

## RBPNのパターン認識への応用

韓, 敏

大連理工大学電子情報工程学院 | 九州大学大学院システム情報科学研究所 : 訪問助教授

金江, 春植

九州大学大学院システム情報科学研究所電気電子システム工学部門

和田, 清

九州大学大学院システム情報科学研究所電気電子システム工学部門

<https://doi.org/10.15017/1515820>

---

出版情報 : 九州大学大学院システム情報科学紀要. 8 (1), pp.67-72, 2003-03-26. 九州大学大学院システム情報科学研究所

バージョン :

権利関係 :

## RBPNのパターン認識への応用

韓 敏\* · 金江春植\*\* · 和田清\*\*

### Applications of RBPN for Pattern Recognition

Min HAN, Shunshoku KANAE and Kiyoshi WADA

(Received December 20, 2002)

**Abstract:** Based on radial basis function neural network (RBFN) and perceptron neural network, this paper built a new four-layer feed-forward neural network named radial basis perceptron network (RBPN). This network can be summarized as follows: (1) It is selective connection between the units of two hidden layers; (2) The number of units of hidden layers is to be defined dynamically; (3) During learning procedure, RBPN adopts input-output clustering (IOC) method, and the appearance parameters of centers is self-adjustable. This is illustrated using an example taken from applications for component analysis of civil building materials. Simulation shows that RBPN can be used to predict the components of civil building materials successfully and gets good generalization ability.

**Keywords:** Artificial neural network, Pattern recognition, IOC, RBPN, Civil building materials

#### 1. はじめに

ANN (Artificial Neural Network) は複雑環境や多目的の制御に適した強い自己学習能力を持っているため、非線形システムや非確定システムの同定やパターン認識等に広く応用されている<sup>1),2)</sup>。ANNの基本式は次のように表される。

$$\hat{y} = ANN[x_1, x_2, \dots, x_n] \quad (1)$$

ここで、 $x = (x_1, x_2, \dots, x_n)^T \in R^n$  は入力ベクトル、 $y = (y_1, y_2, \dots, y_L)^T \in R^L$  は実際の出力、 $\hat{y} \in R^L$  はANNの出力である。すなわちANNはシステムの入出力データを用いて特徴抽出と統計分類の機能を実現し、システムの同定を行っている。

パーセプトロンネットワークはF. Rosenblatt<sup>3)</sup>が1950年代末に提案した一種のニューラルネットワークであり、連続関数を近似する最も簡単なネットワークである。これはサンプル空間を超平面で分割するので、一種の統計分類器とも考えられる。ただし、実際の問題としてサンプル空間が必ず線形分割できるとは限らない。BPN (Back Propagation Network) は一種の教師付き学習ネットワークであり、最急降下法に基づいた最適化ストラテジーを取るため、局所的な最適化問題があるし、また、収束速度が遅いという問題もある。

他の最適化手法、例えば遺伝アルゴリズム<sup>4)</sup>、パター

ンアニーリング<sup>5),6)</sup>等は大域的な最小値を獲得することができるが、計算量が多く、効率が低いという問題がある。RBFN (Radial Basis Function Network) はある程度上記の問題を避けることができる。これはニューロンの非線形変換関数のパラメータの調整により、多くの異なる非線形写像を実現し、これらの線形結合の重みを調整することにより関数近似を行うものである。大域的な収束が保証できると同時に学習速度も高く注目されている。Giroso と Poggio<sup>7)</sup> は、RBFN は最適な近似能力があるが、BPN はそのような能力を持っていないことを指摘した。Monica Bianchini<sup>8)</sup> はRBFNの大域的な収束性を証明した。現在RBFNの学習計算法に関する研究成果は、多くの文献で報告されている<sup>9),10),11),12),13)</sup>。ただし、RBFNで任意の連続関数を近似するには、隠れ層に多くのユニット数を必要とする。Powellが最初に提案した多変数インターポレーションRBF法では、すべての学習データサンプルを利用することが要求されたが、ニューラルネットワークの汎化能力を高めるためには、通常大量の学習サンプルが必要であり、隠れ層ユニットの数が膨大になって、ネットワークの性能とネットワークの構造の間に矛盾が出てくる。そこで、BroomheadとLowe<sup>14)</sup>はあるサンプル数より少ないニューロンの数でネットワークを構成する方法を提案した。ただし、RBFNの利点を保持すると同時にいかに隠れ層ユニット数を減少させて、ネットワークの構造を簡単にすることが一つの重要な課題となっている。Chester<sup>15)</sup>は隠れ層が二層の場合は、一層の場合と比較してより良い近似精度と汎化能力を持つという経験的な現象を理論的に解明した。

