

論文要旨

本研究では、有限要素法における要素分割の自動化を目的として、有限要素をファジィ集合とし、そのパラメータの値を勾配法により算出するニューロ・ファジィ有限要素法を提案する。ニューロ・ファジィモデルには、ガウス基底の重み付き線形和を用いる。また、得られた近似解に対する評価方法の一つとして、ニューロ・ファジィ手法による誤差解析を提案する。ラプラス方程式を用いた数値例により提案手法の有効性を示す。

英文要旨

For approximating nonlinear mappings, especially interpolating the datapoints in a high dimensional space, RBF networks with adaptive learning algorithm are attracting a great deal of interest due to their rapid training, generality and simplicity. RBF network is interpreted as three-layered neural networks and/or fuzzy reasoning rules.

In this paper, we propose a neuro-fuzzy method to find a solution of partial differential equations. The elements in Finite Element Method (FEM) are replaced by fuzzy subsets defined by Gaussian membership functions, and the adaptive learning method is employed.

For almost all problems in practical engineering where FEM is applied, the analytical solutions are not obtained. Hence, the approximate solution of FEM needs some tools for error estimation. So a method of error estimation based on the neuro-fuzzy scheme is also proposed.

英文要旨訳

非線形写像の近似のために、適応的学習アルゴリズムを用いる RBF ネットワークは、高速な学習、汎化能力や簡便さに優れているため大きな関心を集めている。RBF ネットワークは3層ニューラルネットワークやファジィ推論ルールとも解釈されている。

本論文では、偏微分方程式の解を求めるニューロ・ファジィ手法を提案する。有限要素法 (FEM) における要素を、ガウシアンメンバーシップ関数によって定義されるファジィ集合に置き換え、適応学習法を採用する。

FEM が用いられる現実の工学におけるほとんどの問題は解析解が得られない。そのため有限要素法による近似解を評価する手法が必要である。そこで、ニューロ・ファジィ手法にもとづいた誤差推定手法もあわせて提案する。

1. はじめに

非線形写像を近似する問題は与えられたデータ点における近似誤差を表す評価関数の最小化問題として定式化することができる。シグモイド関数を基底関数とする多層型パーセプトロン[1]は任意の関数を近似できることが数学的に証明されていて、広く用いられている。評価関数の最小化は基底関数のパラメータを逐次的に調節するいわゆる学習により行われることが多い。多層型パーセプトロンの学習アルゴリズムには誤差逆伝播学習が用いられるが、十分な近似が得られるまでの学習時間の長さや、計算の複雑さ等のいくつかの問題点が指摘されている。

J.MoodyとC.Darken[2, 3]は入力空間の全領域で連続、かつ単峰性と回転対称性を持つ関数(Radial Basis Function (RBF))を基底関数とするネットワークを提案した。彼らはガウス基底の持つ局所性を活かした高速学習法を提案し、分類問題や時系列予測問題に適用して誤差逆伝播法によるニューラルネットワークの場合と比べて、数百倍～千倍程度の学習速度を持つことを報告している。このネットワークは localized receptive fields と呼ばれ、ニューラルネットワークモデルの一つと考えられる。T.PoggioとF.Girosi[4, 5]はデータが解を一意に決定できるだけ充分多く与えられていない場合に、滑らかさの制約条件を加えた評価関数を用いる正則化理論を適用することを提案し regularization network と呼んでいる。また、正則化から導かれる解が基底関数の線形和で表わされるという意味で3層のネットワークとして解釈できることを示した。さらに、彼らは代表的なRBFであるガウス基底を用いてRBF法と正則化から求められる解の類似性を示している。逆に、データ数が膨大な場合はRBF法および正則化はデータ数と同数の基底関数を必要とするために、多量のメモリーと計算時間を必要とし実用的ではない。T.Poggioらは従来のRBF法が基底関数の位置をデータ座標に固定してその重みのみをパラメータとしていたのに対し、基底関数の中心を可動にして中心座標をパラメータに加えることにより、より少ない基底関数で解を表現する方法を提案した。彼らはこの手法を Generalized RBF (GRBF) と呼んだ。GRBFにおいては、重みと基底の中心座標は逐次学習アルゴリズムを用いて計算され、彼らはその中で最も簡単な最急降下法を用いた場合の計算式を示している。RBFネットワークの出力は単峰形関数の荷重和で表現されるので、ネットワークをファジィ推論ルールとみなすことも可能である[6, 7, 8]。GRBFの応用研究における比較対象として誤差逆伝播を用いる多層型パーセプトロンがよく用いられるが、GRBFとそれらの類似性を指摘する論文も見られる[9]。

ガウス基底を用いたRBF法は様々な問題に応用されている。Y.Leeら[10]やK.Ngら[11]は、音声認識問題やパターン分類問題におけるRBF法とその他の多くの分類手法での分類誤差率の比較を行った。この比較の結果、十分な数のトレーニングデータと注意深いパラメータの調整が保証されるなら、どの手法を用いても同程度の分類誤差率が得られることを報告している。さらに、彼らはトレーニング時間やパラメータ数等の特性に関して各種の方法を比較しており、データの少ない領域や高次元空間におけるRBF法の良好な汎化能力を確認している。T.PoggioとS.Edelman[12]は3次元物体の各頂点の平行投影像を入力とするGRBFを用いて3次元物体の認識を行っている。Spechtらによる Probabilistic network [13]に始まるRBF法は、この他にもニューラルネットワークやファジィ理論の応用分野で、手書き文字認識問題[14]、ロボット制御[15]、断層撮影法[16]などに幅広く用いられている。本論文では適応的学習法を用いるRBF法をニューロ・ファジィと呼び、その偏微分方程式の近似解法への応用について述べる。

