

## アンサンブル自己生成ニューラルネットワークの汎化能力改善特性

井上 浩孝・成久 洋之\*

岡山理科大学大学院工学研究科システム科学専攻

\*岡山理科大学工学部情報工学科

(2000年11月1日 受理)

### 1. まえがき

ニューラルネットワーク (Neural Networks: NN) は、パターン認識, クラスタリング, 診断システム, 関数近似, 時系列予測, 適応制御などの知的処理の分野で広く利用されている<sup>3, 11)</sup>。現在最も多く利用されているニューラルネットワークは、バックプロパゲーション (Backpropagation: BP) 学習則を利用した教師あり学習によるフィードフォワードネットワークであるが、そのネットワーク構造として層数, 中間層のユニット数, および諸パラメータを決定しなければならない。ニューラルネットワークの初期設定は研究者各自の経験や勘を頼りに設定しているのが現状である。問題の規模に応じた最良なネットワークを構築するためには、何度も試行錯誤を繰り返す必要があり、大変煩わしい作業である<sup>17, 14)</sup>。このような煩ましい状況を解決するための手法として、ネットワーク構造を遺伝的アルゴリズム (Genetic Algorithm: GA) を用いて最適化する方法<sup>12)</sup> が提案されているが、大規模な問題を扱う場合多大な処理時間を必要とする。

最近、自己生成ニューラルネットワーク (self-generating neural networks: SGNN)<sup>15)</sup> がネットワーク設計の容易さのために注目を集めている。SGNN は Kohonen の自己組織化地図 (self-organizing maps: SOM)<sup>7)</sup> を拡張したものであり、競合学習 (competitive learning) により自己生成ニューラル木 (self-generating neural tree: SGNT) を生成する。SGNN のような動的ニューラル木ネットワークは Racz ら<sup>10)</sup> や Li ら<sup>8)</sup> により提案された。これらは安定性と可塑性を兼ね備えているため、ニューラルネットワークの学習における問題点として知られている安定性-可塑性ジレンマ<sup>2)</sup> が存在しない。しかしながら、Racz らや Li らによって提案された手法は学習係数などのパラメータに非常に敏感であり、パラメータの設定により得られる解の質が大幅に変化する<sup>1)</sup>。

SGNT アルゴリズムは訓練データ集合から直接ニューラル木を自動的に生成するため、パラメータの設定は必要なく、我々は訓練データを与えてやるだけでよい。この SGNT アルゴリズムは基本的に階層型クラスタリングアルゴリズムであるため、分類やクラスタリング問題に関しては他のニューラルネットワークよりも高速に学習を行い、優れた結果を示すことが知られている<sup>16, 4)</sup>。しかしながら、教師あり学習の BP 法に比べ、解の精度はすべてが良いというわけではない<sup>5)</sup>。そこで我々は、未学習テストデータに対する解の精度 (汎化誤差) を改善するための一手法として、SGNN の高速な処理能力を利用した複数の SGNN によるアンサンブル自己生成ニューラルネットワーク (Ensemble Self-Generating Neural Networks: ESGNN) を提案した<sup>6)</sup>。ニューラルネットワークは与えられた訓練データを用いて学習を行い、未知のデータに対してもその期待出力の近似値を算出することができる。ここで、未学習データに対する適応能力のことを汎化能力と呼ぶ。ESGNN では、ノンパラメトリックで動的に生成される SGNT の構造は与えられる訓練データの提示順に依存することに着目し、訓練データより構成する訓練実例ベクトルの提示順をランダムに入れ換えることにより複数の SGNT を生成させ、テストデータに対する全 SGNT の出力値の平均値をネットワーク出力とすることにより汎化能力の改善を行う。

本論文では、ベンチマーク問題として与えられている MONK's<sup>13)</sup>, Cancer と Card<sup>9)</sup> の分類問題に適用した場合の ESGNN の汎化能力改善特性を分析するため、アンサンブルに使用する SGNT の数と分類率の改善特性に関する評価実験を行う。更に、ESGNN と既存の BP 則を用いた階層型 NN との比較実験を行う。実験結果より、ESGNN は入力訓練データを与えるのみで分類問題に関して高速な学習を行い、アンサンブル効果によりすべての問題に対して汎化能力を改善可能であることがわかった。

