The 23rd International Technical Conference on Circuits/Systems,
Computers and Communications (ITC-CSCC 2008)

Fuzzy Inference Models Appropriate for Digital Circuit

Shinya Nagamine[1], Noritaka Shigei[2] and Hiromi Miyajima[3]
[1]Graduate School of Science and Engineering, Kagoshima University,
[2 3]Faculty of Engineering, Kagoshima University,
1-21-40 Korimoto, Kagoshima 890-0065, Japan
E-mail : [1]bt203089@ms.kagoshima-u.ac.jp, [2]shigei@eee.kagoshima-u.ac.jp,[3]miya@eee.kagoshima-u.ac.jp

Abstract: In this paper, we propose fuzzy learning models suitable for digital circuit implementation. In the first place, we propose a division-free model which is advantageous in the circuit size and the processing speed. Next, we also consider to improve the accuracy of the division-free model by using ensemble learning. The effectiveness of the proposed methods is demonstrated by numerical simulations. Further, we describe a digital circuit design of the proposed model, and show the effectiveness of the design by the implementation on FPGA.

1. Introduction

Fuzzy reasoning has been successfully applied to many problems such as control, classification, pattern recognition, etc[1]. Digital circuit realization of fuzzy reasoning systems is advantageous in mass production and cost performance, compared with software implementation[2]. One of the problems in the digital circuit realization is processing division operations, which increases the circuit size and the processing speed. Some studies have considered to implement fuzzy reasoning systems using FPGAs (Field Programmable Gate Arrays), which are programmable digital circuit devices[3], [4], [5]. However, most of the implemented systems have no self-tuning function of the fuzzy rule set and download the rule set from an offline host computer.

In this paper, we propose fuzzy learning models suitable for digital circuit implementation, which involve no division operation and are capable of learning without any host computer. With the fist feature, division-free, the models are advantageous in the circuit size and the processing speed. We also consider to improve the accuracy of the division-free model by using ensemble learning. The effectiveness of the proposed methods is demonstrated by numerical simulations. Further, we describe a digital circuit design of the proposed model, and show the effectiveness of the design by the implementation on FPGA.

2. Fuzzy reasoning model

Let $\boldsymbol{x} = (x_1, \cdots, x_m)$ denote the input variable. Let y denote the output variable. Then the rules of simplified fuzzy reasoning model can be expressed as

$$R_i \ : \ \text{if } x_1 \text{ is } M_{1j} \text{ and} \cdots x_m \text{ is } M_{im} \text{ then } y \text{ is } w_i \qquad (1)$$

where $i \in \{1, \cdots, n\}$ is a rule number, $j \in \{1, \cdots, m\}$ is a variable number, M_{ij} is a membership function of the antecedent part, and w_i is the weight of the consequent part.

A membership value of the antecedent part μ_i for input $\boldsymbol{x}$ is expressed as

$$\mu_i = \prod_{j=1}^{m} M_{ij}(x_j), \qquad (2)$$

where M_{ij} is the triangular membership function of the antecedent part. Let c_{ij} and b_{ij} denote the center and the width values of M_{ij}, respectively. Then, M_{ij} is expressed as

$$M_{ij}(x_j) = \begin{cases} 1 - \frac{2 \cdot \left| x_j - c_{ij} \right|}{b_{ij}} & (c_{ij} - \frac{b_{ij}}{2} \le x_j \le c_{ij} + \frac{b_{ij}}{2}) \\ 0 & \text{(otherwise).} \end{cases} \qquad (3)$$

The output y^* of fuzzy reasoning can be derived from Eq.(4).

$$y^* = \frac{\sum_{i=1}^{n} \mu_i \cdot w_i}{\sum_{i=1}^{n} \mu_i} \qquad (4)$$

The objective function E is defined to evaluate the reasoning error between the desirable output y^r and the reasoning output y^* of system.

$$E = \frac{1}{2} \left(y^* - y^r \right)^2 \qquad (5)$$

In order to minimize the objective function E, the parameters $\theta \in \{c_{ij}, b_{ij}, w_i\}$ are updated based on the descent method as follows.

