アンサンブル自己生成ニューラルネットワークのための高速枝刈り法

井上浩孝[†,☆] 成久洋之[††]

自己生成ニューラルネットワーク（self-generating neural networks: SGNN）は与えられた訓練データセットより自動的に自己生成ニューラル木（self-generating neural tree: SGNT）を構築することで高速処理を行う．我々は，SGNN の汎化誤差を減少させるため，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．ESGNN は単一の SGNN に比べ高い精度を得られるが，計算コスト（記憶容量，計算時間）はアンサンブル学習に使用する SGNN の数が増えるにつれて増大する．本論文では，分類に対する記憶容量を削減するための新しい ESGNN の枝刈り法を提案する．UCI 機械学習データベースの 10 のデータセットに対し，枝刈り後の SGNN を基本予測器とする ESGNN と最近傍法によるモデルの解の精度，メモリ使用量，計算時間を比較する．実験結果より，本提案手法はメモリ使用量を枝刈り前の 30%から 90%削減し，最近傍法を用いたモデルに比べ高い精度が得られることを示す．さらに，2 つのサンプリング法による枝刈り後の ESGNN の性能評価を行う．

A Fast Pruning Method for Ensemble Self-generating Neural Networks

Hirotaka Inoue[†,☆] and Hiroyuki Narihisa[††]

Self-Generating Neural Networks (SGNN) have a feature of fast processing by automatically constructing self-generating neural tree (SGNT) from given training data set. In our previous work, we introduced an ensemble model called ESGNN (ensemble self-generating neural networks) which can be used to reduce the generalization error of a single SGNN. Although this model can obtain the high accuracy than a single SGNN, the computational cost increase in proportion to the number of SGNN in the ensemble learning. In this paper, we propose a new pruning method for ESGNN to reduce the memory requirement for classification. We compare ESGNN with nearest neighbor classifier using a collection of ten benchmark problems in the UCI machine learning repository. Experimental results show that our method could reduce the memory requirement from 30% to 90% of unpruned ESGNN and improve the accuracy over the accuracy of the nearest neighbor classifier. A performance analysis of pruned ESGNN on two sampling methods is also discussed.

1. はじめに

大量のデータから特徴を抽出し，未学習データに対する入出力関係を予測する数理モデルには，(i) 高い解の精度と，(ii) 取扱いが容易であるという性質が望まれている[9]．ニューラルネットワークは分類，クラスタリング，予測，パターン認識等の知的処理の分野で幅広く用いられている数理モデルである[4]．ニューラルネットワークは生物学，物理学，統計学，計算機科学の分野からさまざまな手法が提案されているが，現在最も広く利用されているのはバックプロパゲーション（backpropagation: BP）[13] 学習を行う階層型ニューラルネットワークである．

階層型ニューラルネットワークは順応性や柔軟性に優れ，非線形入出力写像能力を持つ．しかしながら，中間層数や中間層内の各ユニット数，学習係数等を各設計者が問題の規模や複雑度に応じて決定しなければならない．ユニットを増やすことはネットワークのパラメータの自由度を増すこととなり表現能力を高める

† 岡山理科大学大学院工学研究科
Graduate School of Engineering, Okayama University of Science
†† 岡山理科大学工学部情報工学科
Department of Information & Computer Engineering, Faculty of Engineering, Okayama University of Science
☆ 現在，呉工業高等専門学校電気情報工学科
Presently with Department of Electrical Engineering and Information Science, Kure National College of Technology

ことができるが，問題の規模以上に自由度を増やしすぎるとオーバフィッティングを引き起こす．したがって，ネットワークの正しい構造を求めることはニューラルネットワークの学習において大変重要な要素であり，ニューラルネットワークの実装における最も難しい問題である[17]．そして，与えられた訓練データを繰り返しネットワークに入力し，各データに対する所定の期待出力とネットワークの誤差を最小にするように結合重みの修正を行うため，学習に多大な時間がかかる．したがって，数理モデルに望まれる性質の（i）は満たすものの，（ii）は満たしていない．

一方，自己生成ニューラルネットワーク（self-generating neural networks: SGNN）[15] 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 は代表的な教師なし学習モデルである自己組織化地図（self-organizing maps: SOM）[8] を拡張したものであり，訓練データを一度提示することで競合学習により自動的に自己生成ニューラル木（self-generating neural tree: SGNT）を生成する．このモデルは数理モデルの性質（ii）を満たしているものの，（i）の解の精度は階層型ニューラルネットワークに劣る[5]．

SGNN の解の精度を改善するため，我々は同一の訓練データから生成した複数の SGNN の出力の平均を全体の出力とす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[6]．ESGNN は階層型ニューラルネットワークと同程度の解の精度をより高速に算出することが可能である[6]．しかし，計算時間および記憶容量は使用する SGNN の数が増加するにつれて増大する．

