R.J. Marks II, S.Oh, P. Arabshahi, T.P. Caudell, J.J. Choi and B.G. Song,
"Steepest descent of min-max fuzzy if-then rules", Proceedings of the International Joint Conference on Neural Networks,
Beijing, vol. III, pp. 471-477, November 3-6, 1992.

IJCNN INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS

Beijing, China
November 3–6, 1992

INNS

INTERNATIONAL
NEURAL NETWORK
SOCIETY

CNNC

IEEE NNC

電子工業出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

Steepest Descent Adaptation of Min-Max Fuzzy If-Then Rules

R.J. Marks II, S. Oh, P. Arabshahi *,
T.P. Caudell †, J.J. Choi ‡, B.G. Song *

* Dept. of Electrical Engineering
University of Washington FT-10
Seattle, WA 98195

† Boeing Computer Services
P.O. Box 24346, MS 7L-22
Seattle, WA 98124-0346

‡ Boeing Computer Services
P.O. Box 24346, MS 6C-04
Seattle, WA 98124-0346

Abstract

A new technique for adaptation of fuzzy membership functions in a fuzzy inference system is proposed. The technique relies upon the isolation of the specific membership function that contributed to the final decision, followed by the updating of this function's parameters using steepest descent. The error measure used is thus back propagated from output to input, through the min and max operators used during the inference stage. This is feasible because the operations of min and max are continuous differentiable functions and therefore can be placed in a chain of partial derivatives for steepest descent backpropagation adaptation. More interestingly, the partials of min and max (or any other order statistic, for that matter) act as 'pointers' with the result that only the function that gave rise to the min or max is adapted; the others are not. To illustrate, let $\alpha = \max[\beta_1, \beta_2, \cdots, \beta_N]$. Then $\partial\alpha/\partial\beta_n = 1$ when β_n is the maximum and is otherwise zero. We apply this property to the fine tuning of membership functions of fuzzy min-max decision processes and illustrate with an estimation example.

1 Introduction

Fuzzy membership functions chosen for a control or decision process may require adaptation for purposes of fine tuning or adjustment to stationarity changes in the input data. Use of neural networks to perform this adaptation has been proposed by Lee et al. [1]. Other techniques proposed can be found in [3], [4], [5]. Our method more closely parallels that proposed by Nomura, Hayashi and Wakami [2]. In their work, membership functions were parameterized and steepest descent was performed with respect to each parameter using an error criterion, in order to obtain the set of parameters minimizing the error. To straightforwardly differentiate the error function with respect to each parameter, they used products for the fuzzy intersection operation. The output error backpropagated this way, was used to adjust the fuzzy membership functions.

In this paper, we show that the more conventionally used minimum operation for fuzzy intersection and maximum operation for fuzzy union can be similarly backpropagated. Unlike the method of Nomura *et al.* which updates all fuzzy membership function parameters in each stage, the method proposed herein results only in the adjustment of the fuzzy membership functions that gave rise to the control action or decision output. Backpropagation of fuzzy min-max rules allows for fine tuning and adaptation of membership functions using performance data.

2 Differentiation of MIN and MAX Operations

Differentiation of the min or max operations results in a 'pointer' that specifies the source of the minimum or maximum. To illustrate, let

$$\alpha = \max[\beta_1, \beta_2, \cdots, \beta_N]$$

$$= \sum_{x=1}^{N} \beta_x \prod_{\ell \neq x} U(\beta_x - \beta_\ell) \tag{1}$$

where $U(\cdot)$, a unit step function, is 1 for positive arguments and is zero otherwise. Note that the max operator in Eq. 1 is continuous and can be differentiated as

$$\begin{aligned} \frac{\partial \alpha}{\partial \beta_n} &= \prod_{\ell \neq n} U(\beta_n - \beta_\ell) \\ &= \begin{cases} 1 & ; \text{ if } \beta_n \text{ is maximum} \\ 0 & ; \text{ otherwise} \end{cases} \end{aligned} \tag{2}$$

Similarly, let

$$\begin{aligned} \delta &= \min[\gamma_1, \gamma_2, \cdots, \gamma_M] \\ &= \sum_{x=1}^{M} \gamma_x \prod_{\ell \neq x} U(\gamma_\ell - \gamma_x) \end{aligned} \tag{3}$$

The min function is also continuous and

$$\begin{aligned} \frac{\partial \delta}{\partial \gamma_n} &= \prod_{\ell \neq n} U(\gamma_\ell - \gamma_n) \\ &= \begin{cases} 1 & ; \text{ if } \gamma_n \text{ is minimum} \\ 0 & ; \text{ otherwise} \end{cases} \end{aligned} \tag{4}$$

Indeed, any order statistic operation (*e.g* the third largest number or, for N odd, the median) can likewise be differentiated. In each case, the partial derivative points to the number or index that gives the order statistic result.

