Complex Neuro-Fuzzy Self-Learning Approach to Function Approximation

Chunshien Li and Tai-Wei Chiang

Laboratory of Intelligent Systems and Applications
Department of Information Management
National Central University, Taiwan, ROC.
jamesli@mgt.ncu.edu.tw

Abstract. A new complex neuro-fuzzy self-learning approach to the problem of function approximation is proposed, where complex fuzzy sets are used to design a complex neuro-fuzzy system as the function approximator. Particle swarm optimization (PSO) algorithm and recursive least square estimator (RLSE) algorithm are used in hybrid way to adjust the free parameters of the proposed complex neuro-fuzzy systems (CNFS). The hybrid PSO-RLSE learning method is used for the CNFS parameters to converge efficiently and quickly to optimal or near-optimal solution. From the experimental results, the proposed 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. Three benchmark functions are used. With the performance comparisons shown in the paper, excellent performance by the proposed approach has been observed.

Keywords: complex fuzzy set, complex neuro-fuzzy system (CNFS), PSO, RLSE, function approximation, machine learning.

1 Introduction

Theories of fuzzy logic and neural networks for system identification or system modeling have been widely investigated for applications [1]-[2]. By observing input-output data pairs for an unknown system of interest, a model can be set up for the system, using an intelligent modeling approach. With the model, the relationship of input-output behavior can be approximated. This process can be viewed as system identification, which is also known as system modeling or function approximation. Thus, system identification or modeling 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]-[4], neuro-fuzzy systems, which incorporate the advantages of fuzzy inference, neural structure and learning flexibility, have become popular and fundamental issues in modeling problems. Fuzzy sets can be used to reflect human concepts and thoughts, which tend to be imprecise, incomplete and vague. Complex fuzzy set (CFS) [5]-[7] 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 [5]-[6]. 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 rarely. In this paper, a complex neuro-fuzzy system (CNFS) 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 CNFS. 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 2 the proposed complex neuro-fuzzy approach is specified. In section 3 the PSO-RLSE hybrid learning method is given. In section 4 experimental results for function approximation are given. Finally, the paper is discussed and concluded.

2 Methodology for Complex Neuro-Fuzzy System

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 [8]. For T-S fuzzy model [9], 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 neuro-fuzzy system (NFS), and the fuzzy T-S model is used in the proposed CNFS. The fuzzy theory can be used to represent uncertain or imprecise data, information and concept. The values of applying fuzzy sets for modeling uncertainty, for representing subjective human knowledge, and for emulating human reasoning processes, have been validated [5]. The concept of fuzzy sets is extended to complex fuzzy set (CFS) [5]-[7], which expands the range of membership from the unit interval [0, 1] to the unit disc in the complex plane. 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, $r_S(h)$ is the amplitude function of the complex membership, $\omega_S(h)$ is the phase function. The complex fuzzy set S is expressed as follows.

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

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 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 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. Assume we have a complex fuzzy system with K 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 $\quad z^i = a_0^i + \sum_{j=1}^{M} a_j^i h_j$ (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.

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

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.

$$\begin{aligned}\beta^i(t) &= \mu_1^i(h_1(t)) * \mu_2^i(h_2(t)) * \cdots * \mu_M^i(h_M(t)) \\ &= \mathop{\wedge}_{j=1}^{M} r_j^i(h_j(t))\exp(j\omega_{A_1^i\cap\ldots\cap A_M^i})\end{aligned} \tag{9}$$

$i = 1,2,...,K$, 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.

$$\lambda^i(t) = \frac{\beta^i(t)}{\sum_{i=1}^{K}\beta^i(t)} = \frac{(\overset{M}{\underset{j=1}{\wedge}} r_j^i(h_j(t)))\exp(j\omega_{A_1^i\cap\ldots\cap A_M^i})}{\sum_{i=1}^{K}(\overset{M}{\underset{j=1}{\wedge}} r_j^i(h_j(t)))\exp(j\omega_{A_1^i\cap\ldots\cap A_M^i})} \tag{10}$$

