NOTES

Output-back fuzzy logic systems and equivalence with feedback neural networks

LI Hongxing

Department of Mathematics, Beijing Normal University, Beijing 100875, China

Abstract A new idea, output-back fuzzy logic systems, is proposed. It is proved that output-back fuzzy logic systems must be equivalent to feedback neural networks. After the notion of generalized fuzzy logic systems is defined, which contains at least a typical fuzzy logic system and an output-back fuzzy logic system, one important conclusion is drawn that generalized fuzzy logic systems are almost equivalent to neural networks.

Keywords: output-back fuzzy logic systems, generalized fuzzy logic systems, feedforward neural networks, feedback neural networks.

Recently there was an increasing interest in combining fuzzy logic systems with neural networks in both application and research. This has led to the publication of several books on fuzzy-neural or neural-fuzzy systems in the past three or four years[1-3]. However, the relationships between fuzzy logic systems and neural networks are not well-studied. We have proved the equivalence of fuzzy logic systems and feedforward neural networks before[4], in this note, we propose an interesting fuzzy logic system, which is named an output-back fuzzy logic system. Then we will prove that the output-back fuzzy logic system must be equivalent to the feedback neural network.

1 Output-back fuzzy logic systems

(i) Notations and basic concepts on fuzzy logic systems. We described the fuzzy logic system in a more formal way[4], here we will introduce the fuzzy logic system in a general way. Let $X_1, X_2, \cdots, X_n$ be the universes of the input variables $x^{(1)}, x^{(2)}, \cdots, x^{(n)}$, respectively, and $Y_1, Y_2, \cdots, Y_m$ be the universes of the output variables $y^{(1)}, y^{(2)}, \cdots, y^{(m)}$, and let $\mathscr{A}_i = \{A_{p_i}^{(i)}\}_{(1 \leqslant p_i \leqslant k_i)}$ $(i = 1, 2, \cdots, n)$ and $\mathscr{B}_i = \{B_{p_1 p_2 \cdots p_n}^{(j)}\}_{(1 \leqslant p_1 \leqslant k_1, 1 \leqslant p_2 \leqslant k_2, \cdots, 1 \leqslant p_n \leqslant k_n)}$ $(j = 1, 2, \cdots, m)$, where $A_{p_i}^{(i)} \in \mathscr{F}(X_i)(i = 1, 2, \cdots, n)$ and $B_{p_1 p_2 \cdots p_n}^{(j)} \in \mathscr{F}(Y_j)(j = 1, 2, \cdots, m)$. $\mathscr{A}_i$ and $\mathscr{B}_j$ can be regarded as linguistic variables so that a group of fuzzy inference rules is formed as follows:

If $x^{(1)}$ is $A^{(1)}_{p_1}$ and $x^{(2)}$ is $A^{(2)}_{p_2}$ and $\cdots$ and $x^{(n)}$ is $A^{(n)}_{p_n}$, then $y^{(j)}$ is $B^{(j)}_{p_1 p_2 \cdots p_n}$ (1)

where p_i=1,2,⋯,k_i (i = 1,2,⋯, n) and j = 1,2,⋯, m; $x^{(i)} \in x_i$ and $y^{(j)} \in y_j$ are called base variables. the (p_1, p_2,⋯, p_n)-th inference rule is a fuzzy relation from $\prod_{i=1}^{n} X_i$ to Y_j, $R^{(j)}_{p_1 p_2 \cdots p_n} \triangleq \left(\prod_{i=1}^{n} A^{(i)}_{p_i}\right) \times B^{(j)}_{p_1 p_2 \cdots p_n}$, where

$$R^{(j)}_{p_1 p_2 \cdots p_n}((x^{(1)}, x^{(2)}, \cdots x^{(n)}), y^{(j)}) = \left(\bigwedge_{i=1}^{n} A^{(i)}_{p_i}(x^{(i)})\right) \wedge B^{(j)}_{p_1 p_2 \cdots p_n}(y^{(j)}).$$