3 Fuzzy Min-Max Estimation

To illustrate adjustment of fuzzy membership functions by steepest descent, consider the fuzzy estimation problem illustrated in Fig. 1. We wish to generate an estimate $f(x_1, x_2)$ of a target function $t(x_1, x_2)$ using a set of fuzzy IF ... THEN rules. Here we have:

$$t(x_1, x_2) = \sin(\pi x_1)\cos(\pi x_2) \tag{5}$$

The rule table (Table 1) is generated by partioning the domain of $t(x_1, x_2)$, $\{(x_1, x_2) \mid x_1 \in [-1, 1], x_2 \in [-1, 1]\}$ into 64 (8 × 8) regions and assigning a fuzzy membership function to each region in accordance to the values of $t(x_1, x_2)$ in that region. For instance if $t(x_1, x_2)$ takes on values close to 1 in certain regions, then the membership function used for those regions of the domain will be "Positive High" (PH). Initial membership functions for f are thus formed in this way. The values of x_1 and x_2 are fuzzified in a similar manner. The initial membership functions chosen are Gaussian and are shown in Figure 2 for x_1, x_2 and $f(x_1, x_2)$.

To illustrate, consider the fuzzy IF ... THEN rules with a positive medium (PM) consequent. These are highlighted in Table 1. Reading from left to right from the top of the table, they are:

IF x_1 is NH AND x_2 is NH

OR

IF x_1 is PH AND x_2 is NH

OR

IF x_1 is NM AND x_2 is NM

OR

⋮

IF x_1 is PZ AND x_2 is PH

THEN

$f(x_1, x_2)$ is PM.

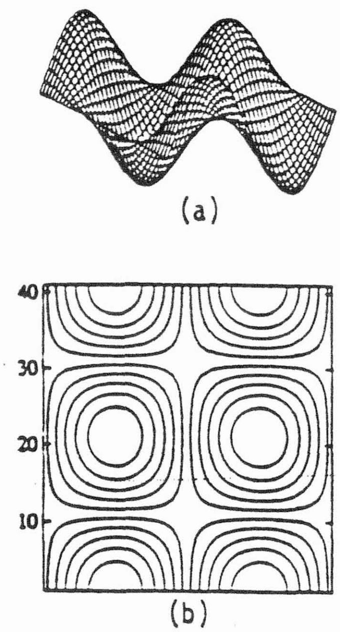

Figure 1: A fuzzy estimation problem: a) 3-D plot and b) contour plot, of the signal to be estimated: $t(x_1, x_2) = \sin(\pi x_1)\cos(\pi x_2)$ over the domain $\{(x_1, x_2) \mid x_1 \in [-1, 1], x_2 \in [-1, 1]\}$.

Similar rules exist for the other five categories of f.

x_1 / x_2	NH	NM	NS	NZ	PZ	PS	PM	PH
NH	*PM*	PS	NS	NM	NM	NS	PS	*PM*
NM	PH	*PM*	NM	NH	NH	NM	*PM*	PH
NS	PH	*PM*	NM	NH	NH	NM	*PM*	PH
NZ	*PM*	PS	NS	NM	NM	NS	PS	*PM*
PZ	NM	NS	PS	*PM*	*PM*	PS	NS	NM
PS	NH	NM	*PM*	PH	PH	*PM*	NM	NH
PM	NH	NM	*PM*	PH	PH	*PM*	NM	NH
PH	NM	NS	PS	*PM*	*PM*	PS	NS	NM

Table 1: Decision Table for fuzzy estimation. Table contents represent the estimated fuzzy value of the output f for a given choice of values for x_1 & x_2. Rules with a consequent of Positive medium (PM) are highlighted.

3.1 Feedforward Procedure

For purposes of analysis, let the membership functions for the variable x_1 be denoted by μ_1^i, $i = 1, 2, \cdots N$, those for the variable x_2 by μ_2^j, $j = 1, 2, \cdots M$, and those for the output variable f by μ_3^k, $k = 1, 2, \cdots K$.