$$\theta(t + 1) = \theta(t) - K_\theta \frac{\partial E}{\partial \theta} \qquad (6)$$

For each parameter $\theta \in \{c_{ij}, b_{ij}, w_j\}$, the updatation value is as follows.

$$\frac{\partial E}{\partial c_{ij}} = \frac{\partial E}{\partial w_i}(w_i - y^*)\frac{2 \cdot \mathrm{sgn}(x_j - c_{ij})}{b_{ij} M_{ij}} \qquad (7)$$

$$\frac{\partial E}{\partial b_{ij}} = \frac{\partial E}{\partial w_i}(w_i - y^*)\frac{1 - M_{ij}}{b_{ij} M_{ij}} \qquad (8)$$

$$\frac{\partial E}{\partial w_i} = \frac{\mu_i}{\sum_{k=1}^{n} \mu_k}(y^* - y^r) \qquad (9)$$

Let $D = \{(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(P)}, y^{(P)})\}$ denote the training data. Then the construction algorithm is presented as follows.

Construction Algorithm

[Step1] Let $p = 1$.
[Step2] Calculate the output of fuzzy inference y^* for $(\boldsymbol{x}^p, y^p)$ according to Eq. (4).
[Step3] Update w_i according to Eqs. (6) and (7) and recalculate the output y^*.
[Step4] Update c_{ij} and b_{ij} of the membership functions in th antecedent part, according to Eqs. (6), (7) and (8).
[Step5] If $p = P$, terminated the algorithm. Otherwise $p \leftarrow p + 1$ and go to Step2.

3. The proposed models

3.1 Division-free model

The fuzzy model described in the previous section involves division operations in Eqs. (3), (4), (7), (8) and (9). In order to make these division operations unnecessary, we consider to impose two constraints on the fuzzy model. The first constraint is imposed on the arrangement of fuzzy rules. The constraint is as follows:

$$\sum_{i=1}^{n} \mu_i = 1. \tag{10}$$

An example of the rule arrangements satisfying the constraint is shown in Fig.1. Specifically, we use the rule arrangement that at most two rules are overlapped at any region as shown in Fig.1.

The second constraint is that the parameters c_{ij} and b_{ij} are not updated. Any rule arrangement satisfying Eq.(10) will be turned into violating one by updating c_{ij} and b_{ij} by Eqs. (7) and (8). Some normalization technique is required for updating c_{ij} and b_{ij}, but this makes the system complex. That is why we adopt the second constraint.

Let us show that the two constraints make division unnecessary. From Eq.(10), Eq.(4) is rewritten as follows:

$$y^* = \frac{\sum_{i=1}^{n} \mu_i w_i}{\sum_{i=1}^{n} \mu_i} = \sum_{i=1}^{n} \mu_i w_i. \tag{11}$$

Eq.(9) is also rewritten as follows:

$$\frac{\partial E}{\partial w_i} = \frac{\mu_i}{\sum_{k=1}^{n} \mu_k}(y^* - y^r) = \mu_i(y^* - y^r). \tag{12}$$

Further, Eqs.(7) and (8) are not calculated, because c_{ij} and b_{ij} are not updated. In Division-free model, The construction algorithm of fuzzy model can be simplified as follows.

Construction Algorithm

[Step1] Let $p = 1$.
[Step2] Calculate the output of fuzzy inference y^* for $(\boldsymbol{x}^p, y^p)$ according to Eq. (11).
[Step3] Update w_i according to Eqs. (6) and (12).
[Step4] If $p = P$, terminate the algorithm. Otherwise $p \leftarrow p + 1$ and go to Step2.

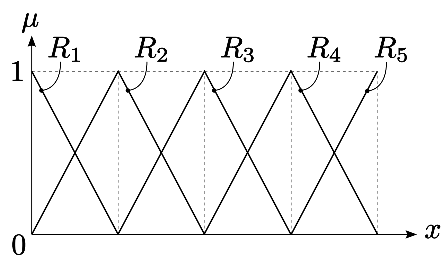


Figure 1. Rule arrangement satisfying constraint $\sum_i \mu_i = 1$.