平成14年12月20日受付

\* 大連理工大学電子情報工程学院, 九州大学大学院システム情報科学研究科訪問助教授

\*\* 電気電子システム工学部門

また、RBFN を用いてパターン認識をする場合は、出力層で線形積和を取るため、各クラスの間にある程度のオーバーラップがあり、クラスタリングの誤差が生じる。それに対して、閾値関数のような非線形関数を用いるとオーバーラップを避けることができる。このような考察に基づき、本論文は RBFN とパーセプトロンネットワークを結合して、二つの異種の隠れ層を持つ新しいニューラルネットワーク、すなわちラジアル基底パーセプトロンネットワーク (RBPN) を提案した。RBPN の第一隠れ層は RBFN の隠れ層に相当し、第二層は単純パーセプトロンネットワークに相当している。隠れ層の非線形パラメータが確定した後は、RBPN の入出力間の関係は RBFN と同様であることから、連立一次方程式で表される。それで重みの学習には LS, RLS 等の線形手法を用いることができ、大域的収束が保証される。また RBPN は RBFN より隠れ層が一層多いことから隠れ層の非線形パラメータの決定法も RBFN とは異なっている。クラスタリングの方法は分類過程において出力のサンプルを考慮するかしないかによって IC (input clustering) 法と IOC (input-output clustering) 法に分けられ、IC 法の中では K-平均法<sup>16)</sup>と隣重類<sup>17)</sup> 法等がもっとも普及しているが、そのうち Chen<sup>18)</sup>等が提案した直交最小二乗法 (OLS) が比較的多く応用されている。この方法は毎回ネットワークの出力エネルギーが最大となるような基本ベクトルを求めるので、自動的に悪条件の問題を避けることができる。ただし、システムの情報は入力と出力の中に同時に含まれているが、分類情報においては出力サンプルの中により多く表れている。従って、分類過程の中で入力情報しか考慮しないと、多くの場合不合理な結果が出てくる<sup>19),20)</sup>。本論文の RBPN は IOC 法を採用し、第一隠れ層に対しては入力空間を基礎として入力サンプルデータから典型的なサンプルを探して、この隠れ層の中心とし、第二隠れ層に対しては出力空間を基礎として、出力データから分類情報を求め、これによって隠れ層の要素の数を決定する。両隠れ層の間は出力サンプルの分類情報で結合しているの、正確な分類が実現可能である。

2. RBPN 法について

ニューラルネットワークを構築するには、主に二つの要素を考慮しなければならない。すなわちネットワークのトポロジーと、ネットワークの学習・動作の規則である。この二つの要素が一緒になってネットワークの主な特性を表す。本論文で提案する RBPN の主な特性は以下のようなものである：1) ネットワークの構造は 4 フィードフォワード型であり、非線形写像は二つの隠れ層によって行われる。各隠れ層のノード数は学習中に動的に決定され、さらに二つの隠れ層の間は可変結合であり、その結合は出力空間によって決められる。2) 学習法においては、分

類過程は入力情報と出力情報を同時に考慮する IOC 法を取り、クラス中心の形状パラメータは適応的に調整され、重みの調整過程は LS あるいは RLS のような線形最適化法が用いられる。

2.1 RBPN の構造

RBPN の構造を図 1 に示す。x, y は前述のとおりであり、 $W = (\omega_1, \omega_2, \dots, \omega_M)^T \in R^M$  は出力層の重みマトリックス、 $\omega_0 = (\omega_{01}, \omega_{02}, \dots, \omega_{0L})^T$  は出力ユニットのバイアス、 $c_i$  はネットワークの第 i 箇の RBF の中心、 $t_j = f(\phi_1(\cdot))$  は第二隠れ層の入力ベクトル、 $a = (a_1, a_2, \dots, a_M)^T \in R^M$  は第二隠れ層の閾値、 $\phi_1(\cdot)$  は第一隠れ層のラジアル基底関数、 $\phi_2(\cdot)$  は第二隠れ層の伝達関数である。