有限要素法[17]の基本的な考え方は、未知な連続関数を線分や平面に代表される区分的多項式群で近似的に置き換えて表現しようとするものである。有限要素法は微分方程式、特に偏微分方程式の第1種境界値問題(ディリクレ問題)および第2種境界値問題(ノイマン問題)の有力な近似解法として用いられている。近似解法であるので、解の近似精度が問題となるが近似精度をあげるには要素数を増加させねばならず、領域を要素化するためのデータ作成に多くの時間が必要となる。データの作成時間の削減策として自動要素分割やアダプティブ・メッシングなどが提案されている[18]・[19]。本研究では要素分割の自動化を目的として、有限要素をファジィ集合とみなしてメンバーシップ関数にガウス基底を導入する方法につ

いて述べる．三角形要素は有限要素法の中で最も多く実用化されているもので，自動分割法の対象もこの手法である場合が多い．これは，領域を任意の節点を持つ2次元三角形要素で分割し境界条件を利用してマトリックス計算を行うことにより近似解を求めるものである．本研究では有限要素をファジィ集合とする．そのメンバーシップ関数をガウス基底とし，その中心座標と分散（幅）を自動調整することによって要素の自動ファジィ分割を行う．ファジィ分割は，ファジィ制御やファジィモデリングではファジィパーティションと呼ばれ，領域をファジィ集合によってあいまいに分割することを意味している．分散の小さなガウス基底が多く集まった部分は領域を小さな要素で細分化していることになる．本論文での提案法は，選点法，重み付き残差法によるものである．変分法に基づくレーリーリッツ法よりも精度が良くかつ簡便であると考えられる[20]．解を近似するファジィモデルはガウス基底（釣鐘型メンバーシップ関数）の線形和でありRBFネットワークに等しい．

また，得られた近似解に対する評価方法の1つとして，誤差解析の手法を提案する．一般に，平面応力問題や熱伝導問題などの工学に現れる実際の問題を解析的に解くことはほとんど不可能であり，解析解は未知である．したがって有限要素法などの近似解法を設計に用いるときに問題となるのが解の信頼性である．解析解が未知であるため，どの程度精度のよい解が得られているかを知るためには，誤差を推定する必要がある．提案する誤差解析法もニューロ・ファジィ手法を用いるもので，近似解法で得られる評価関数の値をもとに誤差を推定するものである．

2. ファジィモデルとRBFネットワーク

$A_{i,k}(x_i)$ を第 i 入力空間($i = 1, 2$)における第 k ファジィルールのメンバーシップ関数とすると第 k ファジィルールは次のように書ける．

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

三角型メンバーシップ関数を用いる簡略ファジィ推論モデルは，1階微分が不連続な区分線形に近い関数であるためにニューラルネットワークモデルのような滑らかさに欠けている[7]．そこで，ファジィモデルを C^∞ 級の滑らかな関数で構成するために次のガウス基底をメンバーシップ関数とする．

$$A_{i,k}(x_i) = \exp\left(-\frac{(x_i - a_{i,k})^2}{b_{i,k}}\right) \quad (1)$$

ただし， k はルール番号で $a_{i,k}$ と $b_{i,k}$ はルール毎に設定するものとする．また，第 k ルールの条件部適合度は次式のようになる．

$$\mu_k = A_{1,k}(x_1) \cdot A_{2,k}(x_2) \quad (2)$$

第 k ルールの結論部実数値を w_k とすると，ファジィモデルの出力 y は

$$y = \sum_k w_k \cdot \mu_k \quad (3)$$

となる．(3)式はRBFネットワークであり，3層のニューラルネットワークともみなされている[2] - [5]．Fig. 1にその概念図を示す．また，評価関数をネットワークの出力 y と理想出力 y^* との差の2乗

$$E = \frac{1}{2}(y - y^*)^2 \quad (4)$$

とすると，評価関数 E を最小とするような結論部 w_k および条件部の $a_{i,k}$ ， $b_{i,k}$ の各パラメータは，次式で表される勾配法にもとづくLMS学習則によって求めることができる．

$$w_k^{NEW} = w_k^{OLD} - \alpha \cdot \mu_k \cdot \delta \quad (5)$$

$$a_{i,k}^{NEW} = a_{i,k}^{OLD} - \beta \cdot \mu_k \cdot \delta_{a_{i,k}} \quad (6)$$

$$b_{i,k}^{NEW} = b_{i,k}^{OLD} - \gamma \cdot \mu_k \cdot \delta_{b_{i,k}} \quad (7)$$

ただし、 δ はモデルの出力 y と理想出力 y^* の差

$$\delta = y - y^* \quad (8)$$

であり、

$$\delta_{a_{i,k}} = \frac{2(x_i - a_{i,k})}{b_{i,k}} \cdot w_k \cdot \delta \quad (9)$$

$$\delta_{b_{i,k}} = \left\{ \frac{(x_i - a_{i,k})}{b_{i,k}} \right\}^2 \cdot w_k \cdot \delta \quad (10)$$

である。また、 α, β, γ はそれぞれ $w_k, a_{i,k}, b_{i,k}$ に対する学習係数であり正の値である。

< (Fig.1)