## 2. SGNT の構成法

SGNT の構成法は、競合学習に基づく教師なし学習法であり、与えられた入力訓練データを順次木構造内に配置する階層型クラスタリングアルゴリズムである。SGNT を構成する基本素子となるニューロンを図 1 に示す。SGNT 内の一つのニューロン  $n_j$  は、子孫内に含まれる葉の個数  $c_j$ 、重みベクトル  $w_j$  を内蔵している。更に、 $n_j$  に直結する  $m_j$  個の子ニューロンとの結合リンク  $l_{js}$  ( $s = 1, \dots, m_j$ ) をもつ。ここで、重みベクトル  $w_j$  は入力訓練データベクトル  $e_i$  の集合  $E$  を SGNT 内に写像するための変数である。

SGNT は根 (root)、葉 (leaf)、根と葉の間に存在するノードにニューロンが割り当てられ、枝 (edge) としてリンクが各ノードに存在する。任意のニューロン  $n_j$  に対する親ニューロンを  $n_{par}$ 、競合学習により選択されたニューロンを  $n_{win}$  と呼ぶ。入力訓練データベクトル  $e_i$  を SGNT の根に投入し、競合学習により  $e_i$  の配置位置を決定する。すなわち、 $e_i$  と  $n_{par}$  の子として連結されているニューロン  $n_j$  の重みベクトル  $w_j$  との距離  $d(e_i, w_j)$  を計算し、最短距離となるニューロンを勝者ニューロン  $n_{win}$  とする。

$$d(e_i, w_{win}) = \min_j d(e_i, w_j) \quad (1)$$

ここで、 $i = 1, \dots, N$ 、 $j = l_{(par)s}$  ( $s = 1, \dots, m_{par}$ ) であり、 $N$  は訓練データ総数を、 $m_{par}$  は  $n_{par}$  に対する子ニューロン総数を表す。距離測度  $d(e_i, w_j)$  はユークリッド距離を用いる。

$$d(e_i, w_j) = \sqrt{\sum_{a=1}^A (e_{ia} - w_{ja})^2} \quad (2)$$

ここで、 $a$  ( $a = 1, \dots, A$ ) は  $A$  次元訓練データベクトル  $e_i$ 、または重みベクトル  $w_j$  中の各要素を表す変数である。ニューロン  $n_j$  の重み  $w_{ja}$  の修正には、次式を用いる。

$$w_{ja} = w_{ja} + \frac{1}{c_j + 1} \cdot (e_{ia} - w_{ja}) \quad (3)$$

式 (3) は、SOM における近傍関数が  $1/(c_j + 1)$  の場合と一致する。

従来の木構造ニューラルネットワークの手法<sup>10, 8)</sup>では、近傍関数の初期値を任意に設定し、繰り返し学習するにつれて単調減少させる。近傍関数の初期設定値の差異により最終的に得られる木構造は変動する。一方、SGNN では勝者ニューロン以下に属する訓練データの期待値を自動的に逐次制御することにより、問題の規模、複雑度に応じて木構造を構築する。

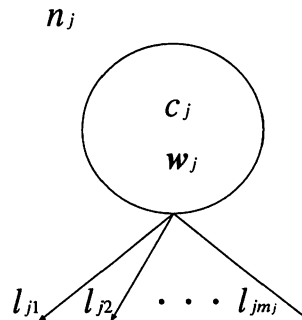


図 1 ニューロンの構造

SGNT 生成手順を以下に示す。

#### SGNT 生成手順

**Step 0.**  $E = \{e_i\}$  ( $i \in N$ ) を読み込み、提示順序を決定する (以後、 $e_i$  の添字  $i$  は提示順を表す。)。しきい値  $\xi \geq 0$ , 距離速度  $d(e_i, w_j)$  を設定する。

**Step 1.** SGNT の根となるニューロン  $n_1$  を生成し、入力訓練データ  $e_1$  を  $n_1$  の重み  $w_1$  に格納 ( $w_1 \leftarrow e_1$ ) し、 $c_1 \leftarrow 1$  とする。更に、訓練データ  $e_2$  を  $n_1$  に入力、 $n_{win} \leftarrow n_1$ ,  $n_{par} \leftarrow n_1$  とし、Step3 へ。