For a given output membership function μ_3^k, the rules, as shown in Table 1, are of the form:

If x_1 is μ_1^i and x_2 is μ_2^j OR If x_1 is μ_1^l and x_2 is μ_2^m OR $\cdots$

Then ... f is μ_3^k.

Let us define a set S_k as follows:

$$S_k = \{l, m \mid \mu_1^l \text{ and } \mu_2^m \text{ are antecedents of a rule with consequent } \mu_3^k\} \tag{6}$$

The familiar operations to arrive at the output are as follows.

1. Perform a pairwise fuzzy intersection (*e.g.* minimum or outer product) on each of the membership values of x_1 and x_2 in μ_1^l and μ_2^m for every rule with consequent μ_3^k, forming activation values ζ:

$$\zeta_{lm}^k = \min_{l,m \in S_k} (\mu_1^l(x_1), \mu_2^m(x_2)) \tag{7}$$

2. Collect activation values for like output membership functions and perform a fuzzy union (*e.g.* maximum).

$$w_k = \max_{l,m \in S_k} (\zeta_{lm}^k) \tag{8}$$

3. These values are defuzzified to generate the output estimated value, $f(x_1, x_2)$, by finding the centroid of the composite membership function μ:

$$\mu = \sum_{k=1}^{K} w_k \mu_3^k \tag{9}$$

$$f(x_1, x_2) = \frac{\sum_{k=1}^{K} w_k c_k A_k}{\sum_{k=1}^{K} w_k A_k} \tag{10}$$

where

$$A_k = \int \mu_3^k(x)\, dx, \tag{11}$$

$$c_k = \frac{\int x \mu_3^k(x)\, dx}{\int \mu_3^k(x)\, dx} \tag{12}$$

A_k and c_k are, respectively, the area and centroid of the consequent membership function μ_3^k.

Backpropagation Adjustment

Expert heuristics are typically used to specify the membership functions for the input (x_1, x_2) and output (f). These functions can be adapted or fine tuned using supervised learning. The steps to adapt the input membership functions are as follows.

We first form the error function by taking the squared difference between the estimated output f, and the desired target value t:

$$E = \frac{1}{2}(f - t)^2 \tag{13}$$

Assume now that we wish to update parameters of a Gaussian membership function that appears either in the antecedent or the consequent of a rule. Denote these parameters by $m_l^i[q]$ and the corresponding membership function by μ_l^i. In our example, for $l = 1, 2$, the index $i = 1, 2, \cdots 8$ and for $l = 3$, the index $i = 1, 2, \cdots 6$; q =1,2, and:

$$\mu_l^i(x) = \exp\left(\frac{(x - m_l^i[1])^2}{2(m_l^i[2])^2}\right) \tag{14}$$

The steepest descent update rule is:

$$m_l^i[q] \Longleftarrow m_l^i[q] - \alpha \frac{\partial E}{\partial m_l^i[q]}. \tag{15}$$

We have, for the general case:

$$\frac{\partial E}{\partial m_l^i[q]} = \frac{\partial E}{\partial f} \sum_{k=1}^{K} \left(\frac{\partial f}{\partial w_k} \frac{\partial w_k}{\partial \mu_l^i} \right) \frac{\partial \mu_l^i}{\partial m_l^i[q]} \tag{16}$$

This in turn can be written in the following way (see Eqs. 7 and 8):

$$\frac{\partial E}{\partial m_l^i[q]} = \frac{\partial E}{\partial f} \sum_{k=1}^{K} \left(\frac{\partial f}{\partial w_k} \sum_{l,m \in S_k} \left(\frac{\partial w_k}{\partial \zeta_{lm}^k} \frac{\partial \zeta_{lm}^k}{\partial \mu_l^i} \right) \right) \frac{\partial \mu_l^i}{\partial m_l^i[q]} \tag{17}$$

From Eqs. 2 and 4, and referring to Eqs. 7 and 8 we obtain:

$$\frac{\partial w_k}{\partial \zeta_{lm}^k} = \delta[w_k - \zeta_{lm}^k] \tag{18}$$

$$\frac{\partial \zeta_{lm}^k}{\partial u_l^i} = \delta[\zeta_{lm}^k - \mu_l^i] \tag{19}$$

where $\delta[\cdot]$, the Kronecker delta function, is equal to one for zero arguments and is zero otherwise.

