

自己生成ニューラルネットワークの性能評価

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

(1999年11月4日 受理)

1. まえがき

ニューラルネットワーク (Neural Networks: NN) は、与えられた訓練データから学習を行い、入出力関係をネットワーク内の結合重みに記憶することにより、さまざまな知的処理を実行することができる。具体的には、パターン認識やクラスタリング、診断システム、関数近似や時系列予測、適応制御などに利用され、さまざまなニューラルネットワークアーキテクチャや学習法が提案されている^{6, 18)}。

現在最も多用されている階層型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であるが、そのネットワーク構造として、層数や中間層のユニット数および学習則としての諸パラメータを決定しなければならない。これらの仕事はネットワーク利用者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適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。

与えられた訓練データよりそれらのデータ間の特徴を自動的に抽出する方法として Kohonen の自己組織化ニューラルネットワーク (Self Organizing Neural Networks: SONN)¹¹⁾ が提案されているが、この場合についても学習係数やその他のパラメータ等を決定しなければならない。SONN は競合学習則により特徴抽出を実現するので、競合学習クラスタリングニューラルネットともみなされている。これに対応するものとして、動的ニューラル木ネットワークが Racz¹⁵⁾ らや Li¹²⁾ らにより提案されていたが、これらは安定性と可塑性を兼ね備えているため、他の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²⁾ が存在しない。しかしながら、学習係数等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¹⁾。最近、これらの木構造ニューラルネットワークにおけるパラメータ設定を省略して、与えられたデータから自動的にニューラル木を構成する自己生成ニューラルネットワーク (Self Generating Neural Network: SGNN)¹⁷⁾ が提案され注目を集めている。これは、本来 SONN と同様に競合学習に基づく手法で自己生成ニューラル木 (Self Generating Neural Tree: SGNT) を構成することにより学習を行うものである^{9, 8, 10)}。

本論文では、SGNN の性能を分析するため、分類問題、パターン認識問題、時系列予測問題に適用した場合の性能特性を他の既存の有効なニューラルネットワークとの比較を行い、処理時間と解の質において特性を分析し、汎用の階層型ニューラルネットワークと同様な応用の可能性を示す。分類問題としてはベンチマーク問題として与えられている Monks¹⁶⁾、Cancer と Card¹⁴⁾ の分類問題、認識問題としては独自に作成したアルファベット 26 文字の文字認識問題、時系列予測問題としては Mackey-Grass 微分方程式¹³⁾ の予測問題を使用する。また、SGNN の性能を分析するため、分類、文字認識に対して数種の階層型 NN との比較を行う。更に、時系列予測では、BP 以外にカウンタープロパゲーション等との比較において良好な結果が得られている。SGNN は入力訓練データを与えるのみで分類、クラスタリングの問題と同様に、関数近似、時系列予測等の実変数写像の問題に対しても高速な学習を行い、良質な結果を得ることができることを示す。

2. SGNT の構成法

SGNT 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

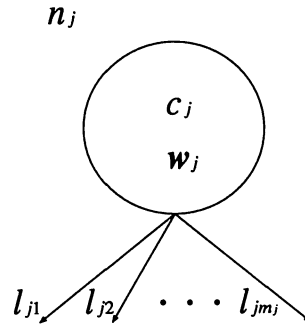


図1 ニューロンの構造

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} ($s = 1, \dots, m_j$) を持つ。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 の根に入力し、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。

$$d(e_i, w_j) = \sqrt{\sum_{k=1}^n (e_{ik} - w_{jk})^2} \quad (2)$$

ここで、 k ($k = 1, \dots, L$) は L 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_{jk} の修正には、次式を用いる。

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

式(3)は、自己組織化マップ (Self-Organizing Maps: SOM) における近傍関数が $1/(c_j + 1)$ の場合と一致する¹¹⁾。

従来の木構造ニューラルネットワークの手法^{15, 12)}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、複雑度に応じて均質な木構造を構築する。

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、 $n_{win} \leftarrow n_1$ 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, $n_{par} \leftarrow n_{win}$ 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, $n_{win} \leftarrow n_{par}$ 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} に結合。

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

$i \leftarrow i + 1$, $j \leftarrow j + 1$ 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。