**Step 2.**  $e_i$  を  $n_{par}$  に入力、 $n_{par}$  の子ニューロンに対し式 (1) を用いて  $n_{win}$  を決定、 $n_{win}$  決定後、式 (3) を用いて  $n_{par}$  の重み  $w_{par}$  の修正を行い、 $c_{par} \leftarrow c_{par} + 1$  とする。以下の条件を満たす場合、 $n_{win} \leftarrow n_{par}$  とし、Step 3 へ。

$$d(e_i, w_{par}) < d(e_i, w_{win}) \quad (4)$$

式 (4) を満たさず、 $c_{win} \neq 1$  ならば、 $n_{par} \leftarrow n_{win}$  とし、同様の作業を繰り返す。

**Step 3.**  $c_{win} = 1$  ならば、新たな葉  $n_j$  を生成 ( $c_j \leftarrow 1$ ,  $w_j \leftarrow w_{win}$ ) し、 $w_{win}$  を式 (3) を用いて修正、 $c_{win} \leftarrow c_{win} + 1$ ,  $n_{win}$  と  $n_j$  を連結。  $j \leftarrow j + 1$ 。

**Step 4.** 次の条件を満たす場合、新たな葉  $n_j$  を  $n_{win}$  の子として生成 ( $c_j \leftarrow 1$ ,  $w_j \leftarrow e_i$ ) し、 $n_j$  を  $n_{win}$  に連結。

$$d(e_i, w_{win}) > \xi \quad (5)$$

$i \leftarrow i + 1$ ,  $j \leftarrow j + 1$  とし、 $i = N$  ならば終了。 $i \neq N$  ならば、 $n_{par} \leftarrow n_1$  とし、Step 2 へ。

SGNT は全訓練データを一度提示することで学習を終了するため、何度も訓練データを提示し学習を行う BP 法に比べ非常に高速である。生成された SGNT の各葉には実際の訓練データが格納され、未学習データに対する分類に用いる。テストも訓練と同様に競合学習により SGNT の根から勝者ニューロンを再帰的に決定し、対応する葉のもつクラス情報をテストデータの属するクラスとする。また、SGNN の記憶容量を押えるためのしきい値  $\xi$  は 0 とし、全訓練データを葉に組み込む。

### 3. アンサンブル自己生成ニューラルネットワーク

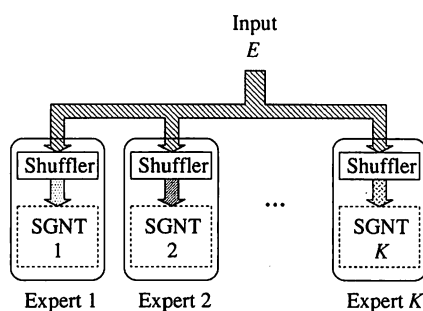
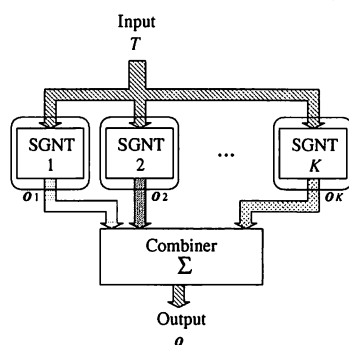
SGNN は高速学習と大規模問題への適用可能性の優れた特性があるが、分類精度は BP 法のような教師あり学習が実装されるフィードフォワード型ネットワークに比べ劣る。そこで我々は SGNN の高速学習特性を利用し、与えられた訓練データからより高い分類精度を引き出すため、 $K$  個の SGNT による ESGNN を考える。SGNT の構造は学習中に動的に変化する。SGNT アルゴリズムは SGNT にすべての訓練データが加えられた後に SGNT の構造を決定する。異なる構造をもつ SGNT は訓練データの入力順を入れ換えることで生成される。