3. 有限要素法への応用

Fig. 1に示すような3層のネットワークの出力は (3) 式で表されるように中間層の各ユニットの出力の荷重和であり、各ユニットに対して1つのガウス基底関数が割り当てられている。従って、近似関数 $y(\mathbf{x})$ は \mathbf{x} に関して任意回数の微分が可能で、任意の微分方程式および境界条件を近似的に表現することができる。本論文では次式で表される2次元のラプラス方程式を例に提案手法の説明を行う。

$$\nabla^2 f(x_1, x_2) = 0 \quad \text{on } R \quad (11)$$

R は任意の対象領域で、 R の境界上で次式のような境界条件を持つ。

$$f(x_1, x_2) = g(x_1, x_2) \quad \text{on } B_1 \quad (12)$$

$$\frac{\partial f(x_1, x_2)}{\partial \mathbf{n}} = h(x_1, x_2) \quad \text{on } B_2 \quad (13)$$

ここで、 $B=B_1 \cup B_2$ は R の境界を示す。また、 \mathbf{n} は B_2 上の各点における法線方向のベクトルを表す。このラプラス方程式をニューロ・ファジィモデルの出力 $y(x_1, x_2)$ を用いて次式のように近似表現する。

$$\nabla^2 y(x_1, x_2) \approx 0 \quad \text{on } R \quad (14)$$

$$y(x_1, x_2) \approx g(x_1, x_2) \quad \text{on } B_1 \quad (15)$$

$$\frac{\partial y(x_1, x_2)}{\partial \mathbf{n}} \approx h(x_1, x_2) \quad \text{on } B_2 \quad (16)$$

たとえば、(14) 式における $\partial^2 y / \partial x_i^2$ は次式のようにになる。

$$\frac{\partial^2 y}{\partial x_i^2} = \sum_k \left(\frac{4(x_i - a_{i,k})^2 - 2b_{i,k}^2}{b_{i,k}^3} \cdot w_k \cdot \mu_k \right) \quad (i = 1, 2) \quad (17)$$

そして、微分方程式の近似に関する誤差と境界条件の近似に対する誤差の自乗和を評価関数 $E(\mathbf{x})$ として、この値の減少方向にパラメータ $w_k, a_{i,k}, b_{i,k}$ を変化させる自乗誤差最小化問題として定式化する。(14) 式、(15) 式、(16) 式の左辺をそれぞれ $F(\mathbf{x}), G(\mathbf{x}), H(\mathbf{x})$ とおき、評価関数を次式のようにその自乗和とする。

$$\begin{aligned} E(\mathbf{x}) &= \frac{1}{2} \cdot \{F(\mathbf{x})^2 + K_1 \cdot (G(\mathbf{x}) - g(\mathbf{x}))^2 \\ &\quad + K_2 \cdot (H(\mathbf{x}) - h(\mathbf{x}))^2\} \end{aligned} \quad (18)$$

ここで、 K_1 と K_2 は各項（誤差）のバランスをとるために用いる正の定数である．ネットワークの学習法は次式で表される簡単な勾配法であり，データ点（選点）毎に行う LMS（Least Mean Square）やデルタルールに相当する．

$$w_k^{NEW} = w_k^{OLD} - \alpha \cdot \delta_{w_k} \quad (19)$$

$$a_{i,k}^{NEW} = a_{i,k}^{OLD} - \beta \cdot \delta_{a_{i,k}} \quad (20)$$

$$b_{i,k}^{NEW} = b_{i,k}^{OLD} - \gamma \cdot \delta_{b_{i,k}} \quad (21)$$

ここでの α , β , γ はそれぞれ w_k , $a_{i,k}$, $b_{i,k}$ に対する学習係数を示し, δ_{w_k} , $\delta_{a_{i,k}}$, $\delta_{b_{i,k}}$ はそれぞれ

$$\begin{aligned} \delta_C &= F(\mathbf{x}) \cdot \frac{\partial F(\mathbf{x})}{\partial C} \\ &+ K_1 \cdot (G(\mathbf{x}) - g(\mathbf{x})) \cdot \frac{\partial G(\mathbf{x})}{\partial C} \\ &+ K_2 \cdot (H(\mathbf{x}) - h(\mathbf{x})) \cdot \frac{\partial H(\mathbf{x})}{\partial C} \\ C &\in \{w_k, a_{i,k}, b_{i,k}\} \end{aligned} \quad (22)$$

である．例えば, (22) 式における $\partial F(\mathbf{x})/\partial C$ は

$$\frac{\partial F(\mathbf{x})}{\partial w_k} = \sum_{i=1}^2 \left(\frac{4(x_i - a_{i,k})^2 - 2b_{i,k}}{b_{i,k}^2} \right) \cdot \mu_k \quad (23)$$

$$\begin{aligned} \frac{\partial F(\mathbf{x})}{\partial a_{i,k}} &= \left(\frac{8(x_i - a_{i,k})^3 - 12(x_i - a_{i,k})b_{i,k}}{b_{i,k}^3} \right. \\ &+ \frac{8(x_i - a_{i,k})(x_j - a_{j,k})^2}{b_{i,k}b_{j,k}^2} \\ &\left. - \frac{4(x_i - a_{i,k})}{b_{i,k}b_{j,k}} \right) \cdot \mu_k \cdot w_k \end{aligned} \quad (24)$$

$$\begin{aligned} \frac{\partial F(\mathbf{x})}{\partial b_{i,k}} &= \left(\frac{4(x_i - a_{i,k})^4 - 10(x_i - a_{i,k})^2b_{i,k}}{b_{i,k}^4} \right. \\ &+ \frac{2}{b_{i,k}^2} + \frac{4(x_i - a_{i,k})^2(x_j - a_{j,k})^2}{b_{i,k}^2b_{j,k}} \\ &\left. - \frac{2(x_i - a_{i,k})}{b_{i,k}} \right) \cdot \mu_k \cdot w_k \end{aligned} \quad (25)$$