3. 実験実施要領

本章では, SGNN の性能を分析するために適用した分類, パターン認識, 時系列予測の各問題と比較のために使用した既存のニューラルネットワークについて述べる。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 Monks¹⁶⁾, ベンチマーク問題集 Proben1¹⁴⁾ 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持つ Card を用いる。すべて 2 つのカテゴリーに分類する問題でそれぞれ 3 セット用意されている。次に文字認識問題に関して, 訓練データとして我々が独自に作成したアルファベット大文字 26 パターン (10×10, 2 値画素) に対する雑音を加えたパターンに対する認識率を調べる。最後に, 時系列予測問題として, Mackey-Glass 微分遅延方程式の予測を行う。

本実験では, 分類問題に関して SUN-4/20H ワークステーション (microSPARCII 200MHz), 文字認識問題, 時系列予測問題に関して 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。

3.1 分類問題

MONK's¹⁶⁾ は六つの異なる属性を持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持つ。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。

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) の部分を学習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

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

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、六つの全属性値を調べなければならない。

Cancer, Card¹⁴⁾ 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九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。それぞれ三つの問題があるが、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。

SGNN の性能を分析するために、比較する既存の学習則としてバックプロパゲーション (Backpropagation: BP), 学習ベクトル量子化 (Learning Vector Quantization: LVQ) アルゴリズムの中から LVQ1¹¹⁾, カウンタープロパゲーション (Counterpropagation: CP)⁷⁾, CP の動的モデルであるビジラントカウンタープロパゲーション (Vigilant Counterpropagation: VCP)⁴⁾ の各 NN 学習アルゴリズムによる実験を行う。

以下、分類問題に適用した各 NN の設定を説明する。BPNN では、Monks, 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。学習係数 η を 1.2, 入出力関数にシグモイド関数を用いる。また、中間層ユニットの数を Monk, Cancer の問題に関して 5 個から 50 個まで 5 個刻みで、Card の問題に関して 10 個から 100 個まで 10 個刻みで、それぞれ 10 種類のネットワークの学習を行う。学習収束条件として平均二乗誤差 (Mean Squared Error: MSE) を 0.0001, 学習回数の上限を 1000 回とする。なお、文献¹⁶⁾ では、Monks の問題に対して各 6 つの属性の情報を 0, 1 の 2 値情報に分割し、入力層を 17 個として学習を行っている。本研究では、SGNN と同一な入力値を与えるため、入力層は 6 個とし、与えられたデータをそのまま用いている。

LVQ1 では、コードブックベクトルの数を Monks, Cancer に対しては 10 個、Card に対しては 20 個とした。学習繰り返し回数は、¹¹⁾ において学習に十分な繰り返し回数であるとされているコードブックベクトルの 40 倍とする。学習係数 α の初期値は 0.03 とし、学習を繰り返し行う際に単調減少させる。

CPNN は、競合学習を行う Kohonen 層とフィルタ学習を行う Grossberg 層から構成されている。今回、前方向のみの写像を学習する CPNN を用いる。各問題に対して、Kohonen 層のユニット数を 100 個、Grossberg 層のユニット数を 100 個に固定し、Kohonen 層における学習係数 α と Grossberg 層における学習係数 β を 0.01 から 0.1 まで 0.01 刻みで、0.1 から 1.0 まで 0.1 刻みで変化させた。ネットワークを並行に 3 つ計算し、各ネットワーク出力の平均出力をネットワークの出力データとする。学習回数は 10 回とする。

VCPNN は、静的なネットワーク構造をしている CPNN を初期に競合に参加しない非競合ユニットを付加することにより、動的な変化を可能にしたネットワークである。非競合ユニットを競合に参加させるかどうかを判定するために、監視ユニット (Vigilance Unit, VU) を設ける。今回、VCPNN のネットワーク構成は CPNN と同一とする。非競合ユニットの上限値を 100 とし、VU の非競合ユニット参加判定条件として、教師信号とネットワーク出力との二乗誤差に対するしきい値を 0.5 とする。

3.2 パターン認識問題

今回使用した問題は、 10×10 の 100 の要素 (各要素の値は 0, 1 の 2 値) からなるアルファベット大文字 26 パターンの認識である。訓練実例データとして、各文字は我々が独自に数値を割り当てて作成したオリジナルパターンを使用する。今回、テストデータとして、訓練データから全文字 5% から 50% まで 5% 毎にランダムにビット反転させたものを、合計 10 種類作成する。