Since $p_1 \times p_2 \times \cdots \times p_n$ inference rules should be joined by "or" (corresponding to set-theoretical operator "$\cup$"), the whole inference relation, with respect to the jth component index, is $R^{(j)} = \bigcup_{i=1}^{n} \bigcup_{p_i=1}^{k_i} R^{(j)}_{p_1 p_2 \cdots p_n}$, i.e.

$$\begin{aligned} &R^{(j)}((x^{(1)}, x^{(2)}, \cdots x^{(n)}), y^{(j)}) \\ &= \bigvee_{i=1}^{n} \bigvee_{p_i=1}^{k_i} \left[\left(A^{(i)}_{p_1}(x^{(1)})\right) \wedge A^{(i)}_{p2}(x^{(2)} \wedge \cdots \wedge A^{(i)}_{p_n}(x^{(n)})) \wedge B^{(i)}_{p_1 p_2 \cdots p_n}(y^{(j)})\right]. \end{aligned} \quad (2)$$

Given $A^{(i)'}_{p_i} \in \mathscr{F}(X_i)(i = 1,2,\cdots,n)$, by the CRI method[5−8], we can determine the inference conclusion $B^{(j)'} \in \mathscr{F}(Y_j)$ with respect to the ith component as $B^{(j)'} \triangleq \left(\prod_{i=1}^{n} A^{(j)'}_{p_i}\right) \circ R^{(j)}$, where

$$B^{(j)'}(y^{(j)}) = \bigvee_{(x^{(1)}, \cdots x^{(n)}) \in \prod_{i=1}^{n} X_i} \left[\left(\bigwedge_{i=1}^{n} A^{(i)'}_{p_i}(x^{(i)})\right) \wedge R^{(j)}(x^{(1)}, x^{(2)}, \cdots, x^{(n)})\right]. \quad (3)$$

In a fuzzy logic system, the inputs must first be fuzzified before using eq. (3), because the inputs are crisp quantities. For a given input $(x^{(1)'}, x^{(2)'}, \cdots, x^{(n)'}) \in \prod_{i=1}^{n} X_i$, the fuzzification is defined as

$$A^{(i)'}_{p_i}(x^{(i)}) = \begin{cases} 1, & x^{(i)} = x^{(i)'}, \\ 0, & x(i) \neq x^{(i)'}. \end{cases}$$

Thus, for every j, we can get an inference conclusion about the input $(x^{(1)'}, x^{(2)'}, \cdots, x^{(n)'}), B^{(j)'}$, as follows:

$$\begin{aligned} B^{(j)'}(y^{(j)}) &= R^{(j)}((x^{(1)'}, x^{(2)'}, \cdots, x^{(n)'}), y^{(j)}) \\ &= \bigvee_{i=1}^{n} \bigvee_{p_i=1}^{k_i} \left[\left(A^{(1)}_{p_1}(x^{(1)'}) \wedge A^{(2)}_{p_2}(x^{(2)'}) \wedge \cdots \wedge A^{(n)}_{p_n}(x^{(n)'})\right) \wedge B^{(j)}_{p_1 p_2 \cdots p_n}(y^{(j)})\right]. \end{aligned} \quad (4)$$

Since $B^{(j)'}$ is a fuzzy set, it is necessary that $B^{(j)'}$ be converted to a crisp value. The process is referred as defuzzification. A common method used for this purpose is the method of centroid:

$$y^{(j)'} = \frac{\int_{y^{(j)} \in Y_j} y^{(j)} B^{(j)'}(y^{(j)}) \mathrm{d}y^{(j)}}{\int_{y^{(j)} \in Y_j} B^{(j)'}(y^{(j)}) \mathrm{d}y^{(j)}}. \quad (5)$$

This completes a simple description of fuzzy logic systems with n inputs and m outputs. Fuzzy logic controllers employ the same structure as fuzzy logic systems. A fuzzy logic system with n inputs and m outputs can be simply illustrated with fig. 1.