RBPN の第一隠れ層は RBFN の隠れ層と同一であり、ノードの伝達関数  $\phi_1(\cdot)$  は、通常の Gauss 関数や二次関数等を使用する。RBPN の第二隠れ層は単純パーセプトロンネットワークと同一であり、伝達関数は Heaviside 関数や線形関数等を採用している。この層のノードと前層のノードの間はクラスに分かれて選択的に結合される。出力層は RBFN と同様であり、線形積和を取る。この重みは学習中にサンプルの特性から適応的に習得される。

以上述べたようにネットワークの第一隠れ層第 i ノードの出力はラジアル基底関数の出力  $\phi_1(x, c_i)$  であり、 $\phi_1(\cdot)$  が任意階まで微分可能な関数であれば、初期値  $x_0$  でのテーラー級数展開は

$$\phi_1(x, c_i) = \phi_1(x_0, c_i) + \frac{d\phi_1(x, c_i)}{dx} \Big|_{x=x_0} (x - x_0) + \frac{1}{2!} \frac{d^2\phi_1(x, c_i)}{dx^2} \Big|_{x=x_0} (x - x_0)^2 + \dots \quad (2)$$

である。ネットワークの第二隠れ層の入力端から考察すれば、第 j ノードの入力  $t_j$  は全ネットワークの入力パターンの複雑な多項式である。第一隠れ層で実現するのはパターン空間の超曲面による分割である。同様にネットワークの出力端から考察すると第 j 出力ノードは

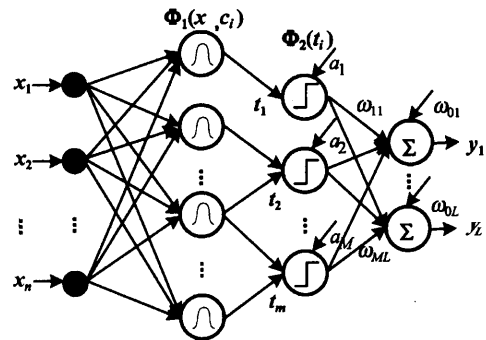


Fig.1 Architecture of RBPN.

$$y_j = \sum_{i=1}^M \omega_{ij} \phi_2(t_i) + \omega_{0j} \quad (3)$$

であり、したがって、第二隠れ層の機能はパターン空間の超平面分割である。パターン分類中、システムの分類情報が予知できれば、第二隠れ層を利用して線形的に分類ができ、各クラスのサンプル中の典型的なサンプルを探して、第一隠れ層の中心とする。それと同時に同じ種類同士に結合させる。すなわち、第一隠れ層を通して入力空間に基づいてクラス別に超曲面を分割する。従って、クラス別に精確に近似ができる。以下は計算法について詳細に述べる。

## 2.2 RBPN 計算法

上記のように、RBPNはRBFNとパーセプトロンの特徴を結合したものであるが、以下の問題を解決しなければならぬ：1) ネットワークの設計、すなわち二つの隠れ層のノードとRBFの中心と形状のパラメータの確定；2) 隠れ層間の結合、隠れ層間の結合選択規則；3) 出力重みの修正。3) の重み修正に関してはLS、RLS等の計算法が良く採用されており、基本的には成熟している。本論文では、如何に二つの隠れ層のパラメータとノード数を決定するかについて述べる。

### 2.2.1 隠れ層ノード数及びRBF中心の決定

まず本論文のネットワークは、出力空間がサンプル集合内の任意点に対するパターン認識を行ってシステムのクラス情報を得ている。サンプルの総数は  $S$  とし、受け入れられるサンプル間の最大距離を距離係数  $r$  で表す。第一、第二隠れ層のノード数を  $M_1, M_2$  で表記する。具体的な方法は以下のようなものである。