である．

<加筆修正>

本提案法は領域上に均等に与えられたデータ点での2乗誤差の最小化問題の解法であるので，LMS 法としてではなく学習データをバッチ的に与えて次元探索を行う共役勾配法等を用いることができる [20]．しかし，アルゴリズムが複雑になることや次元探索では必ず初期値の近くの局所最小値が求まる等の欠点もあり，ヒューマンフレンドリーなソフトコンピューティングとなるように簡便さと分かりやすさを優先した LMS 学習法を採用している．

4. 誤差解析

有限要素法では近似解の誤差解析および要素の再分割のための手法としてアダプティブ・メッシングが提案されている．本章では前章で提案した近似解法を用いて求められた近似解に対する誤差解析の手法を提案する．なお，ネットワークおよび基底関数は偏微分方程式の近似解を求める時と同様のものを使用する．また，本章でもラプラス方程式を用いて説明を行う．

前章ではラプラス方程式 ((11) 式) とその境界条件 ((12) 式, (13) 式) を (14) 式, (15) 式, (16) 式のように近似表現したが, (14) 式, (15) 式, (16) 式の学習後の近似誤差はそれぞれ次式のように表される.

$$\begin{aligned} E_1^{after} &= \nabla^2 f(x_1, x_2) - \nabla^2 y(x_1, x_2) \\ &= \nabla^2 \{f(x_1, x_2) - y(x_1, x_2)\} \\ &= \nabla^2 e(x_1, x_2) \quad \text{on } R \end{aligned} \quad (26)$$

および,

$$\begin{aligned} E_2^{after} &= f(x_1, x_2) - y(x_1, x_2) \\ &= e(x_1, x_2) \quad \text{on } B_1 \end{aligned} \quad (27)$$

$$\begin{aligned} E_3^{after} &= \frac{\partial f(x_1, x_2)}{\partial \mathbf{n}} - \frac{\partial y(x_1, x_2)}{\partial \mathbf{n}} \\ &= \frac{\partial \{f(x_1, x_2) - y(x_1, x_2)\}}{\partial \mathbf{n}} \\ &= \frac{\partial e(x_1, x_2)}{\partial \mathbf{n}} \quad \text{on } B_2 \end{aligned} \quad (28)$$

ただし, E_1^{after} , E_2^{after} , E_3^{after} は学習後のラプラス方程式とその境界条件の近似誤差であり, $f(x_1, x_2)$ は解析解で $e(x_1, x_2)$ は解析解と近似解の誤差を表す関数である. 誤差解析は上記の誤差関数 $e(x_1, x_2)$ を推定することである. (26) 式, (27) 式, (28) 式は前章と同様に次式で近似表現できる.

$$\nabla^2 y_e(x_1, x_2) \approx E_1^{after} \quad \text{on } R \quad (29)$$

$$y_e(x_1, x_2) \approx E_2^{after} \quad \text{on } B_1 \quad (30)$$

$$\frac{\partial y_e(x_1, x_2)}{\partial \mathbf{n}} \approx E_3^{after} \quad \text{on } B_2 \quad (31)$$

ここで, $y_e(x_1, x_2)$ は誤差解析用のニューロ・ファジィモデルの出力で, その概念図は Fig. 1 に示すものと同一のものである. そして, (29) 式, (30) 式, (31) 式の左辺をそれぞれ $F(\mathbf{x})$, $G(\mathbf{x})$, $H(\mathbf{x})$ とすると, 評価関数は次式となる.

$$\begin{aligned} E(\mathbf{x}) &= \sum_{\mathbf{x} \in D} \left(\frac{1}{2} \cdot \{(F(\mathbf{x}) - E_1^{after})^2 \right. \\ &\quad + K_3 \cdot (G(\mathbf{x}) - E_2^{after})^2 \\ &\quad \left. + K_4 \cdot (H(\mathbf{x}) - E_3^{after})^2 \} \right) \end{aligned} \quad (32)$$

K_3 , K_4 は K_1 , K_2 と同様, 各項のバランスをとるために用いる正の定数である. 各パラメータの学習法は前章と同じであり, その学習則も (21) 式で示したとおりである. ただし, δ_{w_k} , $\delta_{a_{i,k}}$, $\delta_{b_{i,k}}$ は

$$\begin{aligned} \delta_C &= (F(\mathbf{x}) - E_1^{after}) \cdot \frac{\partial F(\mathbf{x})}{\partial C} \\ &\quad + K_3 \cdot (G(\mathbf{x}) - E_2^{after}) \cdot \frac{\partial G(\mathbf{x})}{\partial C} \\ &\quad + K_4 \cdot (H(\mathbf{x}) - E_3^{after}) \cdot \frac{\partial H(\mathbf{x})}{\partial C} \\ C &\in \{w_k, a_{i,k}, b_{i,k}\} \end{aligned} \quad (33)$$