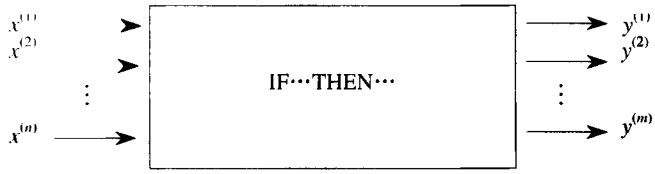

Fig. 1. Fuzzy logic systems with n inputs and m outputs.

The remainder of this section is to state some concepts or conditions to be used later. Let X be a given universe and $\mathscr{A}=\{A_i\}_{(1\leqslant i\leqslant k)}$ be a family of normal fuzzy sets on X, i.e. $(\forall i)(\exists x_i\in X)(A_i(x_i)=1)$. The x_i of the normal fuzzy set A_i is called the peak point of A_i. In this note, we assume that $\mathscr{A}$ has Kronecker's property, i.e. $A_i(x_j)=\delta_{ij}=1$ or 0 depends on $i=j$ or $i\neq j$. Normally $\mathscr{A}$ is a fuzzy partition of X, which means that for every i and j, $(i\neq j\Rightarrow x_i\neq x_j)$, and we get

$$(\forall x\in X)\left(\sum_{i=1}^{k} A_i(x)=1\right). \tag{6}$$

Eq. (6) clearly implies that $\mathscr{A}$ possesses Kronecker's property.

A fuzzy logic system is called a normal fuzzy logic system if every input linguistic variable of the fuzzy logic system forms a fuzzy partition[9,10].

Moreover, we call A_i a base element or the base function of $\mathscr{A}$. Thus we also call $\mathscr{A}$ a group of base elements of X. In this note, when $X=[a,b]\subset\mathbb{R}$ (the field of real numbers), we assume that $a<x_1<x_2<x_k<\cdots<b$.

(ii) Interpolation representation of fuzzy logic systems with n inputs and m outputs. Let $X_i=[a_i,b_i]$ and $Y_j=[c_j,d_j]$ be all real number intervals $(i=1,2,\cdots,n,j=1,2,\cdots,m)$, $\mathscr{A}_i$ be fuzzy partitions and $\left\{x_{p_i}^{(i)}\right\}_{(1\leqslant p_i\leqslant k_i)}$ and $\left\{y_{p_1p_2\cdots p_n}^{(j)}\right\}_{(1\leqslant p_1\leqslant k_1,1\leqslant p_2\leqslant k_2,\cdots,1\leqslant p_n\leqslant k_n)}$ be the peak points of $A_{p_i}^{(i)}$ and $B_{p_1p_2\cdots p_n}^{(i)}$ respectively, with $a_i<x_1^{(i)}<x_2^{(i)}<\cdots<x_{k_i}^{(i)}<b_i\,(i=1,2,\cdots,n)$.

Based on the conclusions in ref. [5], it is easy to get the following result[4]:

Lemma. Under the above conditions, a fuzzy logic system with n inputs and m outputs is approximately an n-ary piecewise interpolation vector value function just taking $A_{p_i}^{(i)}$ for its base functions (i.e. a piecewise interpolation mapping from $R^n\to R^m$):

$$\begin{aligned}Y&\triangleq(y^{(1)},y^{(2)},\cdots,y^{(m)})=F(X)\triangleq F(x^{(1)},x^{(2)},\cdots,x^{(n)})\\&=(F_1(X),F_2(X),\cdots,F_m(X)),\end{aligned} \tag{7}$$

where

$$\begin{aligned}F_j(X)&=F_j(x^{(1)},x^{(2)},\cdots x^{(n)})\\&=\sum_{p_1=1}^{k_1}\sum_{p_2=2}^{k_2}\cdots\sum_{p_n=1}^{k_n}\left(\prod_{i=1}^{n}A_{p_i}^{(j)}(x^{(i)}))y_{p_1p_2\cdots p_n}^{(j)}\right).\end{aligned} \tag{8}$$

This is the interpolation representation of a fuzzy logic system with n inputs and m outputs. So a fuzzy logic system can be simply reillustrated with fig. 2.