実施要領は、まず訓練実例データを各ニューラルネットワークに学習させ、それらの処理時間を求める。次に学習終了後のネットワークにテストデータを入力したときの出力値より、認識率を求める。ここで、認識率は、次のように定義する。

$$\text{認識率 (\%)} = \frac{\text{正しく認識した文字数}}{\text{全文字数}} \cdot 100 \quad (6)$$

前節の分類問題と同様に、SGNN の認識性能を分析するための既存学習則として BP, CP, VCP を使用する。BPNN において、ネットワークの構成は、入力層、中間層、出力層の 3 層とする。各層間のニューロンの数は入力層に 100 個、出力層に 26 個とし、中間層のニューロン数は 5~100 まで 5 ごとに变化させ

て20通りの階層型ニューラルネットワークを作成する。また、入力層、および中間層にはしきい値学習用に、常に入力を1とするユニットが1つずつ加えられている。全学習試行回数は、初期結合重みを-1から1までの範囲でランダムに発生させ、10回行う。学習係数 η は0.9とし、教師信号として、各文字に対応する出力ユニットに1を、その他のユニットに0を与える。CPNNでは、Kohonen層のユニット数を100個、Grossberg層のユニット数を100個とし、Kohonen層の学習係数 α とGrossberg層の学習係数 β はそれぞれ0.01から0.1まで0.01ごとに、0.1から1.0まで0.1ごとに変化させて、400回の試行を行う。VCPNNに関して、VUのしきい値を0.5、非競合ユニットの最大増減値を100とする。その他、VCPNNのネットワークパラメータの設定はCPと同一である。CPNN、VCPNN共にKohonen層を並列に3つ用意し、各ネットワークの出力にはそれらの平均値を用いる。

3.3 時系列予測問題

時系列の予測問題として、一般にリカレントニューラルネットワーク³⁾やRBFネットワーク⁵⁾が利用されている。今回、我々はSGNNを実変数写像を行う時系列予測問題に対して適用する。本研究では、式(7)のMackey-Glass微分遅延方程式¹³⁾によって生ずるランダム信号 $x[t]$ を用いる。

$$\frac{dx[t]}{dt} = -bx[t] + a \frac{x[t-\tau]}{1+x[t-\tau]} \quad (7)$$

入力訓練データを、 $(x[t], x[t-\Delta], x[t-2\Delta], x[t-3\Delta])$ とし、その出力目標値として、 $x[t+T]$ を与える。ここで、 $\Delta=6$ 、 $T=85$ である。Mackey-Glass微分遅延方程式における諸設定値は $a=0.2$ 、 $b=0.1$ 、 $\tau=17$ としてデータを生成した。また、訓練データ数として、500、1000、3000、5000個の4パターンを考慮し、既存の学習則としてBP、CP、VCPを使用する。

SGNNに関して、入力訓練データの提示順をランダムに入れ換えて各10回試行を行う。各訓練データ数に対するSGNNの規模を調べるため、木の深さを求める。BPに関して、ネットワークの構造は入力層4ユニット、中間層10ユニット、出力層1ユニットの3層構造であり、結合重みの初期値を-1から1までランダムに与えて10回試行を行う。学習係数 η は0.9とする。CPNNに関して、Kohonen層のユニット数を100個、Grossberg層のユニット数を100個とする。Kohonen層の学習係数 α 、Grossberg層の学習係数 β をそれぞれ0.1とする。VCPNNに関して、VUのしきい値を0.1、非競合ユニットの最大増減値を100とする。その他、VCPNNのユニット数はCPNNと同一である。CPNN、VCPNN共にKohonen層を並列に3つ用意し、各ネットワークの出力にはそれらの平均値を用いる。テストデータは、式(7)により計算した1000個の訓練データとは独立な時系列データを使用し、予測精度の指標にテストデータに対する正規化平均二乗誤差 (Normalized Mean Squared Error: NMSE) を用いる。

$$NMSE = \frac{\sum_{p=1}^P (y'_p - y_p)^2}{\sum_{p=1}^P (y_p - E[y_p])^2} \quad (8)$$

式(8)において、 y_p は各入力データに対する出力目標値であり、 y'_p はネットワーク出力値である。そして、 P は全訓練データ数である。

4. 実験結果

4.1 分類問題

表2にSGNN、LVQ1、CPNN、VCPNN、BPNNの各ベンチマーク問題に対する実験結果を示す。なお、CPNN、VCPNN、およびBPNNは正解率の最も高かった結果と、そのときのパラメータを示している。また、BPNNに関して、正解率が同値である場合は学習終了時のMSEが最小の結果を掲載している。