本論文では，記憶容量を削減し，汎化能力改善を行うための SGNN の枝刈り法を提案し，枝刈りを行った SGNN を基本予測器とするアンサンブルモデルの性能を検討する．UCI 機械学習リポジトリ[1] の 10 のデータセットを用いて，訓練データの提示順をランダムに入れ換えて生成した枝刈り SGNN，bagging[2] によって生成した枝刈り SGNN による ESGNN の記憶容量削減率，汎化誤差改善特性を比較検討する．また，比較対象とする既存の数理モデルとして，最近傍（nearest neighbor: 1-NN）法とその拡張の k 最近傍（k-nearest neighbor: k-NN）法によるモデルを用いて同一環境で実験を行う．

本論文の構成は以下のとおりである．まず 2 章で SGNN を構築するための SGNT の生成法を，3 章で解の精度を改善するためのアンサンブル学習法について説明する．次に 4 章で記憶容量を削減するための SGNT の枝刈り法について説明し，5 章で計算機を用いた数値実験を行う．最後に 6 章でまとめを述べる．

2. 自己生成ニューラル木の生成法

SGNN は SOM[8] と同様に競合学習を用いて SGNT に与えられた訓練データセット $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}$，$|\mathcal{D}| = N$ 内のデータを動的に組み込む[16]．ここで，$\boldsymbol{x}_i \in \Re^m$ は入力ベクトルで y_i は $\boldsymbol{x}_i$ に対する期待出力を表す．分類問題において，期待出力は $y_i \in \mathcal{C}$，$|\mathcal{C}| = C$ であり，クラスを示すラベルである．以後，このラベルを class k ($k = 0, \ldots, C-1$) で表す．

SGNT は根，節点，葉からなる木構造のモデルである．根は SGNT 内に 1 つだけ存在し，入力データが提示される部分である．葉は未学習データに対する出力候補となる部分であり，競合学習により選択される．根と葉を結ぶ部分に節点が存在する．この節点は SGNT の生成過程で自動的に生成される．任意の節点において，根の方向に結合している節点または根をその節点の親とし，葉の方向に結合している節点または葉をその節点の子とする．根，節点，葉にはそれぞれ m 次元重みベクトル $\boldsymbol{w}$ を参照ベクトルとするニューロンが存在する．これらのニューロンを記号 n で表す．以後，各ニューロンを識別するために添字 j を付けて表す．**図 1** に擬似 C 言語による SGNT 生成アルゴリズムを，**表 1** に SGNT 生成法で使用される副手

Input：
- A set of training examples $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}$, $i = 1, \ldots, N$.
- A threshold value $\xi \geq 0$.
- A distance measure $d(\boldsymbol{x}_i, \boldsymbol{w}_j)$.

Method：

```
1  copy(n_1, x_1);
2  for  (i = 2, j = 2; i <= N; i++) {
3    n_win  = choose(x_i, n_1);
4    minDistance = distance(x_i, w_win);
5    if  (minDistance > ξ) {
6      if (leaf(n_win))  {
7        copy(n_j, w_win);
8        connect(n_j, n_win);
9        j++;
10     }
11     copy(n_j, x_i);
12     connect(n_j, n_win);
13     j++;
14   }
15   update(x_i, w_win)
16 }
```

Output：Constructed SGNT by $\mathcal{D}$.

図 1 SGNT 生成アルゴリズム
Fig. 1 The SGNT algorithm.

表 1 SGNT 生成で用いられる副手続き
Table 1 Sub procedures of the SGNT algorithm.

Sub procedure	Specification
copy(n_j,$\boldsymbol{x}_i$)	Create n_j, copy $\boldsymbol{x}_i$ as $\boldsymbol{w}_j$ in n_j.
distance($\boldsymbol{x}_i$,$\boldsymbol{w}_j$)	Compute $d(\boldsymbol{x}_i, \boldsymbol{w}_j)$.
choose($\boldsymbol{x}_i$,n_1)	Decide n_{win} for $\boldsymbol{x}_i$.
leaf(n_{win})	Check n_{win} whether it is a leaf.
connect(n_j,n_{win})	Connect n_j as a child of n_{win}.
update($\boldsymbol{x}_i$, $\boldsymbol{w}_j$)	Update $\boldsymbol{w}_j$ of n_j.

