Fuzzy Finite Element Method with Gaussian RBF Network

Yoshihiro SATOH*, Tetsuya MIYOSHI*, Kazunori NAGASAKA*, Hidetomo ICHIHASHI*

*Department of Industrial Engineering
College of Engineering, University of Osaka Prefecture
1-1 Gakuencho, Sakai, Osaka 593, Japan

Abstract
In this paper, we propose fuzzy finite element method as an application of Gaussian radial basis fuctions(RBF) networks, in which the finite elements are fuzzy subsets defined by Gaussian membership functions. Illustrative examples of Laplace equation are shown with the computational results of approximate solutions.

Key words: finite element method, radial basis function, Gaussian function, fuzzy set

1 Introduction

For approximating nonlinear mappings, especially interpolating the data points in a high dimensional space, multilayer networks with adaptive learning algorithm are widely applied. Among others, the most popular networks consist of sigmoidal basis functions [1]. In 1988, J.Moody and C.Darken [4] proposed radial basis functions(RBF) networks, a technique for interpolating in a high dimensional space, and reported that RBF networks are potentially 1000 times faster than the sigmoidal basis function networks with backpropagation for comparable error rates. T.Poggio and F.Girosi showed that regularization principles lead to approximation schemes which are equivalent to networks with one hidden layer [8, 9]. Gaussian RBFs have recently applied to the various kinds of problems such as handwritten character recognition, speech recognition, view based object recognition[10], time series prediction and system opmization [3]. The network can be regarded as a three-layered network (input, output and hidden layer) with Gaussian hidden units. The output of the network is a linear combination of hidden outputs and the network can be reinterpreted as fuzzy production rules [2, 5, 6, 7]. The learning algorithm is relatively simple compared with the sigmoidal neural networks with backpropagation. Hence, the network can be applicable to more complicated problems with supervised or unsupervised learning.

In the problem solving a partial differential equation or a system of differential equation, the finite element method (FEM) is commonly used as a powerful tool of numerical procedure. Though FEM has advanced with the development of digital computer, accuracy of the solution of FEM largely depends on the selection of knots, and it is not easy to change the knots adaptively.

In this paper, we propose a solution method to find an approximate solution of partial differential equations as an application of Gaussian RBF networks. The finite elements in the FEM are replaced by fuzzy subsets defined by Gaussian membership functions, and the adaptive learning method is employed. As an illustrative example, we show computational results of the approximate solution of Laplace differential equations.

2 Fuzzy finite element defined by Gaussian membership function

Let $A_{i,k}$ denote the membership function of the k-th fuzzy rule in the domain of the i-th input variable x_i. The k-th rule is written as

"If x_1 is $A_{1,k}$ and x_2 is $A_{2,k}$ ... and x_n is $A_{n,k}$
then y is w_k."

The conclusion part of the fuzzy reasoning rule which infers the output y is simplified as real number w_k. The final output(y) is written as

$$y = \sum_k w_k \cdot \mu_k = \sum_k w_k \cdot (\prod_i A_{i,k}(x_i)) \qquad (1)$$

where μ_k is the compatibility degree of the premise part of the k-th fuzzy rule, which is computed with the algebraic product operator. This model is a simplified fuzzy model[2]. In the case of Gaussian membership function, $A_{i,k}$ is defined by

$$A_{i,k}(x_i) = \exp\left(-\frac{(x_i - a_{i,k})^2}{b_{i,k}}\right)$$

where the parameters $a_{i,k}$ and $b_{i,k}(i = 1, ..., n)$ are given for each k and are changed in the training procedure. μ_k has the form of Gaussian function $F(\| x - a \|)$ and y in Eq.(1) is its linear combination,$i.e.$ the network of Gaussian radial basis functions[4]. The model has three layers, in which the layers are fully interconnected with adjacent layers, but there are no interconnection within the layers (See Figure 1).

