Fifth IFSA World Congress (1993), 1177-1180

Nonlinear Function Approximation by Fuzzy-Neural Interpolating Networks

Il Hong Suh *and* Tae Won Kim

Dept. of Electronics Eng., Hanyang Univ.,
Seoul 133-791, KOREA
and
Engr. Reasrch Center for Adv. Control and Instr. (of SNU)
by Korea Science and Eng. Foundation(KOSEF)

Abstract

In this paper, a fuzzy-neural interpolating network is proposed to efficiently approximate a nonlinear function. Specifically, basis functions are first constructed by Fuzzy Membership Function based Neural Networks (FMFNN). And the fuzzy similarity, which is defined as the degree of matching between actual output value and the output of each basis function, is employed to determine initial weighting of the proposed network. Then the weightings are updated in such a way that square of the error is minimized. To show the capability of function approximation of the proposed fuzzy-neural interpolating network, a numerical example is illustrated.

1. Introduction

It is well known that the nonlinear function approximation can be often solved by finding a set of coefficients for a finite number of fixed nonlinear basis functions[1]. The Radial Basis Function (RBF) network can offer approximation capabilities similar to those of the two-layer neural network, provided that the hidden layer of the RBF network is fixed appropriately[2]. However, the performance of the RBF network critically depends upon the chosen centers of the basis functions[3]. To overcome such difficulties in choosing centers of RBF, the FMF network was proposed in [4,5,6], where it was shown that the structure of the RBF networks could be similar to that of fuzzy logics when employing the additive combination technique for the inference[7] and the FMF could play a role of a basis function. And an simple interpolating network was proposed to reduce difficulties in determining fuzzy rules and membership functions when a large number of input variable are necessary for the function approximation[4]. However, since this interpolating network has only one input node to learn function values, might be difficult to accurately approximate a complex function .

In this paper, a fuzzy-neural interpolating network is proposed to efficiently approximate a nonlinear function. Specifically, basis functions are first constructed by Fuzzy Membership Function based Neural Networks (FMFNN). And the fuzzy similarity, which is defined as the degree of matching between actual output value and the output of each basis function, is employed to determine how much each FMF network should contribute to the approximation of a function. A fuzzy-neural interpolating network is then proposed by combining the FMF network and the fuzzy similarity.

2. Fuzzy Membership Function based Neural Network

Consider the following fuzzy relations :

$$R^i : \text{If } y_1 \text{ is } A_{i1}, y_2 \text{ is } A_{i2}, \cdots, \text{ and } y_n \text{ is } A_{ip}, \text{ then } u \text{ is } B_i. \quad i = 1,2,\cdots,q. \tag{1}$$

Here, y_i, for $i = 1,2,\cdots,p$, is the input variable and u is the output fuzzy variable fuzzified with a singleton membership function. A_{ij} and B_i, for $i = 1,2,\cdots,q$ and $j = 1,2,\cdots,p$, are input and output linguistic (fuzzy-set) values, respectively. And let $\mu^A_{ij}(y_j)$ and $\mu^B_i(u)$ be the membership functions for A_{ij} and B_i, respectively. If we let $\mu^B_i(u)$ be a normal singleton located at $u = \lambda_i$ for each i, and apply the centroidal defuzzification technique to $\mu^B(u)$, then $\mu^B_i(u)$ becomes $\mu^B_i(\lambda_i)$. And thus, regardless of types of inference, the scalar output u can be obtained by

$$u = \sum_{i=1}^{q} \lambda_i \frac{\Phi_i(y_1^0, y_2^0, \cdots, y_p^0)}{\sum_{k=1}^{q} \Phi_k(y_1^0, y_2^0, \cdots, y_p^0)} = \sum_{i=1}^{q} \lambda_i \tilde{\Phi}_i(\underline{y}^0), \tag{2}$$

where $\Phi_i(y_1^0, y_2^0, \cdots, y_p^0)$, $\underline{y}^0$ and $\tilde{\Phi}_i(\underline{y}^0)$ are defined as

$$\Phi_i(y_1^0, y_2^0, \cdots, y_p^0) \triangleq \min\{\mu^A_{ij}(y_j^0) \mid j = 1,2,\cdots,p\}, \tag{3-1}$$

$$\underline{y}^0 \triangleq (y_1^0, y_2^0, \cdots, y_p^0), \tag{3-2}$$

and

$$\tilde{\Phi}_i(\underline{y}^0) \triangleq \frac{\Phi_i(y_1^0, y_2^0, \cdots, y_p^0)}{\sum_{k=1}^{q} \Phi_k(y_1^0, y_2^0, \cdots, y_p^0)}. \tag{3-3}$$