である。手順としては、3章で提案した近似解法で近似解を求めた後、各データ点において近似誤差 $E_1^{after} \sim E_3^{after}$ を求める。そして、本章で提案した誤差解析手法で誤差関数 $e(x_1, x_2)$ を推定する。 ＜加筆修訂＞

アダプティブメッシング法では、一旦有限要素法による解を求め、次に誤差を推定し、その後にメッシュ分割の細分化を行うのに対して、提案法では第1段階でパラメータ a, b の変更によるファジィなメッシュ分割の最適化が行われる。したがって第2段階の誤差解析は、本来誤差を推定し評価するものであるが、ガウス基底の数を変更する場合の目安に用いることもできる。アダプティブメッシング法が通常の三角形要素と変分法による定式化であるのに対して、提案する誤差解析法は第1段階の解法と一貫性のある RBF ネットワークによるアダプティブな手法である。

5. 数値例

5.1 提案手法による偏微分方程式の解の近似

提案手法を次の2つの場合の合計5種類の偏微分方程式に適用し、その有効性を検討する。

Case1：解析解が既知の問題

- (1) ディリクレ条件をもつラプラス方程式
- (2) ノイマン条件をもつラプラス方程式

Case2：解析解が未知の問題

- (1) } 人為的に与えた不連続な境界条件を持つ
- (2) } ラプラス方程式
- (3) }

各基底の初期位置は、Fig. 2に示すように均等に配置する。また、データ点も同様に均等に配置する。 $w_k(k = 1, \dots, N)$ の初期値は0.0, $b_k(k = 1, \dots, N)$ は0.2 とする。提案手法によるシミュレーション結果をFig. 3, 5, 7~9に示す。たとえば、Fig. 3の場合、データ点の数は x_1 軸上で41個、 x_2 軸上で51個、各軸に平行する境界にも同数のデータ点がある。また、内部では $39 \times 49 = 1911$ 個としているので、この比率では内部に対する重みを1 とすると境界条件に対する重みは20程度となる。領域内部よりもまず境界条件を合わせる必要があると考え、 K_1, K_2 の値はさらにその5倍の100とした。Fig. 4およびFig. 6にはCase1の(1)および(2)における解析解と近似解の誤差をそれぞれ示す。また、すべての問題の評価関数の値をTable 1に示す。Table 1における () 内の数値は、境界条件における誤差を示している。なお、シミュレーションに用いた学習係数 α, β, γ の値をTable 2に示す。これらの学習係数は、評価関数が発散しない値の内できるだけ大きな値をとった。 ＜加筆修訂＞

Case1

$$(1) \nabla^2 f(x_1, x_2) = 0 \quad (-1.0 \leq x_1 \leq 1.0, \quad 0 \leq x_2 \leq \pi) \quad (34)$$

境界条件

$$f(x_1, x_2) = \begin{cases} 0 & ; x_2 = 0 \text{ or } x_2 = \pi \\ \cosh 1.0 \sin x_2 & ; x_1 = -1.0 \text{ or } x_1 = 1.0 \end{cases} \quad (35)$$

$$(2) \nabla^2 f(x_1, x_2) = 0 \quad (0 \leq x_1 \leq 1.0, \quad 0 \leq x_2 \leq 1.0) \quad (36)$$

境界条件

$$f(x_1, x_2) = 0 \quad (x_2 = 0 \text{ or } x_2 = 1.0) \quad (37)$$

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = \begin{cases} 0 & ; x_1 = 0 \\ \sin \pi x_2 & ; x_1 = 1.0 \end{cases} \quad (38)$$

Case1 の (1), (2) とともに, Fig. 3と Fig. 5に示すように, 解をほぼ正確に近似しているが, Fig. 4と Fig. 6を比較してみると (1) のほうが近似精度は良いと言える. (2) の誤差は主に境界条件の第2式の近似 ((16) 式) より発生したものであり, 微分方程式および境界条件第1式の近似 ((14), (15) 式) がその影響を受けている.

< (Fig.2)

Case2

$$(1)\nabla^2 f(x_1, x_2) = 0 \quad (0 \leq x_1 \leq 1.0, \quad 0 \leq x_2 \leq 1.0) \quad (39)$$

境界条件

$$f(x_1, x_2) = \begin{cases} 1.0 & ; x_2 = 0 \\ 0.0 & ; \text{その他} \end{cases} \quad (40)$$

$$(2)\nabla^2 f(x_1, x_2) = 0 \quad (0 \leq x_1 \leq 1.0, \quad 0 \leq x_2 \leq 1.0) \quad (41)$$

境界条件

$$f(x_1, x_2) = \begin{cases} 1.0 & ; x_1 \in [0.1, 0.3] \cup [0.7, 0.9], x_2 = 0 \\ 0.0 & ; \text{その他} \end{cases} \quad (42)$$

$$(3)\nabla^2 f(x_1, x_2) = 0 \quad (0 \leq x_1 \leq 1.0, \quad 0 \leq x_2 \leq 1.0) \quad (43)$$

境界条件

$$f(x_1, x_2) = \begin{cases} 1.0 & ; x_1 \in [0, 0.5], x_2 = 0 \\ 0.0 & ; \text{その他} \end{cases} \quad (44)$$

Fig. 7から Fig. 9における $x_2 = 0.0$ 上の太線はそれぞれ境界条件を表している. Fig. 7から Fig. 9を比較してみると, (1), (3) に比較して (2) の近似精度があまり良くない. これは2つの山に対してその極めて近くにある基底のみが反応するためで, この境界条件を充分に表現するためには基底が不足している. これはガウス基底のもつ局所性が近似結果に悪影響を及ぼすという典型的な例である. また, Table 1から明らかに, 本提案手法における評価関数の学習後の値の大部分は第1項 (微分方程式の近似誤差) が占めていて, 評価関数の値から近似精度を評価するのは困難である. (1) ~ (3) の結果から, 十分な数の基底を使用すれば不連続な境界条件を持つ偏微分方程式に対しても, 精度の良い解が計算時間とのトレードオフとして得られると考えられるが, どの程度の数にすべきかを決定するためには誤差解析が必要になる.