図 2 に訓練過程における  $K$  個の SGNT によるアンサンブル SGNN を示す。図 2 のように、各エキスパートは訓練データの提示順をランダムに入れ換えるための "shuffler" が存在する。訓練データ  $e_{ia}$  ( $i = 1, 2, \dots, N; a = 1, 2, \dots, A$ ) を要素とする集合  $E$  が各エキスパートに入力される。ここで  $A$  はデータの入力次元数であり、各分類問題の属性数を表す。"shuffler" により各 SGNT に提示される訓練データの順序が入れ換えられ、SGNT アルゴリズムにより、 $K$  個の異なる構造をもつ SGNT が個別に生成される。

図 3 にテスト過程における  $K$  個の SGNT による ESGNN を示す。テスト過程において、 $M$  個からなるテストデータの集合  $T$  はこの ESGNN に入力される。各出力ベクトル  $o_k \in \mathbb{R}^M$  はテストデータ  $T$  に対するエキスパート  $k$  の出力値を意味する。この ESGNN の出力は各エキスパートの出力の平均を次のように計算する：

$$o = \frac{1}{K} \sum_{k=1}^K o_k. \quad (6)$$

本論文では、2 値の分類問題にこの ESGNN を使用する。ここで、出力が 0.5 以上であればクラス 1、0.5 未満であればクラス 0 と判断する。

図2  $K$  個の SGNT による ESGNN (訓練過程)図3  $K$  個の SGNT による ESGNN (テスト過程)

#### 4. 実験実施要領

SGNN の性能を分析するために適用した分類問題と比較のために使用した既存の BP 則を用いた階層型ニューラルネットワークの設定, 実験実施要領について述べる。

##### 4.1 分類問題

分類問題として, 入力データが離散値であるベンチマーク問題 MONK's<sup>13)</sup>, ベンチマーク問題集 Proben<sup>19)</sup>の中から, 入力データが実測実数値である Cancer, 離散値, 実数値の混合したデータをもつ Card を用いる。すべて 2 つのカテゴリに分類する問題でそれぞれ 3 セット用意されている。

MONK's<sup>13)</sup> は六つの異なる属性をもつロボットの分類問題で, 各属性の要素の組み合わせにより, 三種類の問題がある。各属性値に対応する要素を表 1 に示す。

- 問題  $M_1$  : (頭の形 = 体の形) または (ジャケットの色 = 赤)。432 個の全パターン中, 訓練入力信号として 124 個がランダムに選ばれている。雑音は入っていない。
- 問題  $M_2$  : 6 つの属性中 2 つが最初の要素の値をもつ。432 個の全パターン中, 訓練入力信号として 169 個がランダムに選ばれている。雑音は入っていない。
- 問題  $M_3$  : (ジャケットの色が緑で剣を持っている) または (ジャケットの色が青ではなく体の形が八角でない)。432 個の全パターン中, 122 個がランダムに選ばれている。訓練入力信号中に 5% の誤分類を含む。

表1 MONK's における六つの属性値

$x_1$ :	頭の形 ∈ 丸, 四角, 八角
$x_2$ :	体の形 ∈ 丸, 四角, 八角
$x_3$ :	笑っている ∈ はい, いいえ
$x_4$ :	持っている ∈ 剣, 風船, 旗
$x_5$ :	ジャケットの色 ∈ 赤, 黄, 緑, 青
$x_6$ :	ネクタイをしている ∈ はい, いいえ

各問題の難易度を比較すると、一次のオーダーである  $M_3$  が最も易しい問題である。次に、 $M_1$  は(頭の形 = 体の形)の部分进行学习するために二次のオーダーの関係があるのでやや難しい。そして、最も難しいのは  $M_2$  である。 $M_2$  は正しいクラスに属しているかどうかを示すために、6つの全属性値を調べなければならない。

Cancer, Card<sup>9)</sup> もパターン分類問題である。Cancer は、乳がん細胞に対して、それが良性腫瘍であるか悪性腫瘍であるかを9つの属性より分類する。全データ数699個中、入力訓練データは350個、テストデータを残りのデータから任意に174個選んである。Card は顧客に対してクレジットカードを認可してよいかどうかを51個の属性から分類する。訓練データ中に5%の誤分類を含む。全データ数690個中、入力訓練データは345個、テストデータを残りのデータから任意に172個選んである。それぞれ3つの問題があるが、訓練データとテストデータがランダムに組み替えられたものである。