続きの説明を示す．

生成過程において，競合学習により訓練実例ベクトル $\boldsymbol{x}_i$ に対する SGNT 内の勝者ニューロン n_{win} を決定する．根から n_{win} に至る節点に存在するニューロン n_j の重み w_{jk} は次式を用いて更新する：

$$w_{jk} \leftarrow w_{jk} + \frac{1}{c_j + 1} \cdot (x_{ik} - w_{jk}) \ , \qquad (1)$$
$$1 \le k \le m \ .$$

ここで，c_j は n_j に含まれる葉の数を表す．SOM の場合，学習係数の初期値は学習前に任意に設定し，訓練データセット $\mathcal{D}$ を繰り返し提示するごとに単調減少させる．これに対して，SGNN では全訓練データを SGNT の葉として直接組み込むことで学習を行い，式 (1) によりその節点に含まれる葉の持つ m 次元入力ベクトルの各要素の平均値となるように m 次元重みベクトルを修正する．ゆえに，c_j が学習中に動的に変化することで学習係数が適応的に決定するため，ネットワークの設定およびパラメータの設定が不要である．さらに，SGNN は $\mathcal{D}$ を 1 回提示するだけで学習を終了するため，繰り返し訓練データを提示して学習を行う既存のニューラルネットワーク学習法に比べ高速な学習特性を持つ．また，入力ベクトル $\boldsymbol{x}_i$ に対する期待出力 y_i を SGNT の出力 o_i として葉に与える．最終的に構築された SGNT により，与えられた問題に対する m 次元特徴空間を写像する．

テスト過程において，未学習テストデータセット $\mathcal{T} = \{(\boldsymbol{x}_i, y_i)\}$，$|\mathcal{T}| = M$ の $\boldsymbol{x}_i$ を SGNT の根に入力する．訓練と同様に根より各階層ごとに再帰的に競合学習を行い，到達した葉の持つ出力 o_{win} を SGNN の出力とする．ここで，訓練時は子とその親のニューロンを候補として根から葉へ再帰的に競合学習を行うのに対して，テスト時は SGNN の出力が存在する葉まで到達させるため，親を除いた子のニューロンを候補として競合学習を行う．

3. アンサンブル学習法

訓練データセット $\mathcal{D}$ から複数の予測器の出力を用いて解の精度を改善するためのアンサンブル学習法は，

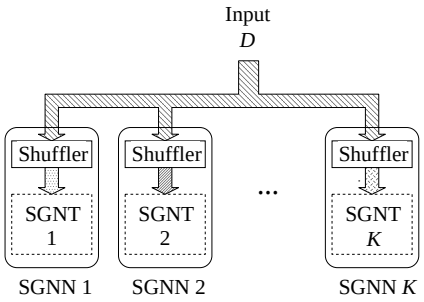

図 2 K 個の SGNT による ESGNN（訓練過程）
Fig. 2 ESGNN of K SGNTs (training phase).

ニューラルネットワークにおいて Perrone らによって報告された[10]．また，分類木の解の精度を改善するため，Breiman は統計学の bootstrap サンプリングに基づく bagging（bootstrap aggregating の略）と呼ばれるサンプリング法を提案した[2]．また，機械学習の分野で逐次訓練データセットを提示するラウンドごとに学習しにくいデータを多く学習させるようにデータの確率分布を変化させて効率的な汎化能力改善を行う boosting という手法が提案された[3]．一般に，bagging と boosting を比較すると，boosting を行った方が高い汎化能力の改善率が得られるが，逆に単一の予測器よりも精度が落ちることがある．これに対し，bagging は改善率は boosting に劣るものの，安定して解の精度を改善する[11]．また，並列分散処理を考慮した場合，boosting は階層型ニューラルネットワークの学習と同様に，逐次訓練データを提示する必要があるため，並列分散処理との親和性は低い．これに対し，bagging は各予測器を独立に生成させることが可能であるため，並列分散処理との親和性が高い．

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[6]．SGNT の構造は学習中に動的に変化する．SGNT アルゴリズムはすべての訓練データが加えられた後に木構造を決定する．異なる木構造を持つ 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．以後，訓練データを入れ換えて作成した入力データを用いる手法を “shuffling” とよぶ．本研究では，shuffling と bagging を用いた ESGNN による実験を行う．

訓練過程において，N 個の要素からなる訓練データセット $\mathcal{D}$ は shuffling と bagging を用いて K セッ

ト用意され，K 個の SGNT からなる ESGNN を生成する（**図 2**）．テスト過程において，M 個の要素からなるテストデータセット $\mathcal{T}$ の $\boldsymbol{x}_i$ を ESGNN に入力する．各 SGNT の $\boldsymbol{x}_i$ に対する葉の持つクラスの多数決をとり，最も多かったクラスをそのテストデータに対する出力とする．

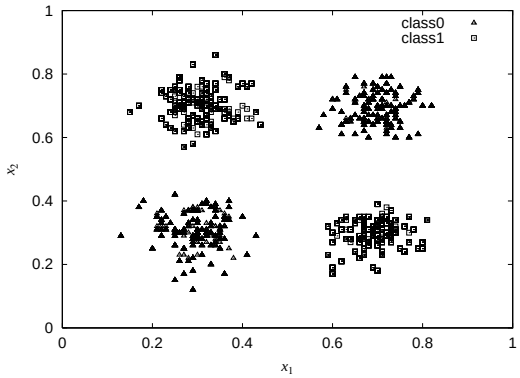

図 3 2 次元 2 クラス問題
Fig. 3 A two dimensional binary classification problem.