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

$$\begin{aligned}
\xi^i(t) &= \lambda^i(t)\times z^i(t) \\
&= \lambda^i(t)\times\left(a_0^i + \sum_{j=1}^{M} a_j^i h_j(t)\right) \\
&= \frac{(\overset{M}{\underset{j=1}{\wedge}} r_j^i(h_j(t)))\exp(j\omega_{A_1^i\cap\ldots\cap A_M^i})}{\sum_{i=1}^{K}(\overset{M}{\underset{j=1}{\wedge}} r_j^i(h_j(t)))\exp(j\omega_{A_1^i\cap\ldots\cap A_M^i})}\times\left(a_0^i + \sum_{j=1}^{M} a_j^i h_j(t)\right)
\end{aligned} \tag{11}$$

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.

$$\begin{aligned}
\xi(t) &= \sum_{i=1}^{K}\xi^i(t) = \sum_{i=1}^{K}\lambda^i(t)\times z^i(t) \\
&= \sum_{i=1}^{K}\frac{(\overset{M}{\underset{j=1}{\wedge}} r_j^i(h_j(t)))\exp(j\omega_{A_1^i\cap\ldots\cap A_M^i})}{\sum_{i=1}^{K}(\overset{M}{\underset{j=1}{\wedge}} r_j^i(h_j(t)))\exp(j\omega_{A_1^i\cap\ldots\cap A_M^i})}\times\left(a_0^i + \sum_{j=1}^{M} a_j^i h_j(t)\right)
\end{aligned} \tag{12}$$

Generally the output of the CNFS is represented as follows.

$$\begin{aligned}
\xi(t) &= \xi_{\mathrm{Re}}(t) + j\xi_{\mathrm{Im}}(t) \\
&= |\xi(t)|\times\exp(j\omega_\xi) \\
&= |\xi(t)|\times\cos(\omega_\xi) + j\,|\xi(t)|\times\sin(\omega_\xi)
\end{aligned} \tag{13}$$

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

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

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

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

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

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.

3 Hybrid PSO-RLSE Learning for CNFS

Particle swarm optimization (PSO) [10] 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. 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 compare 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.

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

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

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

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

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].

The RLSE [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.

$$y = \theta_1 f_1(u) + \theta_2 f_2(u) + \cdots + \theta_m f_m(u) \tag{20}$$

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 $\{\theta_i, i=1,2,\ldots,m\}$ for a unknown target system (or function), a set of input-output data pairs are used as training data, denoted as follows.

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

Substituting data pairs into (17), a set of N linear equations are given as follows.

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

The optimal estimation for $\boldsymbol{\theta}$ can be calculated using the following RLSE equations.

$$
\boldsymbol{P}_{k+1} = \boldsymbol{P}_k - \frac{\boldsymbol{P}_k \boldsymbol{b}_{k+1} \boldsymbol{b}_{k+1}{}^{\mathrm{T}} \boldsymbol{P}_k}{1 + \boldsymbol{b}_{k+1}{}^{\mathrm{T}} \boldsymbol{P}_k \boldsymbol{b}_{k+1}}, \tag{23a}
$$

$$
\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{P}_{k+1} \boldsymbol{b}_{k+1} (\boldsymbol{y}_{k+1} - \boldsymbol{b}_{k+1}{}^{\mathrm{T}} \boldsymbol{\theta}_k) \tag{23b}
$$

k=0,1,…,N-1, where $[\boldsymbol{b}^{\mathrm{T}}{}_{\boldsymbol{k}}, \boldsymbol{y}_k]$ in the k-th row of $[\boldsymbol{A}, \boldsymbol{y}]$. To start the RLSE algorithm in (20), we need to select the initial values for $\boldsymbol{\theta}_0$ and $\boldsymbol{P}_0$ is given as follows.

$$
\boldsymbol{P}_0 = \alpha \boldsymbol{I} \tag{24}
$$

where α is a large value and $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{\theta}_0$ is initially set to zeros.