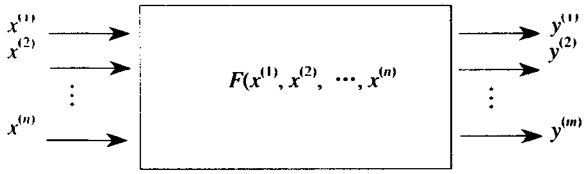

Fig. 2. Fuzzy logic systems with n inputs and m outputs.

(iii) Definition of output-back fuzzy logic systems.

Definition. A fuzzy logic system with n inputs and m outputs shown as fig. 2 is called an output-back fuzzy logic system with n channels, if $n = m$ and the outputs of the fuzzy logic system go back to its inputs shown as fig. 3, where $X(0) = (x^{(1)}(0), x^{(2)}(0), \cdots, x^{(n)}(0))$ is an initial value of the system.

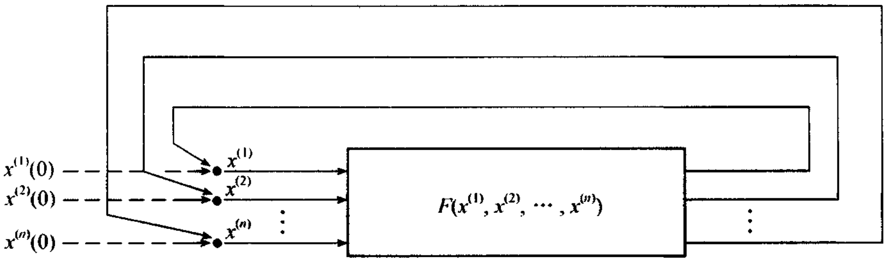

Fig. 3. Fuzzy logic systems with n inputs and m outputs.

2 The equivalence between output-back fuzzy logic systems and feedback neural networks

Theorem. Output-back fuzzy logic systems are equivalent to feedback neural networks.

Proof. Necessity: An arbitrarily given output-back fuzzy logic system with n channels (see fig. 3), if we cut off the links between its inputs and outputs, it will become a typical fuzzy logic system with n inputs and n outputs shown as fig. 4 based on the lemma.

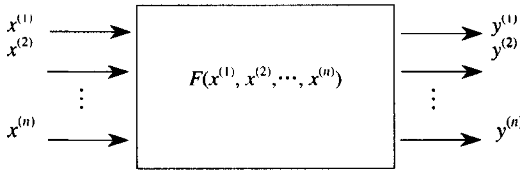

Fig. 4. A typical fuzzy logic sysem with n inputs and n outputs.

According to the conclusion[4], the typical fuzzy logic system with n inputs and n outputs must be equivalent to the feedforward neural network with n inputs and n outputs, which can be illustrated with fig. 5.

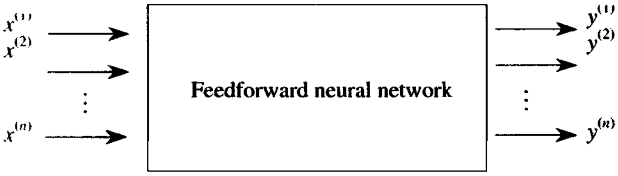

Fig. 5. A feedforward neural network with n inputs and n outputs.

Now we can rewrite the feedforward neural network into a feedback neural network shown as fig. 6 only by linking its inputs with its outputs.