アンサンブル学習による SGNN の汎化能力改善の例を示す．ここで扱う問題は 2 次元 2 クラス問題でクラス 0（△）とクラス 1（□）はそれぞれ 200 個のデータからなり，対称的に分布している（**図 3**）．この問題に対して，単一 SGNN による識別領域の例を**図 4** (a)，図 4 (b)，比較対象とする既存モデルとして，最近傍法による識別結果を図 4 (c) に，10 個の SGNN を基本予測器とした ESGNN の識別結果を図 4 (d) に示す．これらの図において，影付きの領域がクラス 0 と判定した領域を表す．図 4 (a)，図 4 (b) は shuffling により SGNN を構築した．クラス 0 とクラス 1 の識別境界が大きく異なることが確認できる．また，図 4 (c) と比較してその境界がずれていることが分かる．しか

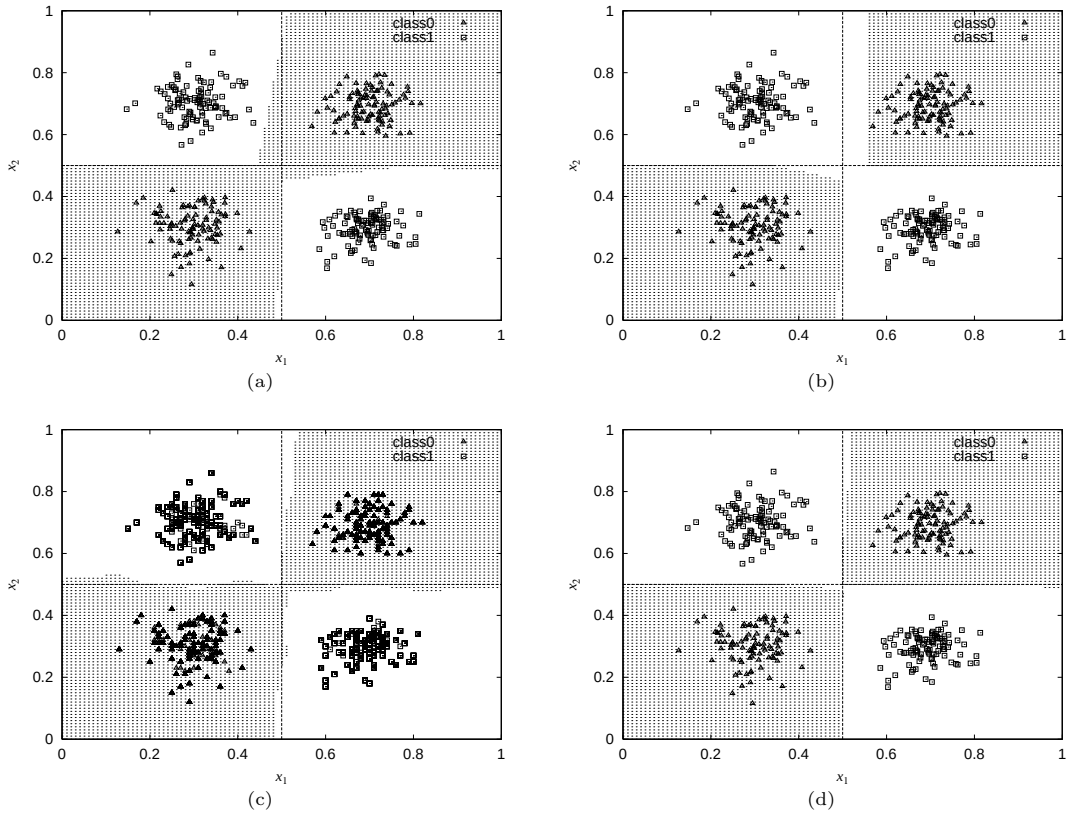

図 4 2 次元 2 クラス問題に対する識別結果，(a) SGNN による結果の一例（1），(b) SGNN による結果の一例（2），(c) 最近傍法による識別結果，(d) 10 個の SGNN を基本識別器とした ESGNN による識別結果．影付きの平面は各識別器によってクラス 0 と見なされた決定領域を表す

Fig. 4 An example of the classification results for the two dimensional binary classification problem, an example of the result of (a) SGNN (1), (b) SGNN (2), (c) nearest-neighbor classifier, and (d) ESGNNs ($K = 10$). The shaded plane is the decision region of class 0 by each classifier.

```
1 begin initialize j = the height of the SGNT
2   do for each subtree's leaves in the height j
3     if all leaves belong to the same class,
4     then merge all leaves to parent node
5     if all subtrees are traversed in the height j,
6     then j ← j − 1
7   until j = 0
8 end.
```

図 5 マージアルゴリズム
Fig. 5 The merge algorithm.

```
1 begin initialize j = 0, N =# of leaves on the
merged tree
2   do j ← j + 1; for each attribute x_j
3     search the nearest leaf of x_j
4     if the nearest leaf differ from the original leaf,
5     then prune the original leaf as the dead leaf
6   until j = N
7 end.
```

図 6 枯葉削除アルゴリズム
Fig. 6 The dead leaf pruning algorithm.