Eq.1 can play a role of approximating a function as the RBF network can do[4,5,6]. Thus we called Eq.1 as "*Fuzzy Membership Function* (*FMF*) *based Neural Network*", where

λ_i 's are the neural weights to be trained by using the linear least square method. In applying Eq.1 to a function approximation, as in [8], we put a nonlinear scalar function $g : R \rightarrow R$ given by

$$g(u) = (1 - \exp(-\beta_1 u)) / (1 + \exp(-\beta_1 u)) \quad (4)$$

at the output node together with the scaling factor K to effectively account for the maximum magnitude of the function output. In Eq.4, β_1 is a constant implying the slope of output node function. For the function approximation, let $f(\underline{y})$ be a scalar function to be approximated, and let $u(\underline{y})$ be an approximation of $f(\underline{y})$. Then $u(\underline{y})$ can be represented as

$$u(\underline{y}) = Kg(\sum_{i=1}^{q} \lambda_i \tilde{\Phi}_i(\underline{y})), \quad (5)$$

if the FMF network is utilized for the function approximation.

When the error function is given by

$$E = \frac{1}{2}(f(\underline{y}) - u(\underline{y}))^2, \quad (6)$$

the weight changes $\Delta\lambda_i$ could be chosen to be proportional to $-\partial E/\partial\lambda_i$, i.e.,

$$\Delta\lambda_i = \eta_1\beta_1 K[f(\underline{y}) - u(\underline{y})]\{1 - g^2[\sum_{j=1}^{q}\lambda_j\tilde{\Phi}_j(\underline{y})]\}\tilde{\Phi}_i(\underline{y})/2, \quad (7)$$

where η_1 is learning-rate parameter. Thus the learning rule for adapting weight can be given as

$$\lambda_i^{t+1} = \lambda_i^t + \eta_1 \beta_1 K[f(\underline{y}) - u^t(\underline{y})]\{1 - g^2[\sum_{j=1}^{q}\lambda_j\tilde{\Phi}_j(\underline{y})]\}\tilde{\Phi}_i(\underline{y})/2 \quad (8)$$

where t is an integer implying the number of learning trials. The schematic diagram of our FMF network with p inputs and a scalar output is depicted in Fig.1.

It is remarked that the initial weight values are known to play important roles of obtaining the global minimum in most of neural networks[9]. In this respect, heuristic choice of the initial values considering the fuzzy rules for the FMF networks can provide better performances than the random choice of the initial weight values for the RBF network, since the weights of the FMF network have clear meanings as locations of the singleton fuzzy membership functions for the 'THEN' parts of the fuzzy rules, but the weights of the RBF networks have no physical meanings. It is also remarked that in some real applications, the number of membership functions and their centers seem to be well-selected by carefully observing data structures rather than the arbitrary selection mechanism usually employed in the RBF network.

It is further remarked that Wand and Mendal [10] recently proposed the Fuzzy Basis Function (FBF) for function approximation, which is similar to our FMF network. However, in their approach, there were no *learning* trials, which implies that FBF was not employed as the type of a neural network. Specifically, two arbitrary sets of initial FMF were first constructed by using input-output data pairs and linguistic IF-THEN rules. And then significant FBF's among initial FBF's were selected based on their error reduction ratio. Thus, the performance of their algorithm might be critically determined by initial choices of basis functions, their membership functions, and the number of basis functions to be finally fixed, which may require many a trial and error to achieve satisfactory performances. Compared to their approach, in our FMF network, membership functions of THEN part in fuzzy IF-THEN rules are adjusted by the gradient descent method as usual in the RBF and the Back Propogation Neural Network. And there is no need to determine many a initial basis functions as well as the number of basis function to be finally fixed, owing to the learning capability of our FMF network. Especially, when a large number of fuzzy variables are necessary for the function approximation, the proposed fuzzy-neural interpolating network can be employed to make fuzzy rules be simple, but there are no such a scheme in [10]. In such view points, our approach can be considered as a better solution of fuzzy-neural fusion rather than the approach in [10].

3. A Fuzzy-Neural Interpolating Network

When comparing the FMF network with the RBF network, we could observe that the FMF network might be considered as an effective fusion of the RBF neural network and fuzzy reasoning technique, since the network effectively reflect human expert's experiences and has a good learning capability[4,5,6]. However, one might have difficulties in determining fuzzy rules and membership functions, when a large number of input variables are necessary for the function approximation.