Substituting the above two equations in Eq. 17, we obtain:

$$\frac{\partial E}{\partial m_l^i[q]} = \frac{\partial E}{\partial f} \sum_{k=1}^{K} \left(\frac{\partial f(w_k)}{\partial w_k} \sum_{l,m \in S_k} \left(\delta[w_k - \zeta_{lm}^k] \delta[\zeta_{lm}^k - \mu_l^i] \right) \right) \frac{\partial \mu_l^i}{\partial m_l^i[q]} \tag{20}$$

It is clear that the two Kronecker delta functions now serve to isolate the membership function whose parameter is being updated. Other membership functions that are not used in the decision process are not adapted. Eq. 20 finally simplifies to:

$$\frac{\partial E}{\partial m_l^i[q]} = \frac{\partial E}{\partial f} \frac{\partial f(\mu_l^i(x_j))}{\partial w_k} \frac{\partial \mu_l^i}{\partial m_l^i[q]} \tag{21}$$

where

$$\frac{\partial f}{\partial w_k} = \frac{A_k \sum_{p=1}^{K} w_p A_p (c_k - c_p)}{(\sum_{p=1}^{K} w_p c_p)^2} \tag{22}$$

In general μ_l^i is a function of many parameters $m_l^i[q]$, $q = 1, 2, \cdots$. For our estimation problem, using Gaussian membership functions, there are two parameters to adapt. These are the mean ($m_l^i[1]$), and the variance ($m_l^i[2]$). We thus have:

$$\frac{\partial \mu_l^i}{\partial m_l^i[1]} = \mu_l^i \frac{(x - m_l^i[1])}{(m_l^i[2])^2} \tag{23}$$

$$\frac{\partial \mu_l^i}{\partial m_l^i[2]} = \mu_l^i \frac{(x - m_l^i[1])^2}{(m_l^i[2])^3} \tag{24}$$

4 Results

We present here results of the application of this technique to the estimation problem discussed in section 3. Fig. 3 illustrates the input and output membership functions after adaptation and Fig. 4 shows the (much improved) estimation result.

Acknowledgements

This work was supported in part by Boeing Computer Services and the Washington Technology Center.

References

[1] C.T. Lin and C.S.G. Lee, "Neural network based fuzzy logic control and decision system", *IEEE Trans. Computers*, vol. 40, pp. 1320–1336, 1991.

[2] H. Nomura, I. Hayashi and N. Wakami, "A learning method of fuzzy inference by descent method", *Proc. IEEE Int. Conf. Fuzzy Systems, FUZZ-IEEE '92*, San Diego, CA, March 8-12, 1992.

[3] Y. Hayashi, E. Czogala, and J.J. Buckley, "Fuzzy neural controller", *Proc. IEEE Int. Conf. Fuzzy Systems, FUZZ-IEEE '92*, San Diego, CA, March 8-12, 1992.

[4] J.S.R. Jang, "Fuzzy controller design without domain experts", *Proc. IEEE Int. Conf. Fuzzy Systems, FUZZ-IEEE '92*, San Diego, CA, March 8-12, 1992.

[5] Li-Xin Wang and Jerry M. Mendel, "Back-propagation fuzzy system as nonlinear dynamic system identifiers", *Proc. IEEE Int. Conf. Fuzzy Systems, FUZZ-IEEE '92*, San Diego, CA, March 8-12, 1992.