#### 4.2 BP ニューラルネットワークの諸設定

SGNN のアンサンブル効果を既存の学習則と比較するために、バックプロパゲーション (Backpropagation: BP) による実験を行う。BPNN では、MONK's, Cancer, Card の各問題に対して、入力層、中間層、出力層の3層構造のネットワークを用いる。なお、入力層、中間層に常に1を入力するしきい値学習用ユニットを1つ備えている。中間層ユニットの数を10、学習係数  $\eta$  を1.2、ゲイン係数  $\varepsilon$  を1とし、入出力関数にシグモイド関数を用いる。学習収束条件として平均二乗誤差 (Mean Squared Error: MSE) が ESGNN と同程度となるまでとする。但し、MSE が局所解に到達してしまい収束しない場合を考慮し、学習回数の上限を5000回とする。

#### 4.3 実験実施要領

以下、実験実施要領について述べる。まず訓練実例データを  $K$  個の NN に学習させる。次に学習終了後の NN にテストデータを入力したときの出力値を算出し、 $K$  個のアンサンブル NN の平均出力より全体の出力値を決定し、全テストデータを提示した後に分類率を求める。ここで、分類率は、次のように定義する。

$$\text{分類率 (\%)} = \frac{\text{正しく分類したデータ数}}{\text{全テストデータ数}} \cdot 100 \quad (7)$$

最後に訓練・テスト終了後総処理時間を求める。ここで、 $K$  の数を SGNN に関して1から100まで20組(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30, 35, 40, 45, 50, 75, 100)とし、BPNN に関しては1と10の2組とする。各試行回数は50回とする。本実験は、COMPAQ DESKPRO (Intel Pentium II 450MHz) を使用し、アルゴリズムはすべてC言語で実装している。

### 5. 実験結果

ESGNN の汎化能力を分析するために、分類問題のベンチマーク問題の中から典型的な9つの問題を使用した。表2に使用した分類データを示す。全問題は2値分類問題である。各分類問題に対し、ESGNN と単一 SGNN および代表的なニューラルネットワーク学習則である BP とそのアンサンブルモデルによる分類精度と処理時間を比較した。

図4(a), (b), (c) に MONK's の3つの問題, Cancer の3つの問題, Card の3つの問題に対するアン

表2 使用した分類データ。#Data - データ数(訓練データ数, テストデータ数), Type - 属性タイプ, #A - 属性数

Name	#Data	#A	Type
MONK's	$M_1$ 432 (124,432)	6	discrete
	$M_2$ 432 (169,432)	6	discrete
	$M_3$ 432 (122*,432)	6	discrete
Cancer	699 (350,174)	9	continuous
Card	690* (345,172)	51	mixed