**Step 1:** 既知サンプル点  $(x^1, y(x^1))$  に対して、 $s = 1$  と設定し、第二隠れ層のノード数を  $M_2 = 1$  とする。また、 $z_1 = y(x^1)$  とおく。

**Step 2:**  $s$  番目のサンプル点  $(x^s, y(x^s))$  を入力する際、第二隠れ層ノード数は  $M_2 = M$  とし、

$$d_i = \min_j \|y(x^s) - z_j\| \quad (4)$$

を計算する。ここで、 $i = 1, 2, \dots, M$  であり、もし  $d_i > r$  ならばネットワークの第二隠れ層に陰ノード一つを加え、また  $z_{M+1} = y(x^s)$ ,  $s = s + 1$ ,  $M = M + 1$  にして Step 2 に戻る。逆に、 $d_i < r$  の場合、 $x^s$  を第  $i$  類に分類し、 $M^i = M^i + 1$ ,  $s = s + 1$  にして Step 2 に戻る。もし  $s = S$  ならば、サンプルのクラス数は  $M_2$  と見なし、Step 3 に行く。

**Step 3:** 第一隠れ層を初期化する。  $M_1 = M_2 = M$  と

し、また二つの隠れ層ノード間には一々対応して接続されているとする。  $H = [h_{ij}]_{M_1 \times M_2}$ ,  $i = 1, \dots, M_1$ ,  $j = 1, \dots, M_2$ 。もし、 $h_{ij} = 1$  ならば第一隠れ層  $i$  ノードと第二隠れ層の  $j$  ノードを接続させて。逆に、 $h_{ij} = 0$  ならば接続を切る。第一隠れ層の初期中心は

$$c_i = \frac{1}{M_i} \sum_{x^s \in z_i} x^s \quad (5)$$

とし、第一隠れ層中心の形状パラメータ初期値は

$$\sigma_i = \alpha \cdot d_{im} \quad (6)$$

とする。ここで、 $d_{im} = \max_i \|x^s - c_i\|$  は、 $z_i$  クラスサンプル点の相対中心の最大距離であり、 $\alpha$  はその幅を決める係数であり、必要に応じて適当に選択する。

次に以上の結果に基づいて各クラスに対して入力サンプルの中で典型的なサンプルを探して、それを第一隠れ層の中心とし、その中心に対応するパラメータを決定する。ネットワーク伝達関数  $\phi_1(\cdot)$  および  $\phi_2(\cdot)$  はそれぞれ Gauss 関数、Heaviside 関数であり式 (11) と (9) で表わされる。

(1) ネットワークの初期条件は前述通りである。

(2) ネットワーク学習が第  $k$  ステップまで進行したとして、現在のネットワークのパラメータを利用して式 (8), (9), (10), (11) によってネットワークの出力  $\hat{y}$  を計算する。

$$\hat{y} = \omega_0 + \sum_{j=1}^{M_2} \omega_j \phi_2(t_j) \quad (7)$$

$$\phi_2(t_j) = \begin{cases} 1 & t_j > a_j \\ 0 & t_j \leq a_j \end{cases} \quad (8)$$

$$t_j = \max(\phi_1(x, c_i) \times h_{ij}) \quad (9)$$

$$\phi_1(x, c_i) = \exp\left\{-\frac{\|x - c_i\|^2}{2\sigma^2}\right\} \quad (10)$$

今ネットワークの目標関数を以下のように定義する。

$$E_S = \frac{1}{2} \sum_{i=1}^L \sum_{s=1}^S \|y_i(x^s) - \hat{y}_i(x^s)\|^2 \quad (11)$$

$E_S$  を計算して期待値になったかを判断する。期待値になったら反復を停止し、学習を終了させる。もし期待値に達していなかったらLS法によって重みを調整する。さらに  $\max \|y(x^s) - \hat{y}(x^s)\|$  を計算し、出力誤差が最大になるサンプルを探してこの誤差が生じる中心の形状パラメータ値を調整する。計算法は2.2.2節で述べる。