< (Fig.3),
(Fig.5),
(Fig.6),
(Fig.8),
(Fig.9),
(Table 2)

5.2 提案手法による誤差解析

4章で提案した誤差解析手法をCase1の2つのラプラス方程式に適用してその有効性を検討する。ただし、1000回の学習後（Fig. 3およびFig. 5の状態）に対して誤差解析手法を適用する。学習のシミュレーション結果をFig. 10およびFig. 11に示す。なお、シミュレーションに用いた学習係数 α 、 β 、 γ の値をTable 3に示す。これらの学習係数は、4章の数値例と同様に、評価関数が発散しない値の内できるだけ大きな値を用いた。

Fig. 4に示す解析解と近似解の誤差には領域内部の左側に大きなくぼみがあり、そこでの誤差が大きいことを示している。Fig. 10は解析解が未知であるとして提案手法により誤差を推定したもので、領域内部の大きなくぼみが復元できている。また、Fig. 6は誤差が領域の両端に存在している様子を示しており、Fig. 11はこれを比較的良好に復元している。Fig. 3の近似解を求めるときと同様にデータ点数を考慮すると境界条件に対する重み K_3 、 K_4 も100程度となる。しかし、一般に、ディリクレ条件に対する境界条件の近似誤差は容易に求めることができるので、領域内部の誤差の推定が重要である。したがって、ここでの誤差解析には評価関数（(32)式）の重みの値を比較的小さくし（ $K_3 = K_4 = 30$ ）、領域内部の誤差の推定を優先した。

6. おわりに

三角型の基底関数（形状関数）の荷重和によって偏微分方程式の解を近似しようとする有限要素法の基本的な考え方は、ニューロ・ファジィモデルの出力が各基底関数の荷重和で表現されるという点で類似している。本研究ではニューロ・ファジィモデルを用いる偏微分方程式の近似解法を提案し、これを種々の境界条件をもつラプラス方程式に適用してその有効性を検討した。提案手法は逐次的にパラメータの調整を行うので他の手法と比較して計算のためのアルゴリズムが簡単である。基底の数を増やすことにより、不連続な境界条件をもつ問題に対しても充分に対応できると考えられる。しかし、学習係数や評価関数中の係数は試行錯誤的に設定する必要がある。これらのパラメータの値により計算結果が変わってくるという問題点も有している。そこで、得られた近似解の近似誤差を推定する手法もあわせて提案し、ある程度の誤差推定が可能であることを示した。

参考文献

- [1] D.E.Rumelhart, J.L.McClelland and the PDP Research Group: Parallel Distributed Processing, Cambridge, MIT Press (1987)
- [2] J.Moody and C.Darken: Learnng with localized receptive fields; Proc. of 1988 Connectionist Models Summer School, Morgan-Kaufmann, pp. 133~143 (1988)
- [3] J.Moody and C.Darken: Fast Learning in Networks of Locally-Tuned Processing Uints; Neural Computation, Vol.1, No.2, pp.281~294 (1989)
- [4] 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)
- [5] T.Poggio and F.Girosi: Regularization Algorithms for Learning That are Equivalent to Multilayer Networks; Sciences, Vol.247, pp.978~982 (1990)
- [6] L.X.Wang and J.M.Mendel: On Fuzzy Basis Functions, Universal Approximation and Orthogonal Least-Squares Learning; IEEE Trans. on Neural Networks, Vol.3, No.5, pp.807~814 (1992)

