

# 自己組織化マップを用いた動径基底関数ネットワークの構築

山下 和彦<sup>†</sup> Goutam Chakraborty<sup>†</sup> 馬淵 浩司<sup>†</sup> 松原 雅文<sup>†</sup>

<sup>†</sup>岩手県立大学 ソフトウェア情報学部

## 1 はじめに

パターン認識などの分類手法に、ニューラルネットワーク (Neural Network: NN) がある。その中の一つに、動径基底関数ネットワーク (Radial Basis Function Network: RBF ネットワーク) [1] があり、主としてパターン認識に使われている。

RBF ネットワークでは、RBF 係数の計算に逆行列を利用しているため、データ数や中間ユニット数が増えると、計算量が増加するという問題がある。

そこで本研究では、RBF 係数の計算に自己組織化マップ (Self-Organizing Maps: SOM)[2] を用い、RBF 係数の計算の高速化をはかる。そして、一般的な RBF ネットワークと比較し、提案手法の有効性を示す。

## 2 ニューラルネットワーク

### 2.1 自己組織化マップ (SOM)

SOM とは、教師なし競合学習の NN であり、入力層と出力層の 2 層から構成される。SOM は、高次元データを 2 次元へ写像することで、高次元データを視覚化する利点を持つ。

### 2.2 RBF ネットワーク

RBF ネットワークとは 3 層の階層型の NN で、中間層には RBF という基底関数が設けられており、この RBF の出力を出力層へ線形結合することによって、分類を行う。中間層の基底関数として、式 (1) のガウス関数が一般的に使われる。RBF の例を図 1 に示す。

$$\phi_i = \exp\left(-\frac{\|x - \mu_i\|^2}{2\sigma_i^2}\right) \quad (1)$$

ここで、 $\mu_i$  は関数の中央値、 $\sigma_i^2$  は分散、 $x$  は入力ベクトル、 $\phi_i$  は各中間ユニットの出力である。中間層から出力層の結合荷重  $W^T$  は、式 (2) により決定する。

$$W^T = (\Phi\Phi^T)^{-1}\Phi^T T \quad (2)$$

$\Phi$  は、 $N$  行  $H$  列の行列であり、 $N$  はサンプル数、 $H$  は中間ユニット数である。また、 $T$  は教師信号であり、クラス情報が格納されている。この行列は  $N$  行  $C$  列の行列であり、 $C$  は分類数である。

### An Efficient Construction of RBF Network Based on SOM Training

Kazuhiko Yamashita<sup>†</sup>, Goutam Chakraborty<sup>†</sup>, Hiroshi Mabuchi<sup>†</sup> and Masafumi Matsuhara<sup>†</sup>

<sup>†</sup>Faculty of Software and Information Science, Iwate Prefectural University

一般的な RBF ネットワークでは、RBF 係数を計算する際、式 (2) のような複雑な行列演算を必要とし、行列  $\Phi$  が膨大になると、計算量が多くなる。

よって、本研究では、SOM を用いた計算手法を提案し、RBF 係数の計算の高速化を目指す。

## 3 提案手法

### 3.1 概要

提案手法は、SOM による前処理と RBF ネットワークの構築という、2 つの処理を持つ [3]。

### 3.2 前処理

前処理では、SOM によって学習を行い、入力層から出力層の結合荷重を計算する。この時、各学習データの出力ユニットの勝利回数も求め、保存しておく。

### 3.3 RBF ネットワークの構築

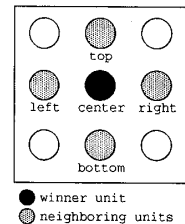
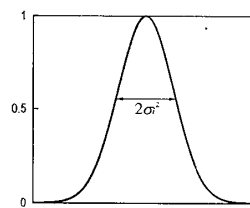


図 1: RBF の例

図 2: SOM の各ユニット

本手法では、SOM の出力ユニット数を RBF ネットワークの中間ユニット数とする。また、入力層から中間層の結合荷重には、SOM の入力層から出力層の結合荷重を利用し、中間層から出力層の結合荷重には SOM によって得られた勝利回数を利用する。中間層から出力層の結合荷重は、各出力ユニットの出力が 1 になるように正規化を行う。

また、RBF の広がり  $\sigma_i$  は、式 (3) により決定する。

$$\sigma_i = \frac{\sum_{j=1}^N d_j}{N} \quad (3)$$

$N$  は近傍ユニット数、 $d_i$  は RBF 係数を求めたい該当ユニットとその近傍ユニットの結合荷重のユークリッド距離を表している。SOM の各ユニットを図 2 に示す。該当ユニットの結合荷重を  $\vec{W}_{center}$  とし、近傍ユニット数が  $N = 4$  の時、四方の近傍ユニットの結合荷重は  $\vec{W}_{top}$ ,  $\vec{W}_{bottom}$ ,  $\vec{W}_{left}$ ,