(3)  $x^s$  が中心にあるかどうかを判断し、もし中心にあるならば  $k = k + 1$  にして、(2) に戻る。そうでなければ

ば  $x^s$  を新しい中心として第一隠れ層中に一つの陰要素を付加し,  $c_{M_1+1} = x^s$  とする. マトリックス  $H$  の中に第  $M_1 + 1$  行を追加する.  $y(x^s)$  と  $z_j, (j = 1, 2, \dots, M_2)$  を比較し, もし  $x^s \in z_j$  なら新しく追加したノードと第二隠れ層ノードを接続し,  $h_{M_1+1,j} = 1$  とする. 逆の場合は接続せず,  $h_{M_1+1,j} = 0$  とする. また,  $k = k + 1$ ,  $M_1 = M_1 + 1$  にして (2) に戻る.

以上説明したように, この計算方法はまず出力サンプル中に含まれているクラス情報から第二隠れ層ノードの数を決定する. 次に学習過程でパターンの特性によって第一隠れ層のノード数を決定し, さらに入力データを詳細に分類する. 二つの隠れ層間の接続は出力空間によって決まる. この方法はまた同時にネットワークの構造とパラメータの調整が実現できる, すなわち中心の個数とパラメータを決定すると同時に, ネットワークの重みも調整する. 総体的に考えれば, ネットワークの入力から出力への写像は非線形であるが, ネットワークの出力は調整できるパラメータについては線形である. 第二隠れ層を増加したが, これは簡単な写像計算であり計算速度に影響はそれほどない. ゆえに, このネットワークは RBF ネットワークの学習速度が速い特長と局所最小問題を避ける等の長所を保持している.

### 2.2.2 RBF形状パラメータの自己調整学習

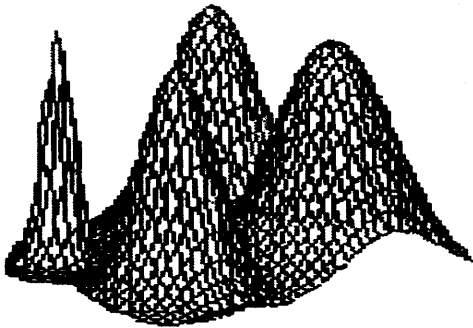


Fig.2 Three dimensional topological drawing of Gauss function.

通常 RBF に関数は Gauss 関数や二次関数を多用する. Gauss 関数を組み合わせた三次元位イメージ図は図2に示した. 図から良く分かるようにパラメータ  $\sigma$  は関数の分布に対する影響が非常に大きい. 従って, 分類器の性能と収束速度にも影響している.  $\sigma$  値が大きければこのノードには比較的大きい範囲のパターンに影響し, 誤差の許容性能がよい.  $\sigma$  値が小さければ誤差の許容性が悪いが局所性が優れている. 従って各システムに対しては気をつけて選択すべきである. 特に複雑な不規則なシステムに対してはそれぞれの中心の  $\sigma$  値が別々になる可能性がある. それで  $\sigma$  はサンプルデータ特性によって適応的に

変化する必要がある. もし,  $t_j = f(\phi_1(\cdot)), \phi_2(\cdot)$  がなめらかな関数ならば, 適応型最急降下法で  $\sigma$  を調整する.

$$\sigma_i(k) = \sigma_i(k-1) + \eta \delta_i(k) \quad (12)$$

$$\begin{aligned} \delta_i(k) &= -\frac{E_s(k)}{\sigma_i(k)} \\ &= -\sum_{j=1}^{M_2} \frac{E_s(k)}{t_j} \frac{t_j}{\sigma_i(k)} \times h_{ij} \\ &= \sum_{j=1}^{M_2} \frac{E_s(k)}{t_j} \frac{t_j}{\phi_1(x, c_i)} \times h_{ij} \\ &\quad \times \exp\left\{-\frac{\sum_{s=1}^S \|x^s - c_i\|^2}{2\sigma_i^2(k)}\right\} \end{aligned} \quad (13)$$

$$(i = 1, \dots, M_1, j = 1, \dots, M_2)$$

ただし実際問題では  $t_j, \phi_2(\cdot)$  がすべてなめらかな関数とは言えない. (9) 式と (10) 式に示した Heaviside 関数と最大値を取る関数を採用する場合, 次元探索において“成功-失敗法”で  $\sigma$  を調整する.