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 (23), respectively. The PSO 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. The RLSE is used to update the consequent parameters, with the normalized firing strengths. With (20), to estimate the consequent parameters, the row vector $\boldsymbol{b}$ and the vector $\boldsymbol{\theta}$ are arranged as follows.

$$
\boldsymbol{b}_{k+1} = [\boldsymbol{bb}^1(k+1) \quad \boldsymbol{bb}^2(k+1) \quad \cdots \quad \boldsymbol{bb}^K(k+1)] \tag{25}
$$

$$
\boldsymbol{bb}^i(k+1) = [\lambda^i \quad h_1(k+1)\lambda^i \quad \cdots \quad h_M(k+1)\lambda^i] \tag{26}
$$

$$
\boldsymbol{\theta} = [\tau^1 \quad \tau^2 \quad \cdots \quad \tau^K] \tag{27}
$$

$$
\boldsymbol{\tau}^i = [a_0^i \quad a_1^i \quad \cdots \quad a_M^i] \tag{28}
$$

$i = 1,2,...,K$, and k=0,1,…,N-1,At each iteration for the hybrid PSO-RLSE learning, the output of the CNFS approximator can be obtained in (13). The error between output of the target and the CNFS is defined as follows.

$$
\mathrm{RMSE} = \left(\frac{1}{N} \sum_{t=1}^{N} (y(t) - \xi(t))^2 \right)^{\frac{1}{2}} = \left(\frac{1}{N} \sum_{t=1}^{N} (y(t) - F(H(t),W))^2 \right)^{\frac{1}{2}} \tag{29}
$$

The error is used further to define the root mean square error (RMSE), which is used as the performance index in the study. The square of RMSE is called the mean square error (MSE).

4 Experiments for the Proposed Approach

Experiments for function approximation are conducted in this section to estimate and verify the performance of the proposed approach. Two subsections are given in the section. In the first subsection, the proposed approach using the CNFS approximator and the hybrid PSO-RLSE learning method is compared to two other compared approaches. The first compared approach uses a traditional NFS approximator and the PSO learning method, and the second compared approach uses the same CNFS approximator and the PSO method alone. The tooth function is used in the 1st subsection. In the 2nd subsection, the proposed approach is compared to the approach in [12]. Two benchmark functions are used in the 2nd subsection for performance comparison.

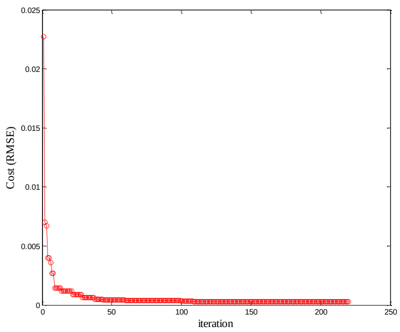
4.1 Comparison for the Proposed Approach to the PSO for CNFS, and the PSO for NFS

The "tooth" function is given as follows.

$$\begin{aligned} y &= 0.08\times\{1.2\times[(u-1)\times(\cos(3u))] \\ &\quad +[(u-(u-1)\times(\cos(3u)))\times\sin(u)]\}) \\ &\quad 3\le u\le 7 \end{aligned} \tag{30}$$

The generated 100 data pairs from the tooth function are used to train the CNFS which is designed with complex fuzzy sets and the NFS which is designed with traditional fuzzy sets. Two inputs are used to the CNFS and the traditional NFS. Each input possesses three fuzzy sets. There are 9 rules in CNFS and the NFS, where 12 premise parameters and 27 consequent parameters are to be updated by the PSO algorithm. In the hybrid PSO-RLSE for the CNFS, PSO is used to adjust the 12 premise parameter of CNFS and the RLSE is to update the 27 consequent parameters. For the output of the CNFS, we select the real part to represent the approximator output. For the PSO settings for the proposed CNFS and the NFS, $\{c_1, c_2\} = \{2, 2\}$ and the population size = 635 are given. And, for the hybrid PSO-RLSE settings, $\{c_1, c_2\} = \{2, 2\}$, $\alpha = 10^4$, and $\boldsymbol{\theta}_0$ = zero-valued vector are given.