表2より、各問題に対してSGNNが最も短い処理時間で終了しているのがわかる。この特徴は、SGNNが訓練データを1回提示するだけで入力特徴空間を木構造内に構成するためである。また、CPNN、VCPNNに関しては、単一のCPUで3つのネットワークを処理しているため、他の1つのネットワークによるものよりも処理時間を要している。

次に学習法と正解率の観点からSGNNと各手法を比較する。まず、SGNNと同様に競合学習に基づくLVQ1アルゴリズムと比較すると、Card1、Card3以外のほとんどの問題では、SGNNが良い結果を示す。

表 2 分類問題に対する実験結果

		M_1	M_2	M_3	Cancer1	Cancer2	Cancer3	Card1	Card2	Card3
SGNN	CR	81.2%	72.0%	83.3%	96.5%	96.0%	96.5%	74.4%	80.8%	73.8%
	CT	0.1	0.13	0.1	0.27	0.24	0.24	0.56	0.52	0.57
LVQ1	CR	70.6%	64.6%	74.8%	96.0%	96.0%	96.0%	82.6%	79.1%	76.7%
	CT	0.23	0.23	0.23	0.31	0.31	0.3	0.92	0.91	0.86
CPNN	CR	80.8%	73.8%	77.1%	93.1%	87.9%	85.6%	72.1%	68.0%	72.1%
	CT	180.65	574.27	299.44	1062.43	1052.41	994.39	1605.12	1045.11	852.22
	α	0.5	0.9	0.02	0.5	0.3	0.01	0.4	0.5	0.08
	β	0.04	0.06	0.04	0.2	0.2	0.2	0.03	0.02	0.02
VCPNN	CR	87.0%	77.1%	75.9%	90.8%	88.5%	89.7%	81.4%	79.7%	77.9%
	CT	38.32	79.71	14.87	1001.48	65.62	43.18	6553.94	6628.24	1652.64
	α	0.7	1.0	0.04	0.8	0.5	0.09	0.01	0.04	0.6
	β	0.01	0.01	0.01	0.2	0.01	0.01	0.3	0.3	0.03
BPNN	CR	98.8%	78.2%	91.7%	97.7%	94.3%	95.9%	85.5%	81.4%	84.9%
	CT	1.25	35.3	4.0	89.12	73.88	87.55	712.68	751.16	898.49
	HU	5	35	10	35	30	35	70	80	90

CR: Correct Rate, CT: CPU Time (in sec.), HU: Hidden Units

表 3 SGNN と他の学習法による認識率

noise	5	10	15	20	25	30	35	40	45	50
SGNN	100.0%	100%	100%	92.3%	96.2%	84.6%	46.2%	26.9%	11.5%	0.0%
BPNN	96.5%	79.2%	55.0%	29.2%	16.5%	6.2%	4.2%	1.9%	0.8%	0.8%
CPNN	46.2%	46.2%	46.2%	42.3%	23.1%	15.4%	3.8%	0.0%	0.0%	0.0%
VCPNN	100%	100%	100%	92.3%	50.0%	7.7%	0.0%	0.0%	0.0%	0.0%

クラス情報を用いてコードベクトルを学習する教師あり学習の LVQ1 と同程度の正解率を得ることができることは、教師なし学習である SGNN が優れたクラスタリング能力を持つことを意味している。次に CPNN と正解率を比較すると、 M_2 を除く他のすべての問題に対して SGNN のほうが高い精度を得ることができる。また、VCPNN と比較すると Monks, Card の問題に対しては VCPNN のほうが全体的に高く、Cancer の問題に対しては SGNN のほうが高い正解率を示す。最後に、教師あり学習である BPNN と比較すると、Card の問題では同程度の正解率であるが、Monks, Card の問題では、Card2 を除くすべての問題で約 1 割程度精度が劣ることがわかる。

4.2 パターン認識問題

表 3 に SGNN, CPNN, VCPNN, BPNN における文字認識問題の各雑音率に対する認識率を、表 4 に学習訓練時間示す。表 3, 表 4 に関して、BPNN では、中間層のニューロン数を変化させて最も認識率の高かった 30 のときの結果を、CPNN, および VCPNN では、学習係数を変化させ 400 通りの試行により、最大の認識率を示した場合を掲載している。

