\lambda^i \quad h_1(k+1)\lambda^i \quad \cdots \quad h_M(k+1)\lambda^i] \tag{30}$$

$$\boldsymbol{\theta} = [\boldsymbol{\tau}^1 \quad \boldsymbol{\tau}^2 \quad \cdots \quad \boldsymbol{\tau}^K] \tag{31}$$

$$\boldsymbol{\tau}^i = [a_0^i \quad a_1^i \quad \cdots \quad a_M^i] \tag{32}$$

$$i = 1,2,\ldots,K$$

$$k = 0,1,\ldots,N-1$$

At each iteration for the hybrid PSO-RLSE learning, the output of the CNFS approximator (13). The error between the output of the target and the CNFS are given as follows.

$$\begin{aligned} MSE &= \left(\frac{1}{N}\sum_{t=1}^{N}(d(t) - \xi(t))^2\right) \\ &= \left(\frac{1}{N}\sum_{t=1}^{N}(d(t) - F(H(t),W))^2\right) \end{aligned} \tag{33}$$

The error is used further to define the mean square error (MSE), which is used as the cost function in the study. The cost in MSE is used as the performance index in the learning process for the proposed CNFS.

IV. EXPERIMENTATION FOR THE PROPOSED APPROACH

Experiments for function approximation are conducted in this section to estimate and verify the performance of the proposed approach. There are two tasks in this section. The first is to compare the PSO, as defined in (17), to the hybrid self-learning approach proposed for CNFS. The PSO is a population based heuristic method, but it is time-consuming and falls into the local minimum easily if dimensions are increasing. Hence, the RLSE, as defined in (27), is included in the proposed approach to improve the rate of learning convergence, for better performance in accuracy. The comparison of PSO to the hybrid PSO-RLSE method is the first task to verify the hybrid approach performance. The second task is to use the hybrid PSO-RLSE for both CNFS and NFS, and to compare their performance. In this paper, complex fuzzy sets are used to replace traditional fuzzy for the proposed CNFS. The complex fuzz sets which are extended of traditional fuzzy sets have superior flexibility. Therefore, the second task is compared CNFS and NFS with hybrid proposed approach to verify the performance of the complex fuzzy sets. Two benchmark functions are used as the examples for performance comparison, listed in Table I.

The training and testing data for cosine function are 400 and 200 sampled pairs and the anther function ($u^{2/3}$) are 201 and 402 sampled pairs. For each of the benchmark functions,

TABLE I
FIVE BENCHMARK FUNCTIONS

Benchmark function	domain
$cos(u)$	[-2, 2]
$u^{2/3}$	[-2, 2]

TABLE II
SETTINGS FOR PSO IN CNFS

Dimensions of particle	79
Swarm size	100
Initialization of particle position	Random in [0, 1]
Initialization of particle velocity	Random in [0, 1]
Learning rate (c_1, c_2)	2

the error norm for function approximation is based on the mean square errors (MSE), as defined in (33). There are two inputs, and each input to the CNFS and the traditional NFS possesses three fuzzy sets. There are 9 rules in the CNFS and the NFS, where 12 premise parameters and 27 consequent parameters are to be updated by the hybrid PSO-RLSE. In the hybrid PSO-RLSE for the CNFS, the PSO is used to adjust the 12 premise parameters of the CNFS and the RLSE is to update the 27 consequent parameters. For the training process, an auto-control mechanism is used to set up the initial iterations $M = 30$ and extra iterations $N=20$ in the experiment. If cost is improved, then the amount of iterations M is extended to $M=M+N$. Moreover, we set the limited times to extend M to 3. For the output of the CNFS, we select the real part to represent the approximator output.

A. The comparison of PSO to the proposed hybrid PSO-RLSE approach

For the first task, main purpose is to demonstrate that the hybrid PSO-RLSE can improve the performance effectively. The cosine function, listed in Table I, is used as the example to test the PSO and the hybrid PSO-RLSE for CNFS. The settings of the PSO for the CNFS approach are given in Table II. And the settings of the PSO-RLSE for the proposed CNFS are given in Table III. The proposed hybrid learning approach get large scales ahead than the PSO learning method at the first iteration, which the hybrid method is about 6.72×10^{-9} and the PSO method is more or less 17.63. And the proposed hybrid learning approach converged to near optimal solution with about 30 iterations (totally 190 iterations were implemented), while compared the PSO learning method used about 700 iterations and it still didn't get comparable performance.