3.2 Division-free with ensemble learning

The division-free model described in the previous subsection is of lower accuracy than the standard model. In order to compensate for the accuracy degradation, we introduce ensemble learning for the division-free model. Ensemble learning is an approach that a high-accuracy system is constructed by combining multiple weak learners, where a weak learner means a small system consists of few parameters such as rules. It is known that the ensemble learning is effective to improve the generalization ability of learning systems.

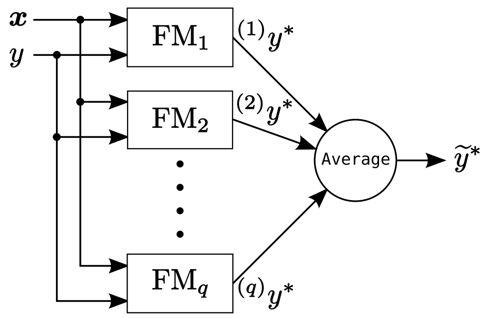


Figure 2. Ensemble learning model.

We use an ensemble learning model as shown in Fig.2. The model consists of q division-free fuzzy sub-systems $\text{FM}_1, \cdots, \text{FM}_q$, each of which is a division-free system described in the previous subsection. Let ${}^{(k)}y^*$ be the output of FM_k. Then, the output of the whole system, $\widetilde{y}^*$ is given as

$$\widetilde{y}^* = \frac{1}{q}\sum_{k=1}^{q} {}^{(k)}y^*. \tag{13}$$

Every sub-systems FM_k, $k \in \{1, \cdots, q\}$ are trained in parallel by using the same learning data. However, in order to successfully perform ensemble learning, it is important that weak learners acquire different properties. Fig.3 shows two types of ensemble systems, each of which consists of three sub-systems and each of whose output is determined by majority vote. In the figure, gray and white cells indicate correct and wrong outputs for some pattern respectively.

As shown in Fig.3(a), if all sub-systems are of same properties their majority vote cannot reduce the error, even if the are of high-accuracy. On the other hand, as shown in Fig.3(b), if all sub-systems have different properties, their majority vote will reduce the error. To this end, we assign every sub-systems FM_k different fuzzy rule arrangements. Specifically, for sub-system FM_k, the membership functions are uniformly distributed over the domain $L-(k-1)\alpha \leq x_j \leq U+(k+1)\alpha$, where α is the degree of variation, and L and U are the lower and the upper bounds for FM_1, respectively.

3.3 Numerical simulations

In order to show the effectiveness of the proposed methods, we perform numerical simulations on the approximation of

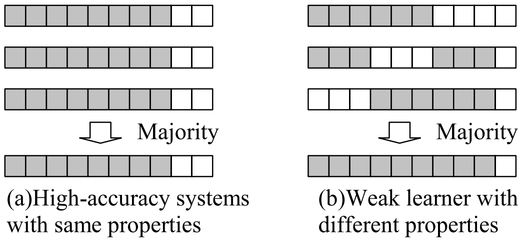


Figure 3. Two types of ensemble learning systems.

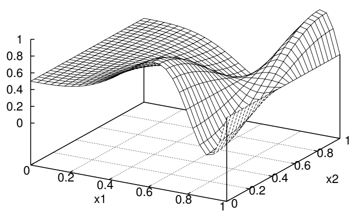
the following two functions.

$$\text{Function I}: \; y = \frac{\sin(2\pi x_1^3)\cdot\cos(\pi x_2)+1}{2} \quad (14)$$

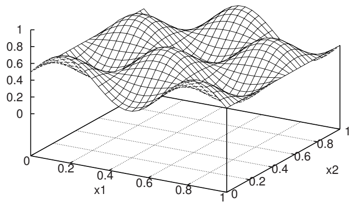
$$\text{Function II}: \; y = \frac{\sin(3\pi x_1)\cdot\cos(3\pi x_2)+2}{4} \quad (15)$$

The numbers of learning and test data are 256 and $P = 2500$, respectively. Conventional, division-free and ensemble learning models are compared in terms of mean squared error (MSE): $MSE = \sqrt{\frac{1}{P}\sum_{p=1}^{P}(y^*(\boldsymbol{x}^{(p)}) - y^{r(p)})^2}$. For the ensemble learning model, the number of units is $q = 4$.