To cope with such difficulties, a fuzzy-neural interpolating network is here proposed. To be specific, let $f(y_1,y_2,\cdots,y_p)$ be a function to be approximated and be represented by M representative functions $H_i(y_1,y_2,\cdots,y_{p-1}) \triangleq f(y_1,y_2,\cdots,y_{p-1},y^*_{p,i})$, for $i = 1, 2,\cdots,M$, where $H_i(y_1,y_2,\cdots,y_{p-1})$ can be obtained by assigning a constant value $y^*_{p,i}$ to a variable y_p of the function $f(y_1,y_2,\cdots,y_p)$. Let $H_i(y_1,y_2,\cdots, y_{p-1})$, for i=1,2,$\cdots$,M, be approximated as $\tilde{H}_i(y_1,y_2,\cdots,y_{p-1})$ by utilizing M FMF networks. Then for a given input $(y_1,y_2,\cdots,y_p)$, if $\hat{y}_p$ is different from $y^*_{p,i}$ for all i, the output value of $f(y_1,y_2,\cdots,y_p)$ needs to be estimated by interpolating $\tilde{H}_i(y_1,y_2,\cdots,y_{p-1})$, for i=1,2,$\cdots$,M. For this, we employ fuzzy rules which inform how much $\hat{y}_p$ is similar to each $y^*_{p,i}$, i=1,2,$\cdots$,M. To be specific, let S_i be the *fuzzy similarity* between $\hat{y}_p$ and $y^*_{p,i}$, and n_s be the number of fuzzy rules for S_i, and let $\pi_{ij} : R \rightarrow [0,1]$ for $i = 1,2,\cdots, M$, and $j = 1,2,\cdots,n_s$ be the membership function for the jth linguistic value of $\left| \hat{y}_p - y^*_{p,i} \right|$. Also let γ_{ij} be the location of singleton membership function for the jth linguistic value of S_i. Then S_i can be represented as

$$S_i = \sum_{j=1}^{n_s} \gamma_{ij}\, \pi_{ij}(\hat{y}_p). \quad (9)$$

Now for a fuzzy-neural interpolation, the sigmoid function given by

$$\hat{g}(x) \triangleq 1 / (1 + \exp(\beta_2 x)) \quad (10)$$

is used as an output node function of each S_i. Since the larger $\hat{g}(S_i)$ is, the more $H_i(y_1, y_2,\cdots,y_{p-1})$ are contributed to finding values of $f(\underline{y})$, it may be reasonable that $f(\underline{y})$ can be found by

$$f(\underline{y}) = Kg[\sum_{i=1}^{M}\hat{g}(S_i)\, \tilde{H}_i(y_1,y_2,\cdots,y_{p-1})], \qquad (11)$$

where $g(\bullet)$ and $\hat{g}(\bullet)$, respectively, are the scalar functions defined as in Eq.4 and Eq.10. The schematic diagram of the fuzzy-neural interpolating network is depicted in Fig.2. It is remarked that since n_s in Eq.9 was given as unity in [4], the result of function approximation was not satisfactory. This implies that there were no rooms for learning other function values except only a pre-learned datum.

Note that since only M representative functions are available, at most M function values can be approximated. Thus to cover the whole input space, we need to divide the input space by M subspaces. By choosing the performance index J_i for the i-th subspace B_i as

$$J_i \triangleq \sum_{\underline{y}\in B_i} (u(\underline{y}) - f(\underline{y}))^2, \qquad (12)$$

and by applying the gradient descent method to minimize J_i, our updating rule for the weight γ_{ij} in Eq.9 can be obtained. Specifically, since $\hat{g}'(u) = \beta_2\, \hat{g}(u)(1-\hat{g}(u))$, the derivative of the performance index J_i with respect to the weight γ_{ij} can be obtained as follows;

$$\begin{aligned}\frac{\partial J_i}{\partial \gamma_{ij}} &= 2\,[\sum_{\underline{y}\in B_i} (u(\underline{y}) - f(\underline{y}))]\,(u(\underline{y}) - f(\underline{y}))' \\ &= 2\,[\sum_{\underline{y}\in B_i} (u(\underline{y}) - f(\underline{y}))]\,(-K\beta_1/2) \\ &\cdot \{\, 1 - g^2[\sum_{k=1}^{M}\hat{g}(\sum_{l=1}^{m}\gamma_{kl}\pi_{kl}(\hat{y}_p))\tilde{H}_i(y_1,y_2,\cdots,y_{p-1})]\} \\ &\cdot \{\, \beta_2\, \hat{g}(\gamma_{ij}\pi_{ij}(\hat{y}_p))\,[1-\hat{g}(\gamma_{ij}\pi_{ij}(\hat{y}_p))] \\ &\cdot \pi_{ij}(\hat{y}_p)\, \tilde{H}_i(y_1,y_2,\cdots,y_{p-1}) \end{aligned} \qquad (13)$$