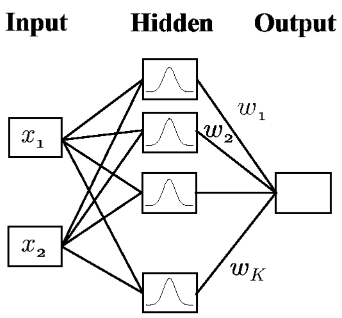

Figure 1: A neural network with Gaussian hidden unit

The triangular finite elements in FEM are replaced by the fuzzy sets whose membership function is Gaussian functions. In other words, the shape function (the basis function) is Gaussian.

Since the approximate function consists of Gaussian functions, it is infinitely differentiable. Hence, any partial differential equations and boundary conditions can be approximated using y in Eq.(1). Let us consider a potential problem governed by the Laplace equation:

$$\nabla^2 f(x_1, x_2) = 0 \qquad on \qquad R$$

where R is some domain subject to the boundary conditions:

$$f(x_1, x_2) = g(x_1, x_2) \qquad on \qquad B_1$$

$$\frac{\partial f(x_1, x_2)}{\partial \boldsymbol{n}} = h(x_1, x_2) \qquad on \qquad B_2$$

where $B = B_1 \bigcup B_2$ is the boundary of R and $\boldsymbol{n}$ is the unit vector of the outward normal to B at the boundary point. The function g and h are prescribed. The Laplace equation is approximated using y in Eq.(1) as:

$$\nabla^2 y(x_1, x_2) = 0 \qquad on \qquad R \qquad (2)$$

$$y(x_1, x_2) = g(x_1, x_2) \qquad on \qquad B_1 \qquad (3)$$

$$\frac{\partial y(x_1, x_2)}{\partial \boldsymbol{n}} = h(x_1, x_2) \qquad on \qquad B_2 \qquad (4)$$

$\frac{\partial^2 y}{\partial x_i^2}$ in Eq.(2) can be written as:

$$\frac{\partial^2 y}{\partial x_i^2} = \sum_k \left(\frac{4(x_i - a_{i,k})^2 - 2b_{i,k}}{b_{i,k}^2} \cdot w_k \cdot \mu_k \right) \qquad (i = 1, 2)$$

Let the left side of the Eqs.(2)~(4) be $F(\boldsymbol{x})$, $G(\boldsymbol{x})$ and $H(\boldsymbol{x})$ respectively. Then, a cost function can be described as:

$$\begin{aligned} E(\boldsymbol{x}) &= \frac{1}{2} \cdot \{F(\boldsymbol{x})^2 + K_1 \cdot (G(\boldsymbol{x}) - g(\boldsymbol{x}))^2 \\ &\quad + K_2 \cdot (H(\boldsymbol{x}) - h(\boldsymbol{x}))^2\} \end{aligned} \qquad (5)$$

where K_1 and K_2 are positive constants. The learning rule based on the least mean square is

$$\begin{aligned} w_k^{NEW} &= w_k^{OLD} - \alpha \cdot \delta_{w_k} \\ a_{i,k}^{NEW} &= a_{i,k}^{OLD} - \beta \cdot \delta_{a_{i,k}} \\ b_{i,k}^{NEW} &= b_{i,k}^{OLD} - \gamma \cdot \delta_{b_{i,k}} \end{aligned}$$

where α, β and γ denote the positive learning constants corresponding to w_k, $a_{i,k}$ and $b_{i,k}$ respectively. δ_w, δ_a and δ_b are

$$\begin{aligned} \delta_C &= F(\boldsymbol{x}) \cdot \frac{\partial F(\boldsymbol{x})}{\partial C} + K_1 \cdot (G(\boldsymbol{x}) - g(\boldsymbol{x})) \cdot \frac{\partial G(\boldsymbol{x})}{\partial C} \\ &\quad + K_2 \cdot (H(\boldsymbol{x}) - h(\boldsymbol{x})) \cdot \frac{\partial H(\boldsymbol{x})}{\partial C} \\ &\quad C \in \{w_k, a_{i,k}, b_{i,k}\} \end{aligned} \qquad (6)$$