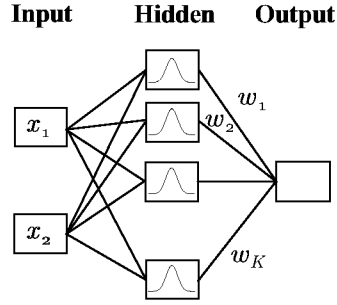
- [7] 市橋: ファジィ制御とモデリング; システム制御情報学会誌, Vol.37, No.1, pp.30~37 (1993)
- [8] H.Ichihashi and I.B.Turksen: A Neuro-Fuzzy Approach to Data Analysis of Pairwise Comparisons; Int.J. of Approximate Reasoning , Vol.9, No.3, pp.227~248 (1993)
- [9] B.E.Segee, M.J.Carter: Comparative Fault Tolerance of Generalized Radial Basis Function and Multilayer Perceptron Networks; Proc. of the 1993 IEEE International Conference on Neural Networks, pp.1847~1852 (1993)
- [10] Y.Lee and R.Lippmann: Practical Characteristics of Neural Networks and Conventional Pattern Classifiers on Artificial and Speech Problems; Advances in Neural Information Processing Systems 2, pp.168~177 (1990)
- [11] K.Ng and R.Lippmann: A Comparative Study of the Practical Characteristics of Neural Networks and Conventional Pattern Classifiers; Advances in Neural Information Processing Systems 3, pp.970~976 (1991)
- [12] T.Poggio and S.Edelman: A Network that Learns to Recognize 3D Objects; Nature, Vol.343, pp.263~266 (1990)
- [13] D.F.Specht : Probabilistic Neural Networks for Classification, Mapping, or Associative Memory; Proc. of the 1988 IEEE Int. Conf. on Neural Networks, Vol.1, pp.525~532 (1988)
- [14] Y.Lee: Handwritten Recognition Using k Nearest-Neighbor: Radial Basis Function, and Backpropagation Neural Networks; Neural Computation, Vol.1, No.3, pp.440~449 (1991)
- [15] H.Ichihashi: Learning Inverse Dynamics Model of a Manipulator in a Hierarchical Fuzzy Model; Proc. of IMACS/SICE RM²S'92, Kobe, pp.41~46 (1992)
- [16] 市橋, 三好, 長坂, 木村: ニューロ・ファジィ逆解析による断層撮影法; 日本ファジィ学会誌, Vol.6, No.2, pp.333~341 (1994)
- [17] 菊地, 岡部: 有限要素システム入門, 日科技連 (1986)
- [18] 尾田, 山崎 : 有限要素解析のための自動分割技法の現状とその利用法 (1) , (2) , (3) ; 機械の研究, Vol.37, No.6, pp.704~708, Vol.37, No.7, pp.835~840, Vol.37, No.8, pp.935~939(1985)
- [19] 大坪: 有限要素法の高精度誤差解析とその利用法, コンピュートロール, コロナ社, No.42, pp.3~9 (1993)
- [20] 佐藤, 市橋: ファジィ有限要素法; 第3回インテリジェントFA シンポジウム講演論文集, pp.119~122 (1991)

図 目 次

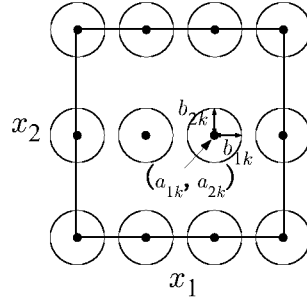
1	A neural network with Gaussian hidden unit	11
2	Initial positions of Gaussian functions. 4×3 Gaussian functions are used.	11
3	Approximate solution of the Laplace equation with the Dirichlet condition, after 1000 learning iterations(the number of Gaussian units 3×6 , the number of datapoints 41×51 , $K_1 = K_2 = 100$).	11
4	Errors between the approximate solution and the analytical solution (Case 1-(1))	12
5	Approximate solution of the Laplace equation with the Neumann condition, after 1000 learning iterations(the number of Gaussian units 3×6 , the number of datapoints 41×51 , $K_1 = K_2 = 100$).	12
6	Errors between the approximate solution and the analytical solution (Case 1-(2))	12
7	Approximate solution of the Laplace equation with the boundary condition of Eq.(40), after 5000 learning iterations(the number of Gaussian units 3×6 , the number of datapoints 41×51 , $K_1 = K_2 = 100$).	13
8	Approximate solution of the Laplace equation with the boundary condition of Eq.(42), after 5000 learning iterations(the number of Gaussian units 3×6 , the number of datapoints 41×51 , $K_1 = K_2 = 100$).	13
9	Approximate solution of the Laplace equation with the boundary condition of Eq.(44), after 5000 learning iterations(the number of Gaussian units 3×6 , the number of datapoints 41×51 , $K_1 = K_2 = 100$).	13
10	The estimated error in Case1-(1) after 1000 learning iterations(the number of Gaussian units 3×6 , the number of datapoints 41×51).	14
11	The estimated error in Case1-(1) after 1000 learning iterations(the number of Gaussian units 3×6 , the number of datapoints 41×51).	15

表 目 次

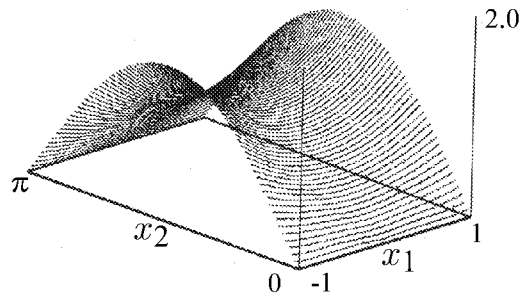
1	The change of values of the cost function as learning proceeds	14
2	The values of learning rates α, β, γ	14
3	The values of learning rates α, β, γ in the error estimation	14



⊠ 1: A neural network with Gaussian hidden unit



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



⊠ 3: Approximate solution of the Laplace equation with the Dirichlet condition, after 1000 learning iterations (the number of Gaussian units 3×6 , the number of datapoints 41×51 , $K_1 = K_2 = 100$).

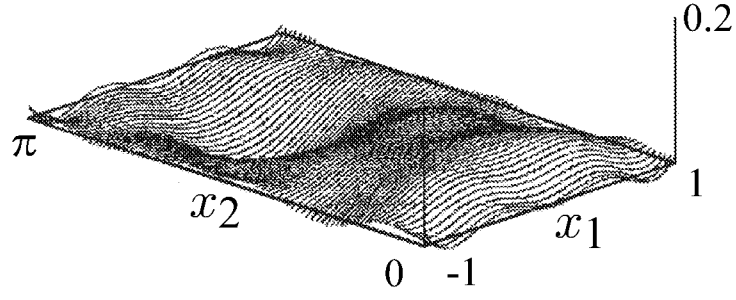


图 4: Errors between the approximate solution and the analytical solution(Case 1-(1)).