TABLE III
SETTINGS FOR HYBRID PSO-RLSE IN CNFS

PSO	
Dimensions of particle	25
Swarm size	100
Initialization of particle position	Random in [0, 1]
Initialization of particle velocity	Random in [0, 1]
Learning rate (c_1, c_2)	2
RLSE	
Number of consequent-part parameters	27
$\boldsymbol{\theta}_0$	27×1 zero vector
$\boldsymbol{P}_0$	$\alpha\boldsymbol{I}$
α	10^8
$\boldsymbol{I}$	27×27 identity matrix

B. The compared CNFS and NFS with the proposed hybrid learning approach

In this paper, the complex fuzzy sets are designed for the CNFS. The primary aim in this task is in order that test the performance of the CNFS and traditional NFS. Moreover, the proposed approach is compared to the approach in [12]. The two benchmark functions are used as the examples for performance comparison, listed in Table I. With the two benchmark function, Table IV and V show the performance comparisons in the MSE. For cosine function that using 20 experimental trials using the proposed approach. For the $u^{2/3}$ function, the results by the proposed approach are compared to the compared approaches in [12]. An approximation response for the cosine functions by the proposed approach for the CNFS is shown in Figs 1. With the Table IV and V, it is shown using the same proposed hybrid learning approach for the CNFS is obtained lesser errors than using it for the NFS. Furthermore, the hybrid PSO-RLSE for the CNFS or the NFS can both get a wide margin ahead than the approach in [12].

TABLE IV
PERFORMANCE COMPARISON FOR COSINE FUNCTION

Method	MSE	
	Mean±std (training)	Mean±std (testing)
Method 1	$5.67\times10^{-10}\pm4.81\times10^{-10}$	$4.17\times10^{-7}\pm8.37\times10^{-7}$
Method 2	$5.81\times10^{-11}\pm4.30\times10^{-11}$	$1.45\times10^{-7}\pm3.72\times10^{-7}$

Method 1: The proposed hybrid learning approach for the NFS.
Method 2: The proposed hybrid learning approach for the CNFS.

V. DISCUSSION AND CONCLUSION

The proposed complex neuro-fuzzy system (CNFS) has been presented in the paper. The CNFS computing paradigm uses complex fuzzy sets to widen the plasticity of neuro-fuzzy system (NFS) for input-output mapping capability. Complex fuzzy sets are with complex-valued membership functions, which are extended from traditional fuzzy sets. This shift has widened the concept of membership between a set and its elements. Membership degrees of complex fuzzy sets are in complex-valued state in the unit disk of the

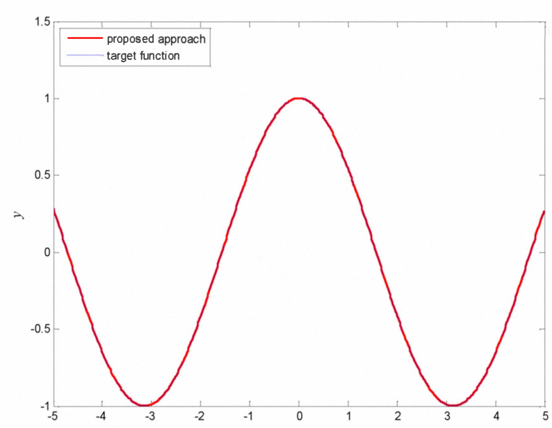

Fig. 1. Result by the proposed CNFS with hybrid learning approach for the cosine function.

TABLE V

PERFORMANCE COMPARISON FOR THE $u^{2/3}$ FUNCTION.

Method	rules	MSE	
		training	testing
FCRM [12]	2	N/A	0.0029
SONFIN [12]	2	N/A	0.0025
GA-PSO for FIS [12]	2	N/A	0.0016
Method 1	9	2.05×10^{-7}	6.60×10^{-7}
Method 2	9	3.47×10^{-8}	4.42×10^{-7}

Method 1: The proposed hybrid learning approach for the NFS.