The learning curve and the result by the proposed PSO-RLSE for the CNFS are given in Figs. 1 to 2. The performance comparison for the proposed approach and the two compared approach is given in Table 1. The approximation errors by the three approaches are shown in Fig. 3.



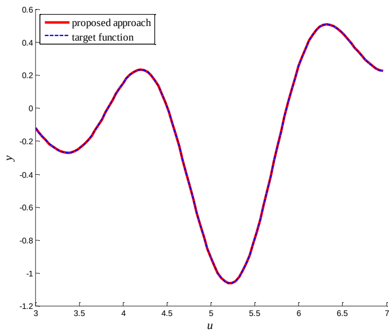


Fig. 1. Learning curve by the proposed hybrid PSO-RLSE for the CNFS for the tooth function.

Fig. 2. Response by the proposed CNFS with PSO-RLSE for the tooth function.

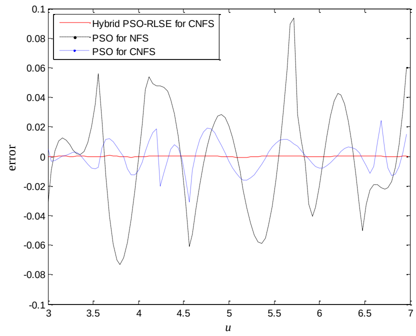


Fig. 3. Approximation errors by the PSO for NFS, the PSO for CNFS, and the hybrid PSO-RLSE for CNFS

Table 1. Performance Comparison.

Methods	**RLSE**
PSO for NFS	7.86×10^{-3}
PSO for CNFS	1.56×10^{-3}
Hybrid PSO-RLSE for CNFS	2.73×10^{-4}

4.2 Comparison for the Proposed Approach to the Approach in [12]

The hybrid PSO-RLSE method for the proposed CNFS is employed to approximate the two benchmark functions which are the exponential function in [-4, 2] and the hyperbolic tangent function in [-5, 5]. For each of the benchmark functions, the error norm for function approximation is based on the mean square errors (MSE), as defined before. The training and testing data for each benchmark function are 400 and 200 sampled pairs. With the 2 benchmark function, Table 2 shows the performance comparisons in MSE for 20 experimental trials using the proposed approach and the compared approach [12]. The approximation responses and errors for the 2 functions by the proposed approach are shown in Figs 4 to 7.

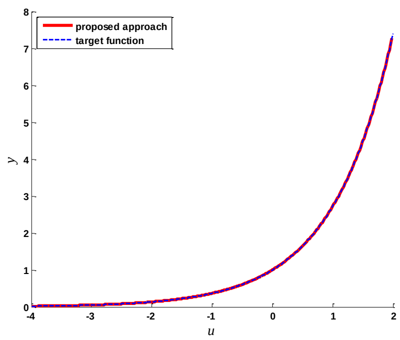


Fig. 4. Result by the proposed CNFS hybrid learning approach for the exponential function.

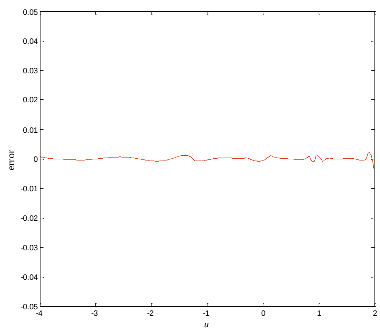


Fig. 5. Approximation error by the proposed CNFS hybrid learning approach for the exponential function.

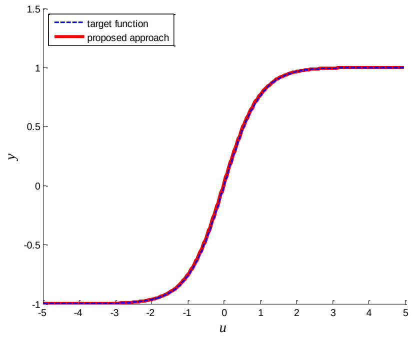


Fig. 6. Result by the proposed CNFS hybrid learning approach for the hyperbolic tangent function.