Thus, the learning rule for updating weights γ_{ij} of the fuzzy-neural interpolating network can be given as

$$\begin{aligned}\gamma_{ij}^{t+1} &= \gamma_{ij}^{t} + K\beta_1\beta_2\eta_2\pi_{ij}(\hat{y}_p)\tilde{H}_i(y_1,y_2,\cdots,y_{p-1})\,) \\ &\cdot\{1 - g^2[\sum_{k=1}^{M}\hat{g}(\sum_{l=1}^{m}\gamma_{kl}^{t}\pi_{kl}(\hat{y}_p))\tilde{H}_i(y_1,y_2,\cdots,y_{p-1})]\} \\ &\cdot\, \hat{g}(\gamma_{ij}^{t}\pi_{ij}(\hat{y}_p))[1-\hat{g}(\gamma_{ij}^{t}\pi_{ij}(\hat{y}_p))]\,[\sum_{\underline{y}\in B_i}(u(\underline{y}) - f(\underline{y}))], \end{aligned} \qquad (14)$$

where η_2 is the learning-rate parameter given as a positive constant not greater than or equal to unity.

4. A Numerical Example

To show the capability of the function approximation of the proposed fuzzy-neural interpolating network, a simulation is performed with a function known as the Mexican hat function given by

$$f(x,y) = \begin{cases} \dfrac{40\sin(\pi\sqrt{x^2+y^2}/35)}{\sqrt{x^2+y^2}/35}, & \text{for } x \neq 0, y \neq 0, \\ 40\pi, & \text{for } x = y = 0, \end{cases} \qquad (15)$$

which is depicted in Fig.3. The input and output universes of discourse are given as $x \in$ [-120, 120], $y \in$ [-120,120] and $f(x,y) \in$ [-27.298,125.6637]. Assume that 169 input-output relations are available and these are divided into 13 subspaces. Then 13 FMF networks are assigned in such a way that each FMF network learns input-output mapping. The fuzzy rules for designing the i-th FMF network can be generated by observing the training data. It is remarked that the whole input space is divided into 13 subspaces and 13 fuzzy rules are required for each subspace.

After completely training 13 FMF networks, the proposed fuzzy-neural interpolating network with the learning rule in Eq.14 is applied to improve the degree of the function approximation. To verify the capability of approximation of our FMF networks incorporating the fuzzy-neural interpolating network, the approximated function is retrieved with 49×49 segmented input data. It may be observed from Fig.4 that our FMF networks incorporating the fuzzy-neural interpolating network can be used as a function approximation, but the network could not be completely reproduce the given function due to insufficient training data. In general, if the number of subspace is sufficiently large, the given function is expected to be satisfactorily approximated by the proposed FMF networks incorporating the fuzzy-neural interpolating network.

5. Concluding Remarks

It was shown that a fuzzy-neural interpolating network could be designed by employing both the fuzzy similarity and the FMF as a basis function. Simulation results showed that the performance of function approximation was satisfactory by incorporating the FMF network with the fuzzy-neural interpolating network. It is remarked that since the structure of the proposed networks could effectively reflect the expert's knowledge, the proposed network is expected to show several desirable performances such as training simplicity, fast convergency, design simplicity, fast learning speed, and no computational complexity when retrieving.

References

1. T. D. Sanger,"A Tree-Structured Adaptive Network for Function Approximation in High-Dimensional Spaces," *IEEE Trans. on Neural Network*, **vol.2**, no.2, March 1991, pp.285-293.
2. M.J.D.Powell, "Radial basis function approximations to polynomials," *in Proc. 12th Biennial Numerical Analysis Conf.* (Dundee), 1987, pp.223-241.
3. S.Chen, C.F.N. Cowan and P.M. Grant, "Orthogonal least square learning algorithm for radial basis function networks," *IEEE Trans. on Neural Networks*, **vol.2**, no.2, March. 1991, pp.302-309.
4. I.H.Suh and T.W.Kim, "Visual Servoing of Robot Manipulators by Fuzzy Membership Function Based Neural Networks," to appear in *Visual Servoing - Automatic Control of Mechanical Systems with Visual Sensors*, K.Hashimoto, Ed., World Scientific Publishing Co.
5. I.H.Suh and T.W.Kim, "Nonlinear Function Approximation by the Fuzzy Membership Function based Neural Networks," in *Proc. of ConFuSE '92* (Seoul, Korea), Oct. 1992, pp.153-156