For instance, $\frac{\partial F(\boldsymbol{x})}{\partial C}$ in Eq.(6) are

$$\begin{aligned} \frac{\partial F(\boldsymbol{x})}{\partial w_k} &= \left(\frac{4(x_i - a_{i,k})^2 - 2b_{i,k}}{b_{i,k}^2} \right) \cdot \mu_k \\ \frac{\partial F(\boldsymbol{x})}{\partial a_{i,k}} &= \left(\frac{8(x_i - a_{i,k})^3 - 12(x_i - a_{i,k})b_{i,k}}{b_{i,k}^3} \right) \cdot \mu_k \cdot w_k \\ \frac{\partial F(\boldsymbol{x})}{\partial b_{i,k}} &= \left(\frac{4(x_i - a_{i,k})^4 - 10(x_i - a_{i,k})^2 b_{i,k} + 4b_{i,k}^2}{b_{i,k}^4} \right) \cdot \\ &\quad \mu_k \cdot w_k \end{aligned}$$

3 Numerical examples

In this section, we apply the proposed method to solve partial differential equations with various boundary conditions. First, we take two examples of Laplace equations with the Dirichlet condition and the Neumann condition, which can be solved analytically (Case 1). Then, we take three examples of Laplace equations with discontinuous boundary conditions, which can't be solved analytically (Case 2). The positions of collocations are chosen uniformly, and the initial positions of Gaussian functions are shown in Figure 2.

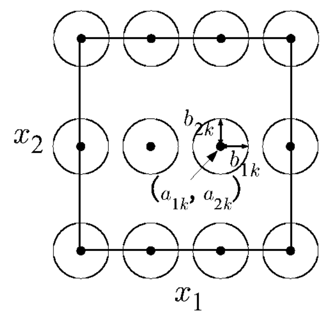

Figure 2: Initial positions of Gaussian functions. 4 × 3 Gaussian functions are used.

The values of K_1 and K_2 in Eq.(5) are set to 30 for all problems. The computational results are shown in Figures 3,5,7-9. Errors between the approximate solution and the analytical solution in the case 1 are shown in Figure 4 and Figure 6. The change of values of the cost function, as learning proceeds, is shown in Table.1.

Case 1

(1) Dirichlet condition:

$$\nabla^2 f(x_1, x_2) = 0 \quad (-1.0 \le x_1 \le 1.0\ ,\ 0 \le x_2 \le \pi)$$

boundary condition

$$f(x_1, x_2) = \begin{cases} 0 & (x_2 = 0\ ,\ x_2 = \pi) \\ \cosh 1.0 \sin x_2 & (x_1 = -1.0\ ,\ x_1 = 1.0) \end{cases}$$

basis functions	3×6
collocations	11×21

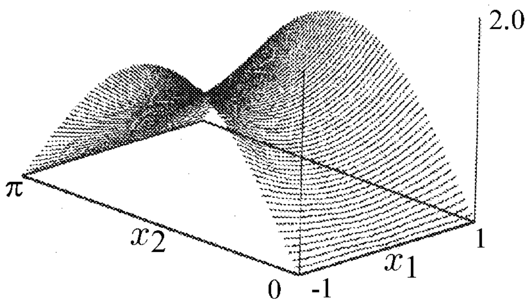

Figure 3: Approximate solution of the Laplace equation with the Dirichlet condition, after 1000 learning iterations.

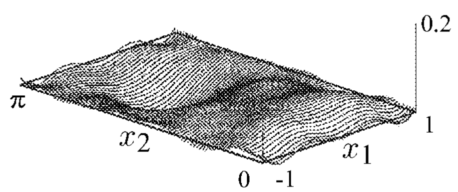

Figure 4: Errors between the approximate solution and the analytical solution.