表 4 より、入力データから自動的にニューラル木を作成する SGNN は、高速な学習が可能であることがわかる。更に、表 3 より、SGNN が他の手法よりもすべての試行において優れた認識率を示していることがわかる。

4.3 時系列予測問題

表 5 に 10 回の試行における各訓練データ数の変化に対する SGNT の深さ (最小, 最大, 平均, 分散) を、表 6 に SGNN による各訓練データ数の変化に対する予測結果として、NMSE (最小, 最大, 平均) と処理時間を示す。表 7 に BPNN による予測結果を、表 8 に CPNN による予測結果を、表 9 に VCPNN による予測結果を表 6 と同じ形式で示す。なお、処理時間は、10 回の試行に対する訓練とテストの総処理時間の平均値である。

表 5, 表 6 より、訓練データが増加するのに比例して、木の規模が増加していることがわかる。また、訓練データが増加するのに比例して NMSE が減少していることがわかる。同一の訓練データ数の場合、分散が小さいことから、木の深さは同程度となり、与えられた入力データによる特徴空間を同程度の木構造で写像していることがわかる。よって、NMSE の値も同程度の値となっている。表 6, 表 7, 表 8, 表 9 より

表4 訓練時間の比較

Learning Method	Learning Time (in second)
SGNT	0.05
BP	11.34
CP	1.75
VCP	0.96

表5 SGNTの深さ

Data	max.	min.	avg.	var.
500	304	263	290.5	11.1
1000	608	560	582.3	11.6
3000	1748	1721	1738.1	9.87
5000	2955	2896	2920	17.4

表6 SGNNによる予測結果

Data	min.	max.	avg.	CT (sec.)
500	0.146	0.261	0.186	0.24
1000	0.093	0.143	0.112	0.34
3000	0.025	0.036	0.031	0.76
5000	0.019	0.023	0.021	1.2

表7 BPNNによる予測結果

Data	min.	max.	avg.	CT (sec.)
500	0.944	1.014	0.984	0.88
1000	0.944	1.093	0.968	1.69
3000	0.944	0.985	0.953	2.75
5000	0.944	0.946	0.944	4.37

表8 CPNNによる予測結果

Data	min.	max.	avg.	CT (sec.)
500	0.408	0.501	0.461	1.54
1000	0.532	0.602	0.574	2.75
3000	0.535	0.569	0.553	7.64
5000	0.488	0.521	0.505	12.56

表9 VCPNNによる予測結果

Data	min.	max.	avg.	CT (sec.)
500	0.211	0.241	0.234	1.37
1000	0.268	0.322	0.29	2.62
3000	0.255	0.276	0.267	7.41
5000	0.277	0.291	0.284	12.22

SGNN と他のニューラルネットワークとの比較を行うと、精度は全てのデータ数で SGNN が優れた結果を示している。SGNN の場合、訓練データ数を増加させることでより高い予測精度を得ることができるのに対し、BP、CP、VCP では、データ数を増やしても予測精度の変化はない。これは、SGNN の精度が訓練データ数に依存していることを意味する。次に処理時間において、SGNN が他の手法よりも高速である。

図2 に SGNN による時系列予測波形の一例として、訓練データ数 N が 500 の場合の SGNN、BPNN、CPNN、VCPNN によるテストデータに対する予測波形の一部を示す。SGNN の場合、他の手法に比べ良好な予測を行えていることがわかる。

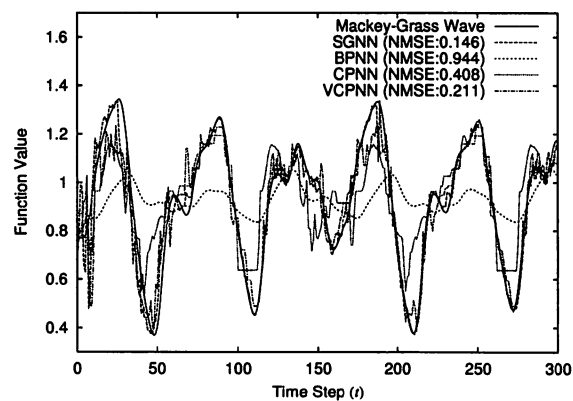


図2 予測波形の一部

5. むすび

本研究では、実時間学習を行なう自己生成ニューラルネットワーク (Self-Generating Neural Networks, SGNN) の特性を分析するために、ベンチマーク問題の分類、パターン認識、時系列予測に適用した。実験結果より、全分野の問題に対して高速な学習を行い、入力訓練データから SGNT 内に特徴空間を写像することがわかった。更に、分類問題以外の実変数写像を行なうような時系列の予測問題にも利用可能であることがわかった。各問題を通じて確認できた SGNN の有効な特性を以下に示す。