\* : included 5% misclassification

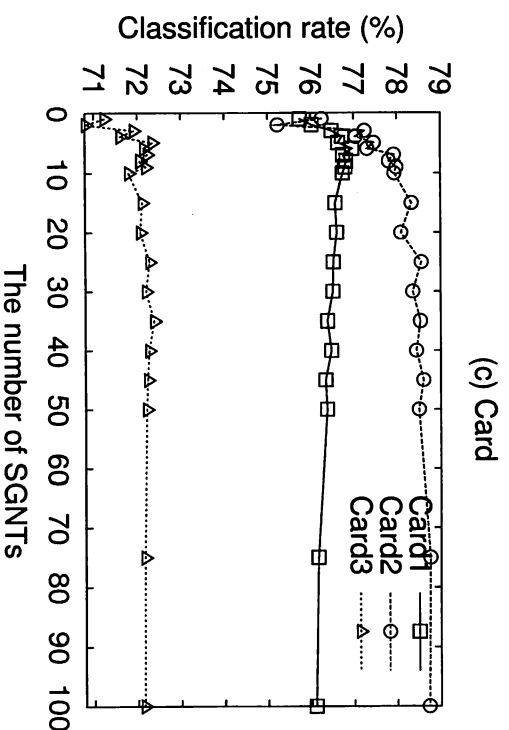
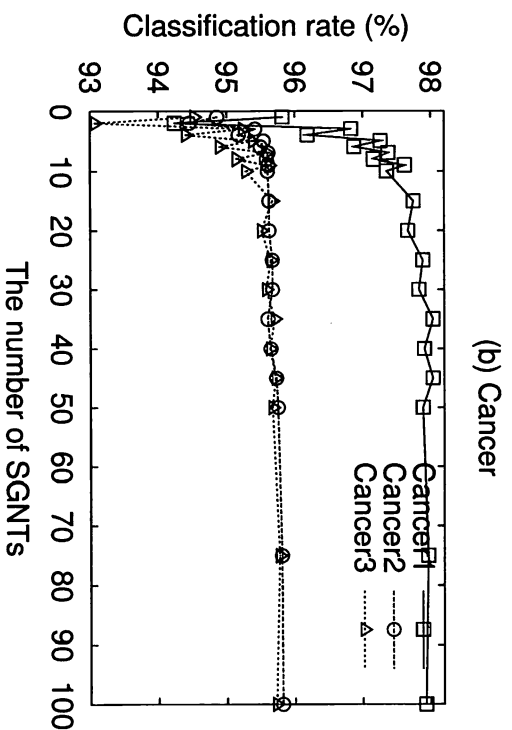
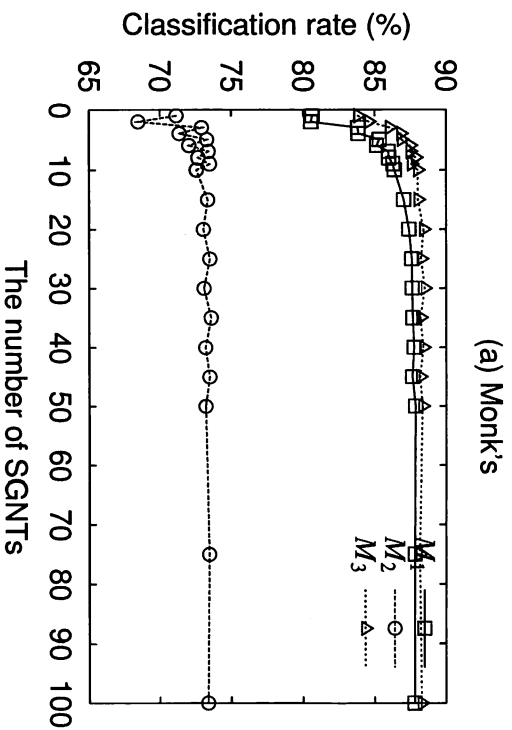


図 4 テンサンプルに使用する SGNT の数 ( $K$ ) と分類率との関係

サンプルに使用する SGNT の数と分類率との関係を示す。ここで、各分類率は 50 回の試行の平均である。図 4 より、すべての問題に対してさまざまな SGNT による ESGNN が分類精度を高めることがわかる。

表 3 分類率 (%) の比較

	$K$	$M_1$	$M_2$	$M_3$	Cancer1	Cancer2	Cancer3	Card1	Card2	Card3
SGNN	1	80.6	71.2	83.8	95.8	94.9	94.5	75.7	76.3	71.2
ESGNN	10	<b>86.4</b>	72.5	<b>88.0</b>	<b>97.4</b>	<b>95.6</b>	<b>95.2</b>	76.7	<b>77.9</b>	71.8
ESGNN	best	87.8	73.6	88.5	98.0	95.8	95.7	77.0	78.8	72.4
BPNN	1	78.9	<b>73.9</b>	81.6	96.3	94.2	<b>95.2</b>	<b>82.5</b>	77.7	<b>79.9</b>
EBPNN	10	79.5	76.9	82.8	97.6	93.9	95.3	85.8	79.8	81.0

表 4 処理時間 (sec.) の比較

	$K$	$M_1$	$M_2$	$M_3$	Cancer1	Cancer2	Cancer3	Card1	Card2	Card3
SGNN	1	0.06	0.08	0.06	0.12	0.12	0.12	0.46	0.44	0.44
ESGNN	10	0.63	<b>0.78</b>	0.65	<b>1.27</b>	<b>1.29</b>	<b>1.24</b>	<b>4.62</b>	<b>4.40</b>	4.46
ESGNN	best	3.13	2.73	1.95	4.25	12.56	8.92	2.78	33.10	15.58
BPNN	1	<b>0.07</b>	6.15	<b>0.26</b>	2.31	2.08	1.34	7.04	11.43	<b>2.86</b>
EBPNN	10	0.66	61.46	2.55	23.13	20.83	13.44	70.43	114.25	28.64