The simulation results are shown in Fig. 5. For both functions, we can observe the following tendencies.
1) The accuracy of the proposed methods "Division-free" and "Ensemble" approach the conventional one as the number of rules increases.
2) The accuracy of "Ensemble" is much enhanced from the one of "Division-free".

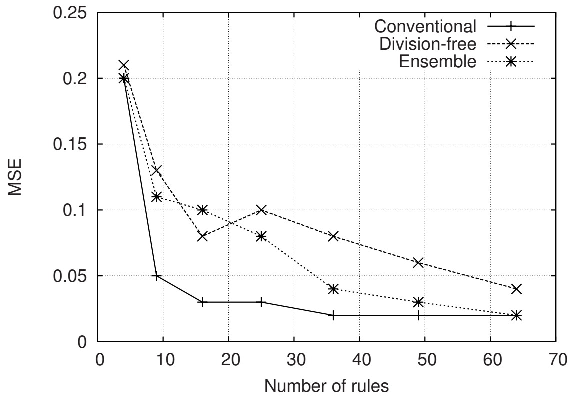


(a) For function I.

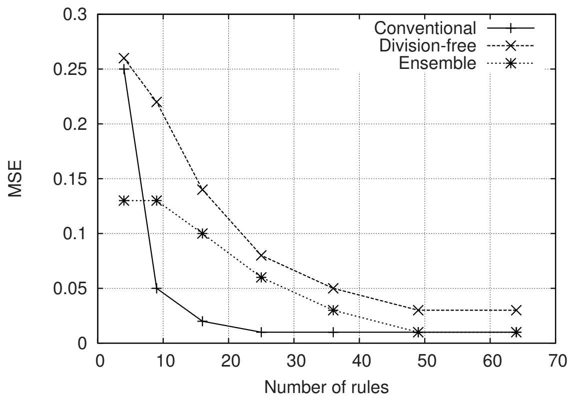


(b) For function II.

Figure 4. Function.



(a) For function I.



(b) For function II.

Figure 5. MSE versus the number of rules.

4. Circuit design

We implement the proposed division-free model on FPGA. The block diagram of the implemented design is shown in Fig.6. In the diagram, italic serif letters in the modules indicate the output signal names of modules. And, letters outside of the modules indicate the input signal names for control as follows.

- we : Write enable signal of RAM w_i.
- en : Enable signal of register y^*.
- reset : Reset signal of register y^*.

Note that the control signals we and en form the counter output by using simple combinational logic. Five ROMS, in advance, store the training data set $D = \{(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(P)}, y^{(P)})\}$ and fuzzy rule parameters c_{i1}, c_{i2}. Only one RAM stores the weight of the consequent part w_i. Note that the width parameters b_{ij} are included modules M_{i1} and M_{i2} as constants. The implemented circuit operates as follows :
[Step1] p = 1.
[Step2] reset = 1, i = 1. Reset y^* = 0.
[Step3] reset = 0.
Allot the training data x_1^p,x_2^p,y^p,center value of antecedent part c_{i1}, c_{i1}. Calculate the membership value of antecedent part μ_i.
[Step4] en = 1. we = 0.

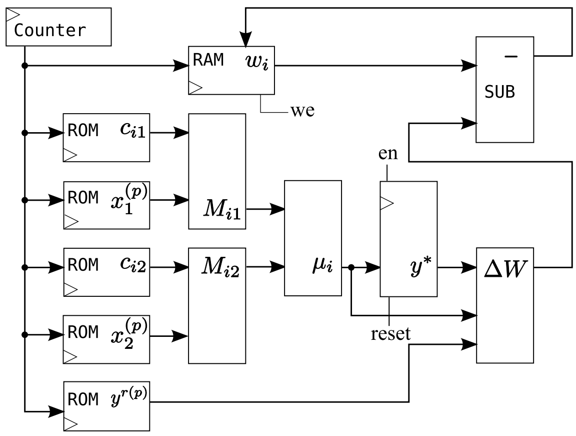


Figure 6. Block diagram of designed circuits.