しながら，図 4 (c) と図 4 (d) を比較すると，ほぼ同様な識別領域を構成していることが分かる．このように，同一訓練データセットで学習後，識別領域が大きく異なる識別器では，複数の識別器を用いてアンサンブル学習を行うことで各クラス間の決定境界付近の解の精度を改善することが可能である．

4. 自己生成ニューラルネットワークの枝刈り

ESGNN は単一の予測器に比べ，複数の予測器を用いることで未学習データに対する解の精度を改善することが可能であるが，アンサンブルに使用する SGNN の数が増加するにつれて計算時間，記憶容量を消費する．そこで，分類問題に対する ESGNN の記憶容量を削減するため，SGNN の枝刈りを行う．

全データが提示された後，木の枝刈りを行う．木の生成は訓練データセット $\mathcal{D}$ を根から葉の方向へトップダウンに学習を行うが，枝刈りは葉から根に向かってボトムアップに行う．本手法はマージアルゴリズムと枯葉削除アルゴリズムの 2 つの操作を交互に行う．以下各操作について説明する．

マージアルゴリズムでは，木の深さ優先探索により段階的にマージする葉を決定する．もし各階層中で同一の親を持つ部分木ですべての葉のクラスが一致すれば，その親にそれらのクラスを与え，新たな葉としてマージし，その階層内の葉をすべて刈り取る．この作業を根へ向かって再帰的に行うことで特徴空間内で同一クラスの密度の高い部分の冗長なデータを削減する（**図 5**）．

枯葉削除アルゴリズムでは，マージアルゴリズムで単純化された SGNT のすべての葉の持つ実例データを木の根から再度入力する．もしそのデータが元の葉と同じ場所に到達しないのであれば，元の葉を枯葉と見なし削除する（**図 6**）．枯葉がなくなるまでマージアルゴリズムと枯葉削除アルゴリズムを繰り返すことで枝刈り前の SGNT の分類能力を維持しながら記憶容量を削減する．

図 7 に 2 次元 2 クラス問題に対する SGNT の枝刈りと識別領域の例を示す．クラス 0 は平均 0.3，標準偏差 0.05 の正規分布に従う 100 個の 2 次元データからなり，クラス 1 は平均 0.7，標準偏差 0.05 の正規分布に従う 100 個の 2 次元データからなる（図 7 (a)）．各図において，影付きの平面は SGNT により得られたクラス 0 の決定領域を表している．図 7 (b) は枝刈り前の SGNT を表している．この場合，高さ 0 の部分に根（○），200 個の葉（クラス 0：△，クラス 1：□）と 117 個の節点（○）が SGNT アルゴリズムにより自動的に生成された．なお，枝刈り前の木の高さは 8 である．SGNT が構築された後，マージアルゴリズムを実行する．図 7 (c) はマージアルゴリズムを一度実行した後の SGNT を示している．この例では，枝刈り前と同じ決定領域を維持しつつ，199 個の葉と 106 個の節点が刈り取られ，根，11 個の葉と 1 個の節点を持つ高さ 2 の木となる．次に，枯葉削除アルゴリズムを実行する．枝刈り後に 1 つだけ残っていた葉（図 7 (c) の△）のデータが高さ 1 の別の葉に到達し，枯葉となる．この葉を削除して，もう一度マージアルゴリズムを実行すると，図 7 (d) のように，根と 2 個の葉を持つ高さ 1 の枝刈り SGNT が得られる．ここで枯葉削除アルゴリズムを実行すると枯葉は存在しないので枝刈りを終了する．今回，簡単な具体例を示すために 2 次元 2 クラスの 200 個のデータを用いた．この場合，記憶容量は，枝刈り前を 1.0 とすると，$3/318 \approx 0.009$ となり，99%の削減がなされたことになる．

ノイズに対する枝刈り法の頑健性を調べるため，**図 8** にデータがオーバラップした 2 次元 2 クラス問題に対する SGNT の枝刈りと識別領域の例を示す．クラス 0 は平均 0.45，標準偏差 0.05 の正規分布に従う 100 個の 2 次元データからなり，クラス 1 は平均 0.55，標準偏差 0.05 の正規分布に従う 100 個の 2 次元データからなる．影付きの平面は SGNT により得られたクラス 0 の決定領域を表している．図 8 (a) は根，200 個の葉，115 個の節点からなる枝刈り前の SGNT を，図 8 (b) は根，40 個の葉，36 個の節点からなる枝刈