(2) Neumann condition:

$$\nabla^2 f(x_1, x_2) = 0 \quad (0 \le x_1 \le 1.0,\ \ 0 \le x_2 \le 1.0)$$

boundary conditions

$$f(x_1, x_2) = 0 \quad (x_2 = 0,\ x_2 = 1.0)$$

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = \begin{cases} 0 & (x_1 = 0) \\ \sin \pi x_2 & (x_1 = 1.0) \end{cases}$$

basis functions	3×6
collocations	11×21

Case 2:

(1)

$$\nabla^2 f(x_1, x_2) = 0 \quad (0 \le x_1 \le 1.0,\ \ 0 \le x_2 \le 1.0)$$

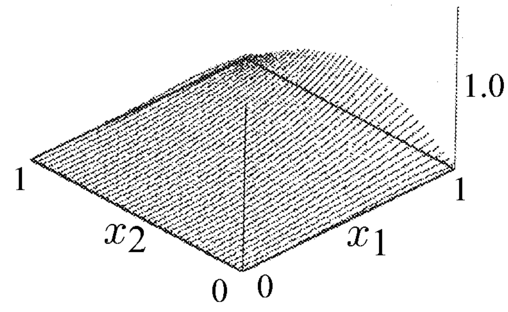

Figure 5: Approximate solution of the Laplace equation with the Neumann condition, after 5000 learning iterations.

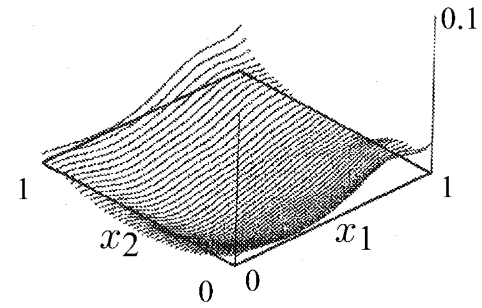

Figure 6: Errors between the approximate solution and the analytical solution.

boundary condition

$$f(x_1, x_2) = \begin{cases} 1.0 & (x_2 = 0) \\ 0.0 & (\text{otherwise}) \end{cases} \tag{7}$$

basis functions	6×3
collocations	21×11

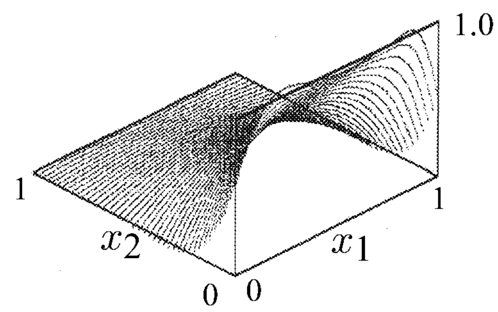

Figure 7: Approximate solution of the Laplace equation with the boundary condition of Eq.(7), after 1000 learning iterations.

(2)

$$\nabla^2 f(x_1, x_2) = 0 \quad (0 \le x_1 \le 1.0,\ \ 0 \le x_2 \le 1.0)$$

boundary condition

$$f(x_1, x_2) = \begin{cases} 1.0 & (x_1 \in [0.1, 0.3] \bigcup [0.7, 0.9], x_2 = 0) \\ 0.0 & (\text{otherwise}) \end{cases} \quad (8)$$

basis functions	6×3
collocations	21×11

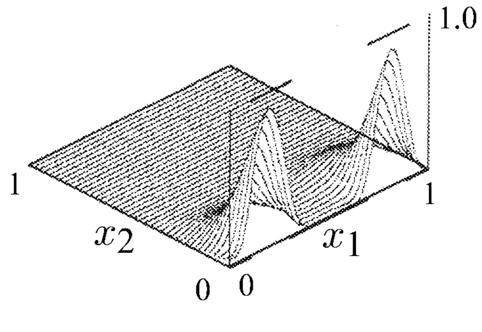

Figure 8: Approximate solution of the Laplace equation with the boundary condition of Eq.(8), after 1000 learning iterations.