Method 2: The proposed hybrid learning approach for the CNFS.

complex plane. The proposed approach has been applied to the problem of function approximation to verify the mapping performance of the proposed CNFS.

The hybrid PSO-RLSE learning method has been applied to the proposed CNFS to adapt its system parameters. The system parameters are divided into two subsets to make easier the learning process for the optimal solution to application performance. The two subsets are the premise set of parameters and the consequent set of parameters. The former subset includes the parameters in defining the premise fuzzy sets for the CNFS, and the later subset collects the consequent parameters in defining the consequent parts of the rules in the CNFS. The well-known PSO is used to update the premise subset of parameters and the RLSE is for the consequent subset of parameters. This hybrid learning method is very efficient to find the optimal (or near optimal) solution for the CNFS in application performance.

The experiments have shown that the proposed hybrid learning approach can converge to near optimal solution in about 30 iterations, while the compared PSO learning method uses more than 700 iterations and it still doesn't get comparable performance. Moreover, it is found in Table V that the proposed hybrid learning approach for CNFS is superior to the compared methods in [12]. For the cosine function, the proposed hybrid learning approach with MSE= 5.81×10^{-11} is much better than the traditional NFS with the proposed hybrid learning method with MSE= 5.67×10^{-10}. The performance comparisons are shown in Table IV and V, which two benchmark function are involved.

The complex neuro-fuzzy system is an adaptive computing paradigm that combines the theories of complex fuzzy logic and neural network. In order to develop the adaptability of the CNFS, the newly proposed PSO-RLSE hybrid learning algorithm has been used to tune the premise parameters and the consequent parameters in hybrid way to achieve fast and stable learning convergence. With the experimental results, the merits of the proposed hybrid learning have been observed. Through the comparison experiments, the proposed approach has shown excellent performance than the proposed CNFS has been presented in the paper. The CNFS computing paradigm uses complex fuzzy sets to widen the plasticity of the NFS for input-output mapping capability. The proposed system has been applied to the problem of function approximation and it has shown excellent mapping performance.

REFERENCES

[1] C. F. Juang and C. T. Lin, "An online self-constructing neural fuzzy inference network and itsapplications," *IEEE Transactions on Fuzzy Systems*, vol. 6, pp. 12-32, 1998.

[2] S. Paul and S. Kumar, "Subsethood-product fuzzy neural inference system (SuPFuNIS)," *IEEE Transactions on Neural Networks*, vol. 13, pp. 578-599, 2002.

[3] L. X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonalleast-squares learning," *IEEE Transactions on Neural Networks*, vol. 3, pp. 807-814, 1992.

[4] A. Kandel, D. Ramot, R. Milo, and M. Friedman, "Complex Fuzzy Sets," *IEEE Transactions on Fuzzy Systems*, vol. 10, pp. 171–186, 2002.

[5] S. Dick, "Toward complex fuzzy logic," *IEEE Transactions on Fuzzy Systems*, vol. 13, pp. 405-414, 2005.

[6] J. S. R. Jang, C. T. Sun, and E. Mizutani, *Neuro-fuzzy and soft computing*, Prentice Hall Upper Saddle River, NJ, 1997.

[7] W. A. Farag, V. H. Quintana, and G. Lambert-Torres, "A genetic-based neuro-fuzzy approach for modeling and control ofdynamical systems," *IEEE Transactions on Neural Networks*, vol. 9, pp. 756-767, 1998.

[8] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modelling and control," *IEEE transactions on systems, man, and cybernetics*, vol. 15, pp. 116-132, 1985.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," presented at Neural Networks, 1995. Proceedings., IEEE International Conference on, 1995.

[10] T. C. Hsia, *System identification: Least-squares methods*, D. C. Heath and Company, 1977.

[11] G. C. Goodwin and K. S. Sin, *Adaptive filtering prediction and control*, Prentice-Hall Englewood Cliffs, NJ, 1984.

[12] Z. J. Lee, " A novel hybrid algorithm for function approximation," *Expert Systems with Applications*, vol. 34, pp. 384-390, 2008.