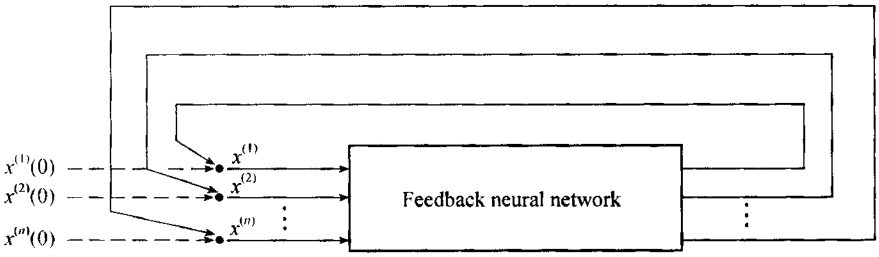

Fig. 6. A feedback neural network reformed from fig. 5.

Clearly, the feedback neural network is equivalent to the given output-back fuzzy logic system. This finishes the proof of Necessity.

Sufficiency: An arbitrarily given feedback neural network with n channels can be illustrated by fig. 6. If we cut off the links between its outputs and its inputs, the feedback neural network will become a feedforward neural network with n inputs and n outputs which was shown as fig. 5. Based on the result[4], the feedforward neural network must be equivalent to the fuzzy logic system with n inputs and n outputs illustrated as fig. 4. Then we reform the fuzzy logic system into an output-back fuzzy logic system with n channels.

Clearly, the output-back fuzzy logic system is equivalent to the given feedback neural network.

Q.E.D.

3 Conclusion

We first proposed an interesting concept, i.e. the output-back fuzzy logic system. This system is based on such an important concept that we are able to discuss and analyze the stability of the fuzzy logic system as well as to research the capacity and memory of the system.

According to the result[4], we proved that the output-back fuzzy logic system must be equivalent to the feedback neural network, which is the main result of this note. It is worth noticing that the word "equivalent" here means that, for an arbitrarily given output-back fuzzy logic system, it can be approximately represented as a feedback neural network, where the approximation can be reached within an arbitrarily given accuracy.

The generalized fuzzy logic system, which we have just defined, must contain a typical fuzzy logic system and an output-back fuzzy logic system at least.

It is well known that feedforward neural networks and feedback neural networks are two main parts of the neural networks. From the result[4] and the conclusion of the Theorem in this note, we can draw the following conclusion: generalized fuzzy logic systems are almost equivalent to neural networks, where "almost" means that we have not considered other kinds of neural networks such as random neural networks (for example, Boltzmann machines).

Acknowledgements This work was a part of the project supported by the National Natural Science Foundation of China (Grant No.69644014).

References

1. Brown, M., Harris, C., Neurofuzzy Adaptive Modelling and Control, New York: Prentice Hall, Inc. 1994.
2. Kosko, B., Neural Networks and Fuzzy Systems, Englewood Cliffs, New York: Prentice Hall, Inc., 1992.
3. Jiang, J. R., Sun, C., Mizutani, E., Neuro-Fuzzy and Soft Computing, Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1997.
4. Li Hongxing, Fuzzy logic systems are equivalen to feedforward neuralnetwork, Science in China, Ser. E (to appear).
5. Li Hongxing, Interpolation mechanism of the fuzzy control, Science in China, Ser. E, 1998, 41(3): 312.
6. Li Hongxing, The mathematical essence of fuzzy controls and fine fuzzy controllers, in Advances in Machines Intelligence and Soft-Computing (ed. Paul P. Wang), Vol. IV, Durham: Bookwright Press, 1997, 55—74.
7. Li Hongxing, Adaptive fuzzy controllers based on variable universe, Science in China, Ser. E, 1999, 42(1): 10.
8. Zhang Naiyao, Structural analyse of typical fuzzy controllers, Fuzzy Systems and Mathematics (in Chinese), 1997, 11(2): 10.
9. Wang Guojun, On the logic foundation of fuzzy reasoning, Lecture Notes in Fuzzy Mathematics and Computer Science, Omala: Greighton Uni., 1997, 4: 1.
10. Liu Xihui, Wang Haiyan, Network Fuzzy Analytical Methods (in Chinese), Beijing: Electronic Press, 1991, 1—14.

(Received June 3, 1999)