Step 1: 初期点  $\sigma_0$  を決め,  $\lambda$  を探索する;

Step 2: 第  $k$  次まで学習してから  $\sigma_k = \sigma_{k-1}, E_s(k)$  を計算する.

Step 3: もし  $E_s(k) < E_s(k-1)$  ならば探索は成功, 2.2.1 の Step 3 に進む. もし  $E_s(k) \geq E_s(k-1)$  ならば探索は失敗,  $\lambda$  を  $-\lambda \times \beta$  に変更し, Step 2 に戻って引き続き探索する. ここで  $\beta$  は減衰係数であり, 状況によって適切に選択する.

### 3. RBPN の応用とシミュレーション

建築材料のモデルの構築は, その材料の性質を把握し, 合理的に材料を選択するあるいはさらに性能を改善する等の場合, とても重要な役割を果たしている. しかし, 建築材料の内部化合物の形成は, その中に含まれる化学元素の比率や外部環境の温度, 圧力等の多くの要素に影響されるため, 伝統的なモデリング方法ではうまくいかない. 現在主に用いられている差熱解析法や X 線干渉法は, 実験条件の制限により, 局所のあるいは離散的な性質しか得られない. しかし一方では, それらのモデルを構築する過程で多くの経験やデータ蓄積しており, ニューラルネットワークのような新しい手法の適用に多くの素材を残してくれた.

ニューラルネットワークを建築材料の領域に応用する主な理由は, ニューラルネットワークが非線形関数に対して強い近似能力を持つことと, システム初期モデルが必要でないことである. 現在建築材料の特定の特性, 例えば建材の強度などに対しては, ニューラルネットワークを適用してモデル化した事例はすでに文献に報告されている. 本論文はニューラルネットワークを利用して

建築材料の成分モデルを構築する可能性について検討した。以下は  $C_aO - Al_2O_3 - SiO_2$  三次元システムの材料化合物の成分分析への RBPN の適用について説明する。

$C_aO - Al_2O_3 - SiO_2$  三次元相図から 220 組のデータを取り出し、そのうちの 180 組をネットワークの学習に、40 組を検証に用いる。図3に示したように、相図の三つの辺はそれぞれ  $C_aO$ ,  $Al_2O_3$ ,  $SiO_2$  の含量を代表し、図中の + 記号はある温度線上にあるサンプル点を示す。モデルをたてる前に、すべてのサンプルを  $[0, 1]$  に規格化した。

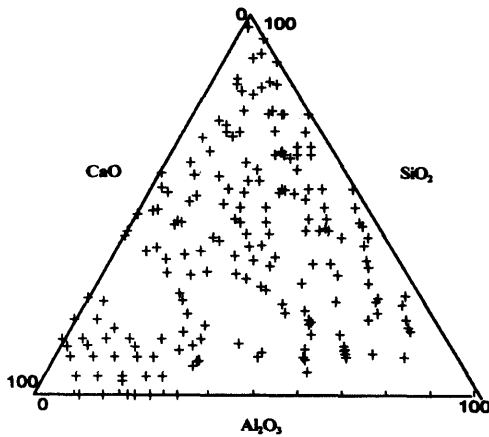


Fig.3 The samples.

コメント：

1)  $\phi_1(\cdot)$  は Gauss 関数,  $\phi_2(\cdot)$  は Heavside 関数,  $t_j$  の表現は (10), (11), (12) に示したとおりである。閾値関数のパラメータは  $a = 0.9$ , Gauss 関数の幅係数は  $\alpha = 2.19$  をとり、繰り返しが開始時にすべてのサンプル所定のクラスに含まれることをはかった。すなわち,  $t_j > 0.9$ , ( $j = 1, 2, \dots, M_2$ ) である。これにより、学習誤差は主にあるクラスの分類の誤りで、他のクラスに含まれてしまって生じたものと見なせる。“成功-失敗法”で形状パラメータ  $\sigma$  を調整する時、適当に  $\sigma$  の値を減少させるだけで良い。こうすれば探索時間を減らすばかりか、学習速度も速くできる。減衰係数は  $\beta = 0.1$  である。