高速性：与えられた訓練入力信号から、競合学習により、入力特徴空間を木構造内に動的に構築するため、何度も繰り返し訓練データを提示する他の手法に比べ、高速な学習を行う。

大規模な問題への適用性：訓練データ数が大量に存在し、各訓練データ内の属性値の次元数が大きくても訓練データと結合するノードからなる SGNT を記憶しておくメモリ領域さえあれば、さまざまな問題に対して学習が可能である。

簡易性：従来の手法では、各問題の規模に応じて、我々が静的なネットワーク構造、およびパラメータを決めて学習を行う。良い精度の結果を得るためには、与えられた問題を NN に実装するための知識や経験が必要である。SGNN では、ネットワーク構造、パラメータは学習中に自動的に決定するので、あらゆる問題に対して柔軟に対処できる。故に、我々は訓練データを提示してやるだけでよく、その問題の実装に関する知識、経験を必要としない。

以上のことから、SGNN は高速学習が可能で汎化性に優れたネットワークであるといえる。しかしながら、出力値は有限の訓練データ内に存在する出力値に限定されるため、実変数写像を行う時系列予測や関数近似の問題に対する精度を向上するためには、SGNT に入力特徴空間をより正確に写像させるために、大量のデータが必要である。処理をより効率的にするためのアプローチとして、処理の並列化、大量データの分散化などが考えられる。

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, USA, 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) J. T. Connor and R. D. Martin. Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, Vol. 5, No. 2, pp. 240-254, mar 1994.
- 4) B. Freisleben and G. Gleichmann. Controlling airline seat allocations with neural networks. In P. K. Simpson, editor, *Neural Networks Theory, Technology, and Applications*, pp. 655-661. IEEE, New Jersey, 1989.
- 5) 福田典明, 成久洋之. RBF を用いた NN による時系列予測について. 情報処理学会 技術報告 数理モデルと問題解決 17-2, 情報処理学会, Jan 1998.
- 6) S. Haykin. *Neural Networks: A comprehensive foundation*. Prentice-Hall, second edition, 1999.
- 7) R. Hecht-Nielsen. Counterpropagation networks. *Applied Optics*, Vol. 26, No. 23, pp. 4979-4984, December 1987.
- 8) 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. 信学技法 562, 電子情報通信学会, Jan 1999.
- 9) 井上浩孝, 成久洋之. ニューラルネットにおける自己生成学習特性について. 平成 9 年度電気・情報関連学会中国支部連合大会講演論文集, p. 477, 1998.
- 10) Hiroataka Inoue and Hiroyuki Narihisa. Performance of self-generating neural network applied to pattern recognition. In *ISAS'99 (5th International Conf. on Information Systems Analysis and Synthesis)*, Vol. 5, pp. 608-614, Orlando, Florida, U.S.A., Aug. 1999.
- 11) T. Kohonen. 自己組織化マップ, 第 2 章. シュプリンガー・フェアラーク東京, 1996.
- 12) 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.

- 13) M. C. Mackey and L. Glass. Oscillation and chaos in physiologist control systems. *Science*, Vol. 197, pp. 287–289, 1977.
- 14) L. Prehelt. PROBEN1 – a set of neural network benchmark problems and benchmarking rules. Technical report 21/94, Universität Karlsruhe, 1994.
- 15) 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.
- 16) S. Thrun, et al. The MONK's problems: A performance comparison of different learning algorithms. Technical report CMU-CS-91-197, Carnegie Mellon University, 1991.
- 17) W. X. Wen, A. Jennings, and H. Liu. Learning a neural tree. In *IJCNN'92 (International Joint Conf. on Neural Networks)*, Beijing, China, 1992.
- 18) 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.

Performance Analysis of 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 4, 1999)

Self-generating neural networks (SGNN), which are developed from the concept of Self-Organizing Neural Networks (SONN), have received much attention because of their design simplicity and high accuracy for clustering or classification problems. In this paper, we present a performance analysis of SGNN. In order to analyze the performance of SGNN, we compare them with other existing competent neural networks in terms of accuracy and processing time on classification problems, pattern recognition, and time series prediction. Results show that SGNN offer superior performance to other existing neural networks in these various applications.