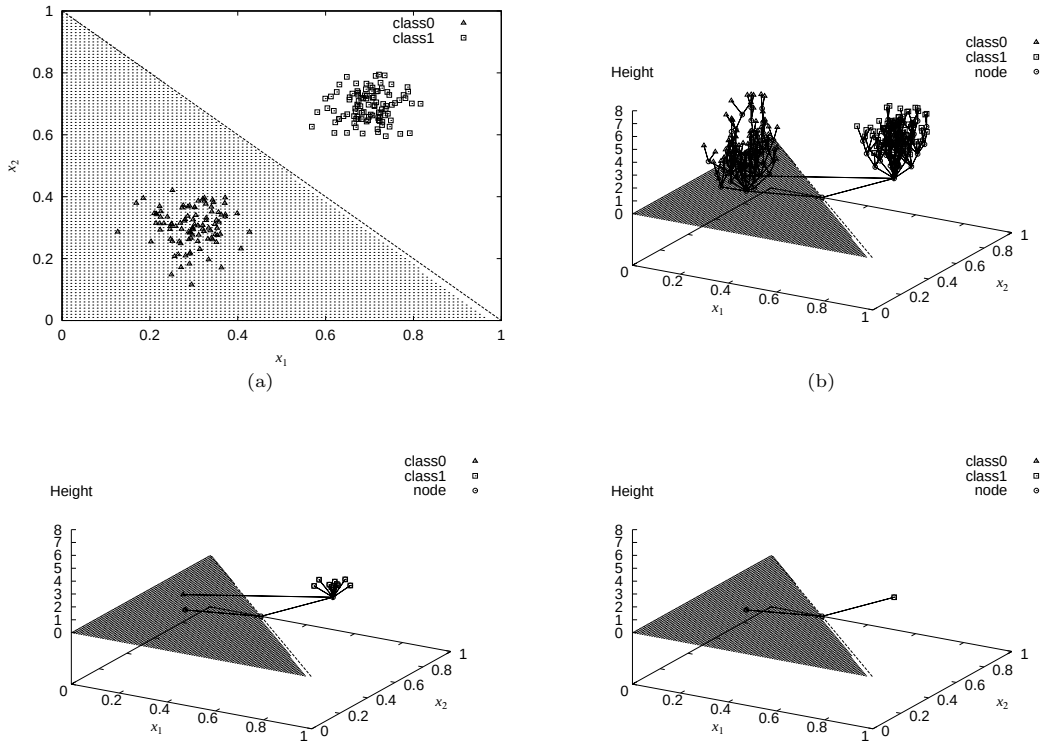

図 7 2 次元 2 クラス問題に対する SGNT 枝刈りの例，(a) 2 次元 2 クラス問題，(b) 枝刈り前の SGNT，(c) マージアルゴリズムを一度実行した後の SGNT，(d) 枝刈り後の SGNT．影付きの平面は SGNT によってクラス 0 と見なされた決定領域を表す

Fig. 7 An example of the SGNT pruning algorithm, (a) a two dimensional classification problem with two equal circular Gaussian distribution, (b) the structure of the unpruned SGNT, (c) the structure of the pruned tree after the merge algorithm, and (d) the structure of the final pruned SGNT. The shaded plane is the decision region of class 0 by the SGNT.

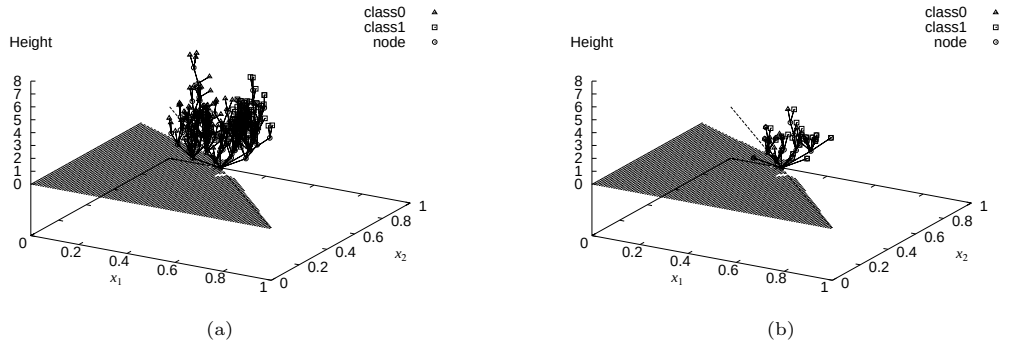

図 8 データがオーバラップしている 2 次元 2 クラス問題に対する SGNT 枝刈りの例，(a) 枝刈り前の SGNT，(b) 枝刈り後の SGNT．影付きの平面は SGNT によってクラス 0 と見なされた決定領域を表す

Fig. 8 An example of the SGNT pruning algorithm, this example have the overlap of data, (a) the structure of the unpruned SGNT, and (b) the structure of the final pruned SGNT. The shaded plane is the decision region of class 0 by the SGNT.