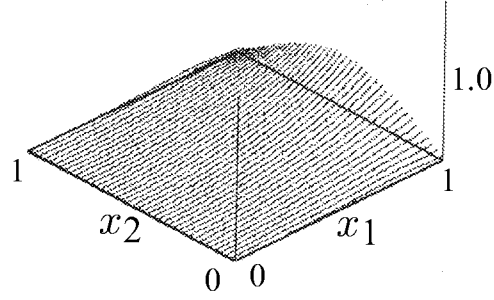


图 5: Approximate solution of the Laplace equation with the Neumann condition, after 1000 learning iterations(the number of Gaussian units 3×6 , the number of datapoints 41×51 , $K_1 = K_2 = 100$).

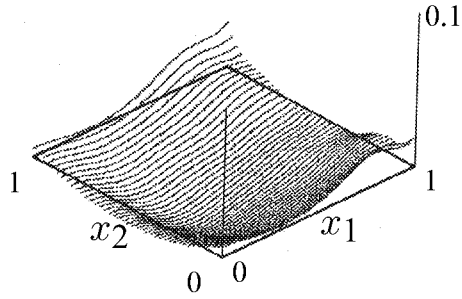


图 6: Errors between the approximate solution and the analytical solution(Case 1-(2)).

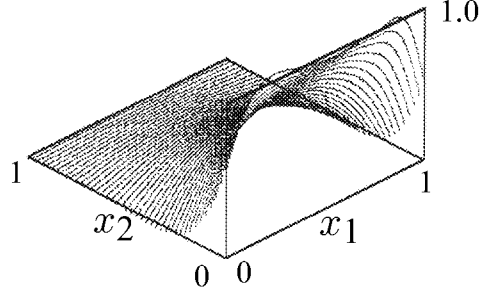


图 7: Approximate solution of the Laplace equation with the boundary condition of Eq.(40), after 5000 learning iterations(the number of Gaussian units 6×3 , the number of datapoints 21×11 , $K_1 = K_2 = 30$).

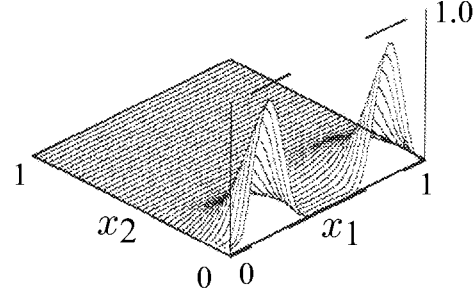


图 8: Approximate solution of the Laplace equation with the boundary condition of Eq.(42), after 5000 learning iterations(the number of Gaussian units 6×3 , the number of datapoints 21×11 , $K_1 = K_2 = 30$).

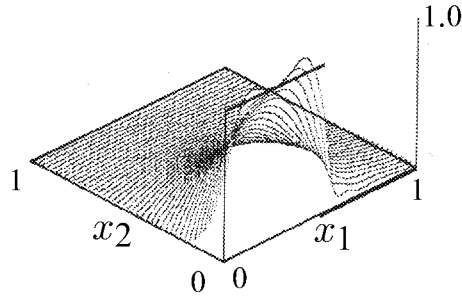


图 9: Approximate solution of the Laplace equation with the boundary condition of Eq.(44), after 5000 learning iterations(the number of Gaussian units 6×3 , the number of datapoints 21×11 , $K_1 = K_2 = 30$).

表 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.

表 2: The values of learning rates α, β, γ

	α	β	γ
Case1-(1)	10^{-5}	10^{-3}	10^{-3}
Case1-(2)	10^{-6}	10^{-6}	10^{-6}
Case2-(1)	10^{-7}	10^{-7}	10^{-7}
Case2-(2)	10^{-7}	10^{-7}	10^{-7}
Case2-(3)	10^{-6}	10^{-6}	10^{-6}

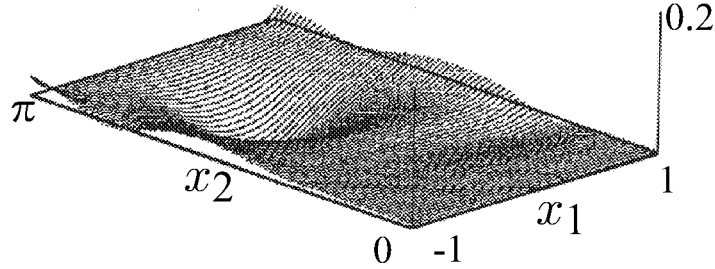
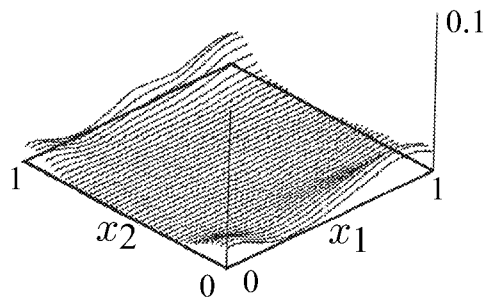


图 10: The estimated error in Case1-(1) after 1000 learning iterations(the number of Gaussian units 3×6 , the number of datapoints 41×51).

表 3: The values of learning rates α, β, γ in the error estimation

	α	β	γ
Case1-(1) error estimation	10^{-5}	10^{-3}	10^{-3}
Case1-(2) error estimation	10^{-5}	10^{-3}	10^{-3}



11: The estimated error in Case1-(2) after 1000 learning iterations(the number of Gaussian units 3×6 , the number of datapoints 41×51).