改善能力は 2 から 10 までの SGNT が最も効果的に得られている。20 以上の SGNT の ESGNN では、分類率がほぼ収束してしまう。以上の結果より、分類率の改善効果、および処理時間から判断して、アンサンブルに使用する SGNT の数は 10 個から 20 個が妥当であると考えられる。

表 3 に単一 SGNN ( $K = 1$ )、単一 BPNN ( $K = 1$ )、ESGNN、アンサンブル BPNN (EBPNN) による分類率の比較を、表 4 に処理時間の比較を示す。ESGNN に関して、10 個の場合と 20 種類のアンサンブル数の中で最高の認識率を算出したときの結果を、EBPNN に関して 10 個の場合の結果を示す。すべての結果は各試行回数 50 回の平均を示している。

表 3、表 4 より、単一の SGNN/BPNN よりもアンサンブルを計算した方が処理時間を犠牲に分類率が向上しているが (BPNN の Cancer2 以外)、単一 SGNN と BPNN、10 個の SGNN と BPNN を用いたアンサンブルモデルを比較すると、すべての場合で SGNN を用いた方が高速であることがわかる。 $M_1$ ,  $M_2$ ,  $M_3$  に関して、 $M_2$  の結果は単一 BPNN の分類率に比べ劣るが、短時間で精度を近づけることができ、 $M_1$ ,  $M_3$  においては単一 SGNN でも単一 BPNN よりも高い分類率を得られているが、更に 10 個の ESGNN により 10 個のアンサンブル BPNN に比べ短時間で高い分類率を得られている。Cancer の問題に関しては、単一の SGNN でも 95% 近くの高い分類率を示しているが、10 個のアンサンブル SGNN により単一の BPNN を計算するより短時間で分類率を高めることができ、10 個のアンサンブルを計算した BPNN と同程度かそれ以上の分類率が得られている。Card に関して、Card1, Card3 については SGNN は BPNN には及ばないが、Card2 に関しては 10 個のアンサンブル SGNN により単一の BPNN よりも短時間で BPNN と同程度の分類率を得ることができる。

表 3、表 4 において、太字で示している結果は 10 個の ESGNN と単一の BPNN の比較である。9 問中 6 問で ESGNN が単一の BPNN よりも高い、もしくは同等の分類率が得られ、9 問中 6 問で ESGNN が単一の BPNN よりも短い処理時間となっている。

## 6. むすび

本論文では、分類精度を改善するための ESGNN の性能分析を行った。実験結果より、ESGNN は 9 問中 6 つの問題に対して代表的なニューラルネットワーク学習則である BP よりも短時間に分類率を改善可能であることがわかった。BP でも同様にアンサンブル効果が期待できるが、それには多大な処理時間を必要とする。故に、ESGNN は短時間で複数のアンサンブルを取ることができ、BP によるフィードフォワード型ニューラルネットワークと同程度の分類率を算出可能なシンプルで汎化能力のあるニューラルネットワークの一手法であるといえる。

ESGNN の問題点は、訓練データが大量に存在する場合、記憶容量が不足するおそれがあることがあげられる。更に、アンサンブルに使用する SGNN の数に比例して処理時間を費やす。そこで、これらの問題を解決するためには並列計算機を用いた並列分散処理が有効であると考えられる。今後、MIMD 型の並列計

算機を用いて ESGNN の各 SGNN を各プロセッサに割り当てて並列分散処理を行う予定である。各 SGNN は同一の訓練データ集合から SGNT アルゴリズムによって自動的に学習を行うため、理想的な並列効率が期待される。