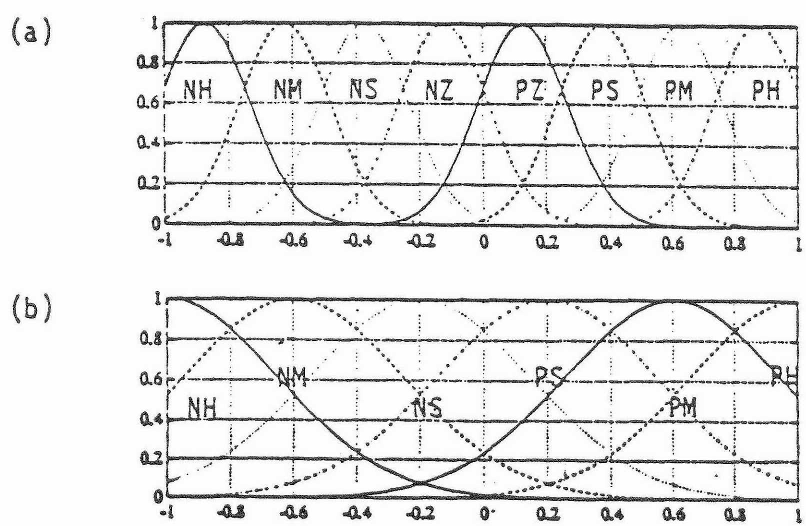

Figure 2: Initial membership functions for a) x_1, x_2 and b) $f(x_1, x_2)$. Here NH ≡ Negative High, NM ≡ Negative Medium, NS ≡ Negative Small, NZ ≡ Negative Zero, PZ ≡ Positive Zero, ···

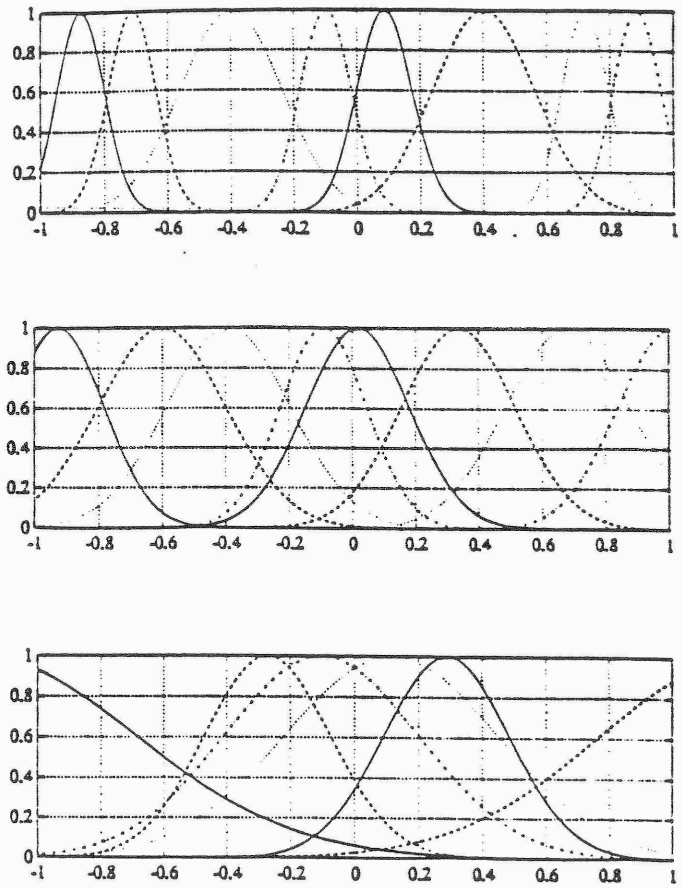

Figure 3: Final membership functions for a) x_1 b) x_2 and c) $f(x_1, x_2)$. Here NH $\equiv$ Negative High, NM $\equiv$ Negative Medium, NS $\equiv$ Negative Small, NZ $\equiv$ Negative Zero, PZ $\equiv$ Positive Zero, $\cdots$

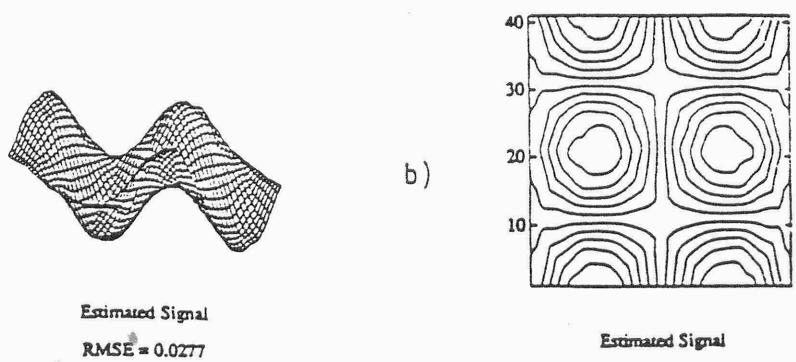

Figure 4: Result of fuzzy estimation: a) 3-D plot and b) contour plot, of the estimated signal: $f(x_1, x_2) = \sin(\pi x_1)\cos(\pi x_2)$ over the domain $\{(x_1, x_2) \mid x_1 \in [-1, 1], x_2 \in [-1, 1]\}$