2) 同様に、学習中に新しく付加したノードの中心  $\sigma$  に初期値を設定するとき、同一クラスに所属する中心  $\sigma$  の中から最大の  $\sigma$  の値を取る。このようにすると、学習中に揺らぎが生じるという欠点があるが、汎化能力は逆に向上する。

図4はネットワーク学習誤差と検証誤差を示している。破線が検証誤差、横軸が学習回数である。ネットワークの学習誤差は零まで減少し、汎化誤差も小さくなっている。

ネットワーク中心形状パラメータの変化状況を説明するため  $A_3S_2$ ,  $CS$ ,  $C_2AS$  の三種類の物質を例として、そ

れらの最初の中心形状パラメータ値の変化状況を図5に示した。 $\sigma$  値が大きい値を取れば形成された分割平面はあらゆる当該クラスのサンプル点を含み、誤差が生じる。この場合は  $\sigma$  値を減らし、クラスタリングを強くして誤差を減少させる。

表1は学習中に  $\sigma$  値が適応調整により変化した場合と  $\sigma$  値が定数の場合とを比較したものである。期待学習誤差は 0.002 とした。 $\sigma$  値が適応調整により変化する場合、検証率は最高であり、学習回数に与える影響が少ない。 $\sigma$  値が定数の場合、その値の大きさはネットワーク学習回数と検証誤差に非常に大きい影響を与える。ただし、実際のシステムで適当な  $\sigma$  値を探すのは大変困難である。もしネットワークが具体的な状況によって  $\sigma$  を適応調整できるならば、モデル化の難しさも軽減し、もっと良い結果が得られると考えられる。

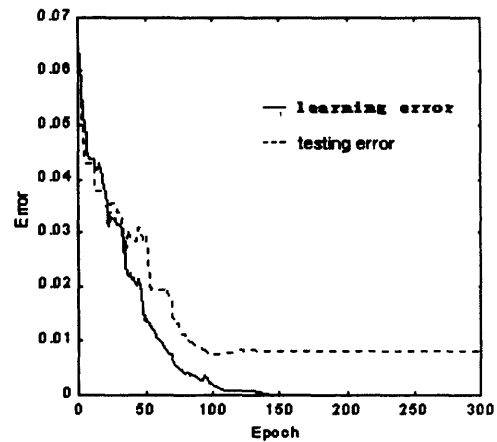


Fig.4 Curves of learning error and testing error.

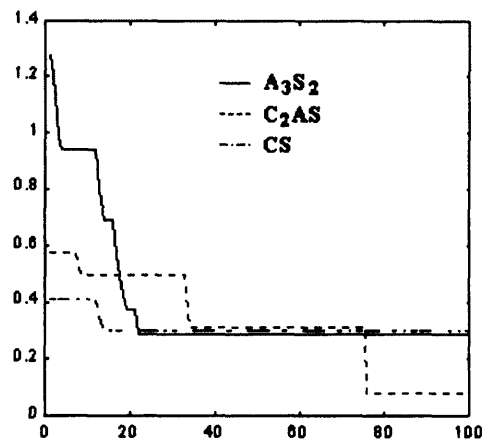


Fig.5 Self-adjustable  $\sigma$ .

**Table 1** Learning times and testing rate of networks with constant  $\sigma$  and self-adjustable  $\sigma$ .

	constant $\sigma$				self-adjustable $\sigma$
	0.1	0.15	0.2	0.25	
learning times	130	127	128	184	145
testing rate	46.0%	56.8%	54.1%	67.6%	78.4%

#### 4. ま と め

本論文は RBF ネットワークとパーセプトロンネットワークに基づいて新しいネットワークを提案し、これを複雑な建築材料成分解析に応用した。以下のような結論が得られた。1) 本論文で提案したネットワークは二つの隠れ層の選択的接続で入力データを写像し、精確な分類を実現し、比較的高い汎化能力を得た。ネットワークの構造とパラメータは同時に調整でき、RBF 計算法の学習速度は速く、局所最小値に陥らない長所を保持している。2) 材料内部成分の構成を解析することができた。これはニューラルネットワークによるシステム同定が非線形システムを近似する有効な方法であること、材料領域への応用が有効であることを示している。