If $i = n$, then go to Step5. otherwise $i \leftarrow i + 1$ and Step2.
[Step5] en = 0. we = 1. Calculate ΔW and update w_i.
[Step6] If $p = P$, terminate the circuit. Otherwise $p \leftarrow p + 1$ and go to Step2.

The design environment is Xilinx ISE WebPack 9.1i, the targeted devise is Spartan2E XC2S300E, and the design is coded in VHDL. Major specifications of the implemented system are summarized in Table.1.

The implemented hardware system is compared with a software implementation in terms of processing speed and accuracy. The environments of the software implementation are as follows: Intel Celeron 2.53GHz, 1.25 GB Memory, Borland C++ Compiler 5.5 and floating number representation of 4 Byte. The processing speed is evaluated in the case where the number of learning iterations is 1000 and the number of learning data is 256.

The results are summarized in Tables 2 and 3. Table 2 shows that the software and the hardware implementations are not much different in terms of accuracy. Further, Table 3 shows that the hardware implementation is faster than the software one.

The circuit size and the operating speed of the division-free system are compared with the conventional one, which contains a single divider to perform Eqs. (3), (4), (7), (8) and (9). The results are summarized in Table 4. The results show that the proposed implementation is approximately 4 times faster and twice smaller than the conventional one.

Table 1. Specifications of the implemented system.

Item	Spec.
Numeric coding	16 bits fixed-point
# of rules	16, 64
Input data	2-dimensional vector
Output data	scalar
# of training data	256

Table 2. Accuracy comparison in MSE.

	# rules	Software	Hardware
Function I	16	0.0842	0.1349
	64	0.0406	0.0593
Function II	16	0.1452	0.1020
	64	0.0259	0.0384

Table 3. Processing speed comparison.

# rules	Software	Hardware
16	439 ms	213 ms
64	1109 ms	851 ms

Table 4. Operating speed and circuit size for conventional and division-free implementations.

	Conventional	Division-free
Speed	9.18 MHz	37.34 MHz
Size	307,693 Gates	110,885 Gates

5. Conclusions

In this paper, we have presented two methods of fuzzy learning models suitable for digital circuit implementation and have examined their validity through numerical experiments.We can observe the following tendencies.
1) The accuracy of the proposed methods "Division-free" and "Ensemble" approach the conventional one as the number of rules increases.
2) The accuracy of "Ensemble" is much enhanced from the one of "Division-free".
Further, we implement the proposed division-free model on FPGA, and we show its effectiveness as follows.
1) Hardware implementation is faster than the software one.
2) The proposed implementation is approximately 4 times faster and twice smaller than the conventional one.

One of our future works is to implement proposed division-free with ensemble learning model on FPGA.

References

[1] J.M. Mendel, "Fuzzy Logic Systems for Engineering: A Tutorial", Proc. of IEEE, vol.83, no.3, pp.345-377, 1995.

[2] Z. Kovacic and S. Bogdan, Fuzzy controller Design: Theory and Applications, CRC Press, 2005.

[3] E. Lago, M.A. Hinojosa, C.J. Jimenez, A. Barriga, and S. Sanchez-Solano, "FPGA Implementation of Fuzzy Controllers", Proc. of 12th Conf. on Design of Circuits and Integrated Systems, pp.715-720, 1997.

[4] M. Mckenna and B.M. Wilamowski, "Implementing a Fuzzy System on a Field Programmable Gate Array", Proc. of the Int. Joint Conf. on Neural Networks, pp.189-194, 2001.

[5] M.A. Manzoul and D. Jayabharathi, "FPGA for Fuzzy Controllers", IEEE Trans. on Systems, Man and Cybernetics, vol.25, no.1, pp.213-216, 1995.