表 2 shuffled SGNN，bagged SGNN，最近傍法に対する 10 回の試行における平均記憶容量と計算時間 (s)
Table 2 The average memory requirement and the computation time (s) of ten trials for shuffled SGNN, bagged SGNN, and nearest neighbor.

	shuffled SGNN		bagged SGNN		nearest neighbor	
Dataset	memory	time(s)	memory	time(s)	memory	time(s)
balance-scale	0.384	0.08	0.304	0.07	0.664	0.4
breast-cancer-w	0.125	0.11	0.093	0.10	0.703	0.63
glass	0.582	0.03	0.459	0.02	0.654	0.05
ionosphere	0.423	0.10	0.299	0.09	0.697	0.26
iris	0.196	0.02	0.153	0.01	0.657	0.03
letter	0.290	15.91	0.239	14.14	0.677	979.23
liver-disorders	0.690	0.06	0.483	0.05	0.670	0.15
new-thyroid	0.270	0.02	0.235	0.02	0.654	0.05
pima-diabetes	0.570	0.15	0.428	0.14	0.672	0.72
wine	0.238	0.02	0.179	0.01	0.703	0.05
Average	0.376	1.65	0.282	1.47	0.675	98.16

り後の SGNT を表している．枝刈り前の決定領域を維持しつつ，冗長な葉，節点が刈り取られているのが確認できる．記憶容量は，枝刈り前を 1.0 とすると，77/316 ≈ 0.243 となり，75.7%の削減がなされたことになる．

5. 実験結果

5.1 実験実施要領

SGNN，K 個の SGNN を用いた ESGNN，最近傍法（nearest neighbor: NN），そして最近傍法の拡張で，与えられた訓練データ（プロトタイプ）の中から最も近いもの上位 k 個のクラスの多数決を出力とする k-最近傍法（k-nearest neighbor: k-NN）の計算コストと解の精度を算出する．各手法に対して，UCI 機械学習リポジトリ[1] の 10 のデータセット☆を shuffling と bagging を用いて実験を行った．ここで，shuffling は訓練データの順序をランダムに入れ換え必ず一度学習に用いる．これに対して，bagging[2] は重複を許す復元抽出により訓練データ数と同数のデータを訓練データセットからサンプリングする．各モデルの分類精度の評価法として，10-fold cross-validation[14] を用いた．各モデルで使用する距離測度として，次に示す修正ユークリッド距離測度を使用した：

$$d(\boldsymbol{x}, \boldsymbol{w}) = \sqrt{\sum_{i=1}^{m} a_i \cdot (x_i - w_i)^2} \ , \qquad (2)$$

ここで，m は属性数すなわち入力次元数を，x_i と w_i は m 次元ベクトル $\boldsymbol{x}$ と $\boldsymbol{w}$ の i 番目の属性を表す．また，a_i は i 番目の属性に対する重み係数であり，次式で表される：

$$a_i = \frac{1}{\max_i - \min_i} \ , \qquad (3)$$

ここで，$\max_i$ と $\min_i$ はそれぞれ i 番目の属性における最大値と最小値である．アンサンブル学習に使用する SGNN の数 K を 25 とし，k-NN で用いる近傍点の数 k は，1，3，5，7，9，11，15，25 の中で平均して最も高い解の精度が得られた 3 とした．なお，すべてのアルゴリズムは C で実装し，数値実験に PC-AT 互換機のパーソナルコンピュータ（CPU: Intel Pentium II 450 MHz，Memory: 322 MB，OS: Linux 2.2.18）を使用した．

5.2 計算コスト

表 2 に SGNN と最近傍法に shuffling と bagging を用いた場合の 10 回の平均計算コストを示す．計算コストとして，平均記憶容量と計算時間を算出した．ここで，平均記憶容量は枝刈り前の SGNN の平均ニューロン数を 1 とした場合の割合として表しており，0 に近いほど高い削減がなされたことを意味する．また，計算時間は 10 回の試行の訓練とテストに必要な総処理時間を表している．表 2 より，節点が増加するために枝刈りを行う前の SGNN は最近傍法よりも大きな記憶容量を必要としているが，枝刈りを行うことで枝刈り前の 30%から 90%の記憶容量を削減し，節点と葉を合計したものでも全数記憶方式の最近傍法の記憶容量を下回ることが分かる（shuffling による liver-disorders を除く）．shuffling と bagging で比較すると，bagging の方が高い削減がなされている．これは，重複するデータが枯葉となり枝刈りがなされるためである．また，計算時間に関しては，木を生成し，木の枝刈りを行い，その木を用いて分類を行う必要があるものの，学習を必要としない最近傍法よりも短時間で処理ができている．これは，最近傍法では学習サンプ

☆ データセットの説明を付録に示す．

表 3 shuffling による SGNN，ESGNN，最近傍法（1-NN），3-最近傍法（3-NN）に対する 10 回の試行における平均分類精度（括弧内は標準偏差を示す）

Table 3 The average classification accuracy of ten trials for single SGNN, ESGNN, nearest neighbor (1-NN), and 3-nearest neighbor (3-NN) with shuffling. The standard deviation is given inside the bracket.