(3)

$$\nabla^2 f(x_1, x_2) = 0 \quad (0 \leq x_1 \leq 1.0, \ \ 0 \leq x_2 \leq 1.0)$$

boundary condition

$$f(x_1, x_2) = \begin{cases} 1.0 & (x_1 \in [0, 0.5], \ x_2 = 0) \\ 0.0 & (\text{otherwise}) \end{cases} \quad (9)$$

basis functions	6×3
collocations	21×11

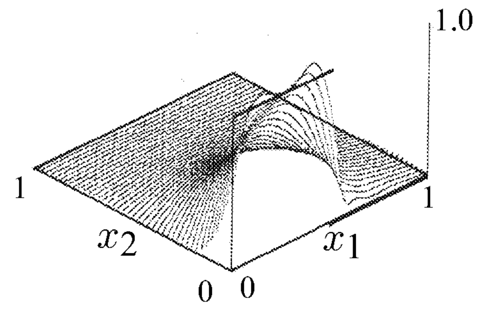

Figure 9: Approximate solution of the Laplace equation with the boundary condition of Eq.(9), after 1000 learning iterations.

Table 1: The change of values of the cost function as learning proceeds.

iteration	0	1000	5000
Case1-(1)	4989.67(4989.67)	121.25(0.55)	–
Case1-(2)	2500.65(2500.65)	309.97(10.58)	–
Case2-(1)	630.00(630.00)	60.54(7.21)	50.92(5.98)
Case2-(2)	300.00(300.00)	70.19(16.52)	58.92(12.06)
Case2-(3)	330.00(330.00)	89.34(21.76)	63.83(16.21)

Note: () indicates the error for the boundary condition.

4 Conclusion

In this paper, we proposed a solution method to partial differential equations. As the output of the model is described with a linear combination of basis functions, the proposed method can be regarded as a kind of FEM in a broad sense. Since the finite element is fuzzified by Gaussian membership function, we call it Fuzzy Finite Element Method (Fuzzy FEM). The major difficulty of the proposed FEM is to determine the learning constants and the weightings in the cost function. These must be determined by trial and error. And it requires long computational time. But with the assistance by the development of high speed computers, Fuzzy FEM can be a convenient solution method of partial differential equations. Fuzzy FEM has the following advantages.

1. Because the adaptive fuzzy meshing is provided, no finite element grid is needed.
2. It requires simpler algorithms and smaller memory size.
3. Accuracy of the solution can be improved with the increase of collocations and basis functions.

References

[1] D.E.Rumelhart, J.L.McClelland and the PDP Research Group, "Parallel Distributed Processing", (Cambridge, MA: MIT Press), 1987.

[2] H.Ichihashi, "Iterative fuzzy modeling and a hierarchical network", Proc. of IFSA 4th World Congress, Belgium, pp.49-52, 1991.

[3] H.Ichihashi, "Neuro-fuzzy finite element method and system optimization", Proc. of 8th Fuzzy System Symposium, Hiroshima, May 26-28, pp.213-216, 1992.

[4] J.Moody and C.Darken, "Learnnig with localized receptive fields", Proc. of 1988 Connectionist Models Summer School, Morgan-Kaufmann, 1988.

[5] L.A.Zadeh, "Fuzzy algorithms", Information Control, Vol.12, pp.94-102, 1968.

[6] L.A.Zadeh, "Outline of a new approach to analysis of complex systems and decision processes", IEEE Transactions on System Man and Cybernetics, Vol.SMC-2, pp.28-44, 1973.

[7] M.Sugeno and G.T.Kang, "Structure identification of fuzzy model", Fuzzy Sets and Systems, Vol.28, pp.15-33 ,1988.

[8] T.Poggio and F.Girosi, "A theory of networks for approximation and learning", A.I. Memo No.1140, C.B.I.P. paper No.31, 1989.

[9] T.Poggio and F.Girosi, "Regularization algorithms for learning that are equivalent to multilayer networks", Sciences,Vol.247,pp.978-982, 1990.

[10] T.Poggio and S.Edelman, "A network that learns to recognize 3D objects", Nature, Vol.343, pp.263-266, 1990.