表 1: 認識率

データ	比較項目	既存手法		提案手法	
		TRAIN	TEST	TRAIN	TEST
Iris	平均値	98.2%	95.7%	95.9%	95.8%
	標準偏差	0.58	4.81	0.77	4.44
Vowel	平均値	72.6%	72.4%	75.8%	75.3%
	標準偏差	1.15	5.36	1.71	5.13
SPECT Heart	平均値	87.8%	82.7%	83.5%	83.2%
	標準偏差	0.78	5.51	0.94	4.85

$\vec{W}_{right}$  のように定義される。そして、RBF 係数を求めたい該当ユニットと近傍ユニットのユークリッド距離は  $d_1(\vec{W}_{center}, \vec{W}_{top})$ ,  $d_2(\vec{W}_{center}, \vec{W}_{bottom})$ ,  $d_3(\vec{W}_{center}, \vec{W}_{left})$ ,  $d_4(\vec{W}_{center}, \vec{W}_{right})$  と定義される。その各ユークリッド距離の平均値を、 $\sigma_i$  とする。

このような計算手法を用いることにより、一般的な RBF ネットワークより高速にパラメータを計算することができるものと考えられる。

## 4 評価実験

### 4.1 概要

今回は、Iris, Vowel, SPECT Heart の 3 種類のベンチマークデータを用いて実験を行い、一般的な RBF ネットワークとの比較を行った。ベンチマークデータの概要を表 2 に示す。学習データ (TRAIN) とテストデータ (TEST) は試行毎に組み合わせを変更しており、その割合は学習データが 9 割、テストデータが 1 割である。

評価には、認識率の平均、標準偏差、処理時間の 3 項目を用いた。なお、実験パラメータを表 3 に示す。ここで、提案手法の評価回数がデータ毎に異なるのは、SOM での学習を考慮したためである。

表 2: ベンチマークデータの概要

	Iris	Vowel	SPECT Heart
データ数	150	671	267
属性数	4	2	22
分類数	3	10	2

表 3: 実験パラメータ

	既存手法	提案手法
試行回数	50 回	50 回
評価回数	-	Iris: 1,000 回 Vowel: 3,000 回 SPECT Heart: 1,000 回
学習率	-	0.1
近傍サイズ	-	3
中間ユニット数	64 個	64(8×8) 個

### 4.2 実験結果および考察

認識率の平均値と標準偏差を表 1 に示す。Iris と SPECT Heart では、TEST に対して、提案手法の方

が既存手法より認識率がやや高い結果となった。しかし、TRAIN では既存手法の方が提案手法より認識率が高かった。また、Vowel では、TRAIN, TEST とともに、提案手法の方が既存手法より高い認識率を示すことができた。標準偏差は、両手法とも同程度の値となり、提案手法は既存手法と同じく安定した認識が可能であることがわかった。

処理時間の結果を表 4 に示す。処理時間は既存手法を基に比で表しており、提案手法の方が処理時間が短いことが確認できた。このようにして、提案手法の方が既存手法より少ない計算量で、RBF ネットワークのパラメータを計算できることが示された。

表 4: 処理時間の比

	既存手法	提案手法
Iris	1	0.87
Vowel	1	0.90
SPECT Heart	1	0.82

## 5 おわりに

本研究では、RBF ネットワークのパラメータの計算を高速化するために、SOM を用いた計算手法を提案し、一般的な RBF ネットワークとの比較を行った。実験結果から既存手法と同程度の精度を示すことができ、処理時間は提案手法の方が短かったことから、提案手法の有効性を示すことができた。

今後は、他の種類のベンチマークデータを用いて実験を行い、本手法が問題特性に依存していないかを検証した後、実問題にも適用させ実験を行う。また、学習データ数や入力次元数が異なるデータを用いて、処理時間の差異も確認する予定である。

## 参考文献

- [1] 坂和正敏, 田中雅博: ニューロコンピューティング入門, 森北出版, 1997.
- [2] T. コホネン: 自己組織化マップ, シュプリンガー・フェアラーク東京, 2005.
- [3] 山下和彦, Goutam Chakraborty, 水野佑治: An Efficient Construction of RBF Network Based on First Layer Training by SOM, 第 7 回情報科学技術フォーラム一般講演論文集, F-007, pp.325-326, September 2008.