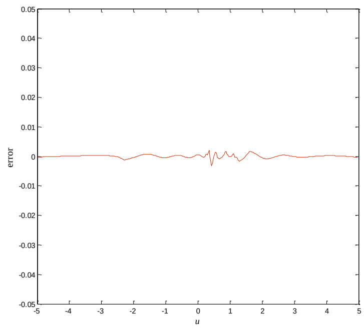


Fig. 7. Approximation error by proposed CNFS hybrid learning approach for the hyperbolic tangent function.

5 Discussion and Conclusion

The proposed complex neuro-fuzzy system (CNFS) has been presented 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 theproposed 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.

Table 2. Performance Comparison.

Function / **Method**	exp(u)	
	Mean±std (training)	Mean±std (testing)
Compared approach [12]	8.04×10^{-2} ±0	9.89×10^{-2} ±0
Proposed approach	1.2×10^{-10} $\pm1.04\times10^{-10}$	4.50×10^{-4} $\pm3.80\times10^{-4}$
Function / **Method**	tanh(u)	
	Mean±std (training)	Mean±std (testing)
Compared approach [12]	8.27×10^{-2} ±0	9.38×10^{-2} ±0
Proposed approach	3.86×10^{-8} $\pm1.69\times10^{-8}$	3.58×10^{-5} $\pm1.77\times10^{-5}$

It is found the proposed hybrid learning approach is superior to the two other compared methods, which are the traditional NFS using the PSO and the CNFS using the PSO. The performance comparison for the three approaches is shown in Table 1. The proposed hybrid learning approach with RMSE=2.73×10^{-4} is much better the traditional NFS with PSO learning method with RMSE=7.86×10^{-3}. The proposed approach has also been compared to other research approaches [12]. The performance comparisons are shown in Table 2, in which two benchmark functions 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 merit of the hybrid learning has been observed. Through the comparison experiments, the proposed approach has shown excellent performance.

References

1. Juang, C.F., Lin, C.T.: An online self-constructing neural fuzzy inference network and its applications. IEEE Transactions on Fuzzy Systems, vol. 6, pp. 12-32, (1998)
2. Paul, S., Kumar, S.: Subsethood-product fuzzy neural inference system. IEEE Transactions on Neural Networks, vol. 13, pp. 578-599 (2002)
3. Hornik, K., Stinchcombe, M., White, H.: Multilayer feed forward networks are universal approximators. Neural networks, vol. 2, pp. 359-366 (1989)
4. Wang, L.X., Mendel, J.M.: Fuzzy basis functions, universal approximation, and orthogonalleast-squares learning. IEEE Transactions on Neural Networks, vol. 3, pp. 807-814 (1992)
5. Kandel, A., Ramot, D., Milo, R., Friedman, M.: Complex Fuzzy Sets. IEEE Transactions on Fuzzy Systems, vol. 10, pp. 171–186 (2002)
6. Dick, S.: Toward complex fuzzy logic. IEEE Transactions on Fuzzy Systems, vol. 13, pp. 405-414 (2005)
7. Ramot, D., Friedman, M., Langholz, G., Kandel, A.: Complex fuzzy logic. IEEE Transactions on Fuzzy Systems, vol. 11, pp. 450-461 (2003)
8. Farag, W.A., Quintana, V.H., Lambert-Torres, G.: A genetic-based neuro-fuzzy approach for modeling and control of dynamical systems. IEEE Transactions on Neural Networks, vol. 9, pp. 756-767 (1998)
9. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE transactions on systems, man, and cybernetics, vol. 15, pp. 116-132 (1985)
10. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks (1995)
11. Hsia, T.C.: System identification: Least-squares methods. D. C. Heath and Company (1977)
12. Wu, J.M., Lin, Z.H., Hus, P.H.: Function approximation using generalized adalines. IEEE Transactions on Neural Networks, vol. 17, pp. 541-558 (2006)