今後の課題として：1) 中心の決定には、誤差が生じる最大点を採用したが、これが最適な結果にいたるとは言えない。2)  $\sigma$  値の適応調整についてのことであるが、第二隠れ層がなめらかでない閾値関数を採用しているので、 $\sigma$  値の調整は次元探索法を使用した方が、いかに実際データに基づいて  $\sigma$  値を探索するかは興味ある研究テーマである。

#### 参 考 文 献

- 1) Hunt, K.J., Sbarbaro, D., Zbikowski R., and Gawthrop P.J.: Neural Networks for Control Systems - A Survey, *Automatica*, **28**(6), pp1083-1112, 1992.
- 2) Huang, D.: Pattern Recognition System Theory of Neural Network, *Electronic Industrial Press*, Beijing, 1996, (in Chinese).
- 3) Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain, *Psychological Review*, **65**, pp386-408, 1958.
- 4) Cho, S.-B.: Pattern recognition with neural networks combined by genetic algorithm, *Fuzzy Sets and Systems*, **103**, pp339-347, 1999.
- 5) Gonzalez-Monroy Luis I., Cordoba A.: Optimization of energy supply systems with simulated annealing, *continuous and discrete descriptions*, *Physica A*, **284**,

- pp433-447, 2000.
- 6) Alkhamis, T.M., Ahmed, M.A., Tuan V.K.: Simulated annealing for discrete optimization with estimation, *European Journal of Operational Research*, **116**, pp530-544, 1999.
- 7) Girosi, F. and T. Poggio: Networks and the best approximation property, *Biological Cybernetics*, **63**, pp169-176, 1990.
- 8) Monica B., et al.: Learning without Local Minimum in Radial Basis Function Networks, *IEEE Trans. Neural Networks*, **6**(3), pp749-755, 1995.
- 9) Schwenker F., Kestler H.A.a. Palm G.: Three learning phases for radial-basis-function networks, *Neural Network*, **14**, pp439-458, 2001.
- 10) Pedrycz W.: Conditional Fuzzy Clustering in the Design of Radial Basis Function Neural Networks, *IEEE Trans. Neural Networks*, **9**(4), pp601-612, 1998.
- 11) Wang Z.-O., Zhu T.: An efficient learning algorithm for improving generalization performance of radial basis function neural networks, *Neural Networks*, **13**, pp545-553, 2000.
- 12) Wei H., Xu S. and Song W.: An evolutionary selecting algorithm for the learning of RBF nets, *Control Theory and Applications*, **17**(4), pp604-608, 2000, (in Chinese).
- 13) Zhu Q., Cai Y., Liu L.: A global learning algorithm for a RBF network, *Neural Networks*, **12**, pp527-540, 1999.
- 14) D.S. Broomhead and D. Lowe: Multivariable functional interpolation and adaptive networks, *Complex Syst.*, **2**, pp321-355, 1988.
- 15) Chester, D.: Why two hidden layers are better than one, *In IEEE Int. Joint Conf. on Neural Network*, pp265-268, 1990.
- 16) S. Chen, S.A. Billings and P.M. Grant: Recursive hybrid algorithm for nonlinear system identification using radial basis function networks, *Int. J. Contr.*, **55**(5), pp1051-1070, 1992.
- 17) Zhu M.-X., Zhang D.-L.: Study on the algorithms of selecting the Radial Basis Function Center, *Journal of Anhui University Natural Science Edition*, **24**(1), pp72-78, 2000, (in Chinese).
- 18) S. Chen, C.F.N. Cowan. and P.M. Grant: Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks, *IEEE Trans. Neural Networks*, **2**(2), pp302-309, 1991.
- 19) Z. Uykan, C. Guzelis and M. Ertugrul Celebi: Analysis of Input-Output Clustering for Determining Centers of RBFN, *IEEE Trans. Neural Networks*, **11**(4), pp851-858, 2000.
- 20) Li J., Zhang Y., Fu X.: An improved closest cluster learning algorithm, *Control Theory and Applications*, **17**(5), pp735-738, 2000, (in Chinese).