6. I.H.Suh and T.W.Kim, "Fuzzy Membership Function Based Neural Networks with Applications to the Visual Servoing of Robot Manipulators", submitted to *IEEE Trans. on Neural Networks*
7. W. Pedrycz, *Fuzzy control and Fuzzy systems*, Research Studies Press LTD., 1989.
8. T.Fukuda and T.Shibata, "Theory and Applications of Neural Networks for Industrial Control Systems," *IEEE Trans. on Industrial Electronics*, **vol.39**, no.6, Dec.1992, pp.472-489
9. D.E.Rumelhart, G.E.Hinton and R.J.Williams: "Learning internal representations by error propagation," in *Parallel Distributed Processing*. Cambridge, MA:MIT Press, 1986, **Ch.8**, pp318-36.
10. L.X.Wand and J.M.Mendel, "Fuzzy Basis Function, Universal Approximation, and Orthogonal Least-Squares Learning," *IEEE Trans. on Neural Networks*, **vol.3**, no.5, Sep. 1992, pp.807-814

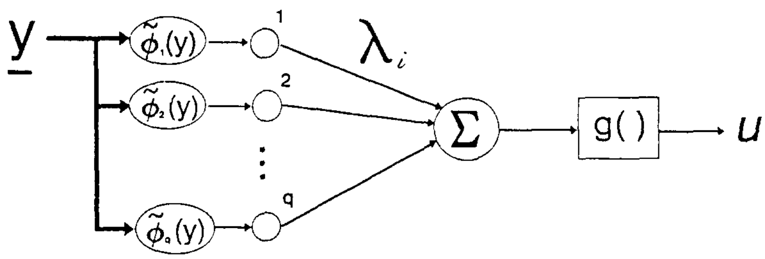

Fig.1. Schematic diagram of an FMF based neural network

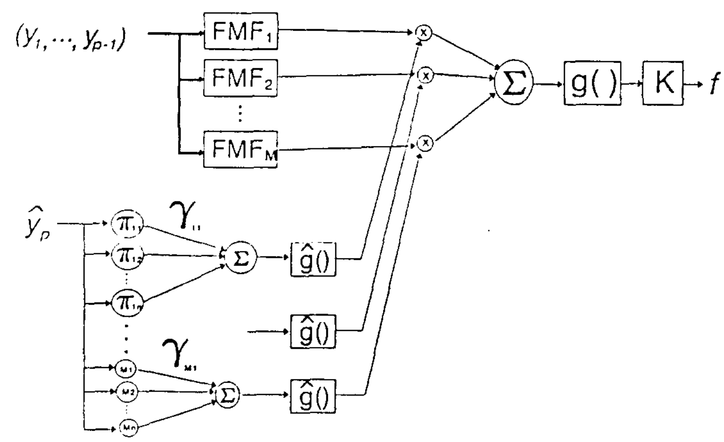

Fig.2. Schematic diagram of the fuzzy-neural interpolating network

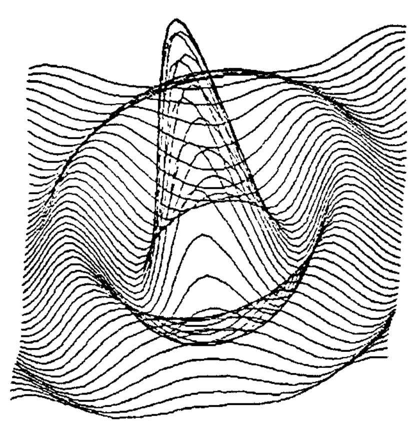

Fig.3.Graphical representation of the function

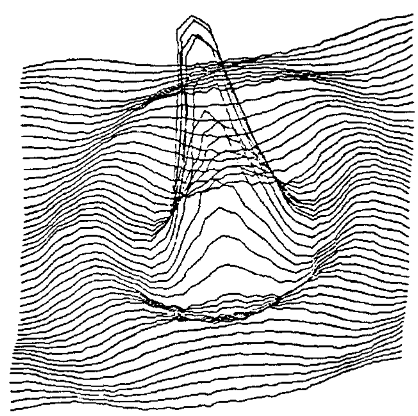

Fig.4. Graphical representation of the function in Eq.15 approximated by 13 FMF networks incorporating the fuzzy-neural interpolating network