#### 参考文献

- 1) K. Butchart, N. Davey, and R. Adams. Comparative study of two self organizing and structurally adaptive dynamic neural tree networks. In J. G. Taylor, editor, *NEURAL NETWORKS AND THEIR APPLICATIONS*, chapter 7, pp. 93–112. John Wiley & Sons, New York, 1996.
- 2) G. A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, Vol. 37, pp. 54–115, 1987.
- 3) S. Haykin. *Neural Networks: A comprehensive foundation*. Prentice-Hall, Upper Saddle River, NJ, second edition, 1999.
- 4) 井上浩孝, 古橋秀樹, 成久洋之. 自己生成ニューラルネットワークの学習特性に関する研究. 電子情報通信学会技術研究報告, COMP98-79, Jan. 23 1999.
- 5) Hirotaka Inoue and Hiroyuki Narihisa. Performance of self-generating neural network applied to pattern recognition. In *5th International Conf. on Information Systems Analysis and Synthesis (ISAS'99)*, Vol. 5, pp. 608–614, Orlando, FL, Aug. 1999.
- 6) Hirotaka Inoue and Hiroyuki Narihisa. Improving generalization ability of self-generating neural networks through ensemble averaging. In Takao Terano, Huan Liu, and Arbee L P Chen, editors, *The Fourth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-2000)*, Vol. 1805 of *LNCS*, pp. 177–180, Springer-Verlag, Kyoto, Japan, Apr. 18–20 2000.
- 7) T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, 1995.
- 8) T. Li, Y. Tang, S. Suen, and L. Fang. A structurally adaptive neural tree for recognition of large character set. In *Proc. 11th IAPR*, pp. 187–190, 1992.
- 9) L. Prehelt. PROBEN1 — a set of neural network benchmark problems and benchmarking rules. Technical report 21/94, Universität Karlsruhe, 1994.
- 10) J. Racz and T. Klotz. Knowledge representation by dynamic competitive learning techniques. In *SPIE Applications of Artificial Neural Networks II*, Vol. 1469, pp. 778–783. SPIE, 1991.
- 11) 坂和雅俊, 田中雅博. ニューロコンピューティング入門. 森北出版, 1997.
- 12) Hiroki Takahashi and Masayuki Nakajima. Evolutional design and training algorithm for feedforward neural networks. *IEICE TRANS. INF. & SYST.*, Vol. E82-D, No. 10, pp. 1384–1392, Oct. 1999.
- 13) S. B. Thrun et al. The MONK's problems — a performance comparison of different learning algorithms. Technical report CMU-CS-91-197, Carnegie Mellon University, 1991.
- 14) Philip D. Wasserman. ニューラル・コンピューティング — 上級編 —, 第 6 章, 森北出版株式会社, 1998.
- 15) W. X. Wen, A. Jennings, and H. Liu. Learning a neural tree. In *International Joint Conf. on Neural Networks (IJCNN'92)*, Beijing, China, 1992.
- 16) W. X. Wen, V. Pang, and A. Jennings. Self-generating vs. self-organizing, what's different? In P. K. Simpson, editor, *Neural Networks Theory, Technology, and Applications*, IEEE Technology Update Series, pp. 210–214. IEEE Technical Activities Board, Piscataway, NJ, 1996.
- 17) 山田敬嗣. ニューラルネットによるパターン認識 [i] — 基本的な考え方と応用例. 電子情報通信学会誌, Vol. 82, No. 8, pp. 852–859, Aug. 1999.



# Characteristics of Improving Generalization Capability of Ensemble Self-Generating Neural Networks

Hiroataka INOUE and Hiroyuki NARIHISA\*

*Graduate school of Engineering,  
Okayama University of Science.*

*1-1 Ridai-cho, Okayama 700-0005, Japan.*

*\*Department of Information and Computer Engineering,  
Okayama University of Science.*

*1-1 Ridai-cho, Okayama 700-0005, Japan.*

(Received November 1, 2000)

Self-generating neural networks (SGNNs) have the features of the simplicity on networks design and the fast processing by automatically constructing self-generating neural tree (SGNT) from a given training data set. We have proposed ensemble self-generating neural networks (ESGNNs) to improve the generalization capability for classification problems. The ESGNN is composed of plural SGNTs each of which is generated by shuffling the order of the given training data independently, and the output of the ESGNN is averaged all outputs of the SGNTs. In this paper, we investigate the characteristics of improving generalization capability of ESGNNs comparing with the standard backpropagation (BP) method. Our results show that ESGNNs are comparable to the standard BP method less computation time.