Dataset	SGNN	ESGNN	1-NN	3-NN
balance-scale	0.781(0.053)	0.843(0.059)	0.771(0.057)	0.816(0.049)
breast-cancer-w	0.954(0.020)	0.967(0.023)	0.954(0.025)	0.963(0.024)
glass	0.632(0.102)	0.692(0.075)	0.669(0.086)	0.697(0.078)
ionosphere	0.858(0.042)	0.883(0.043)	0.872(0.034)	0.858(0.044)
iris	0.953(0.055)	0.967(0.047)	0.960(0.047)	0.960(0.047)
letter	0.893(0.007)	0.958(0.003)	0.960(0.004)	0.956(0.005)
liver-disorders	0.574(0.086)	0.635(0.055)	0.626(0.066)	0.623(0.064)
new-thyroid	0.944(0.070)	0.953(0.049)	0.958(0.040)	0.940(0.063)
pima-diabetes	0.687(0.078)	0.711(0.070)	0.692(0.084)	0.737(0.058)
wine	0.938(0.042)	0.960(0.039)	0.949(0.032)	0.960(0.027)
Average	0.821	0.856	0.841	0.851

表 4 bagging による SGNN，ESGNN，最近傍法（1-NN），3-最近傍法（3-NN）に対する 10 回の試行における平均分類精度（括弧内は標準偏差を示す）

Table 4 The average classification accuracy of ten trials for single SGNN, ESGNN, nearest neighbor (1-NN), and 3-nearest neighbor (3-NN) with bagging. The standard deviation is given inside the bracket.

Dataset	SGNN	ESGNN	1-NN	3-NN
balance-scale	0.768(0.055)	0.850(0.047)	0.786(0.054)	0.822(0.044)
breast-cancer-w	0.959(0.030)	0.973(0.017)	0.957(0.021)	0.964(0.024)
glass	0.593(0.107)	0.702(0.060)	0.603(0.110)	0.589(0.093)
ionosphere	0.849(0.042)	0.869(0.041)	0.863(0.038)	0.835(0.036)
iris	0.960(0.056)	0.960(0.047)	0.953(0.063)	0.953(0.063)
letter	0.874(0.009)	0.955(0.003)	0.948(0.003)	0.939(0.005)
liver-disorders	0.574(0.077)	0.614(0.051)	0.617(0.077)	0.583(0.053)
new-thyroid	0.945(0.065)	0.967(0.050)	0.949(0.047)	0.963(0.037)
pima-diabetes	0.671(0.056)	0.719(0.050)	0.715(0.067)	0.706(0.062)
wine	0.938(0.063)	0.960(0.039)	0.933(0.035)	0.944(0.037)
Average	0.813	0.856	0.819	0.830

ルが N 個存在する場合，各テストデータに対して N 回比較を行う必要があるのに対して，SGNN は $\sum_i P_i$ 回でよいためである．ここで，P_i は第 i 層における節点の数である（$P_i \ll N$）．実際，データ数 20,000，各属性の次元数 16 の letter の問題に対して計算時間において差が顕著に現れている．アンサンブル学習を導入することにより，記憶容量，計算時間は識別器の数に比例して増大するものの，25 個の SGNN を用いた ESGNN では，計算時間は最近傍法の半分以下の 392.9 s（shuffling），363.9 s（bagging）であった．ゆえに，本手法は大規模な問題に対して特に有効な手法であるといえる．なお，ESGNN は並列分散処理を行うことで線形的な並列効果（台数効果）を得ることができ，記憶容量を並列処理で使用する各計算機に分散し，計算時間を単一の SGNN で処理するのと同程度まで短縮することができる[7]．

5.3 解の精度

表 3 と**表 4** に shuffling と bagging を用いた SGNN，ESGNN，最近傍法，3-最近傍法の各問題に対する 10 回の試行における平均分類精度を示す．すべての問題に対して ESGNN は解の質を改善し，かつ最近傍法の結果より高いもしくは同等の分類精度が得られている．shuffling の場合，ESGNN は最近傍法，3-NN 法と比べて，10 のデータセット中 7 つで上回っている．残りのデータセットに関しても同程度もしくは多少劣るという結果が得られている．また，bagging の場合は SGNN，最近傍法，3-NN 法では shuffling よりも解の精度は劣る．これは，重複を許した復元抽出による情報の欠損に起因する．一方，ESGNN では，shuffling に比べ大幅に解の精度を改善し，10 のデータセット中 5 つで shuffling による ESGNN の解の精度を上回っている．平均では shuffling と bagging の ESGNN は同程度の解が算出されている．以上の結果より，shuffling と bagging の手法を比較すると，記憶容量を多く削減し，解の精度は同程度の bagging の方が ESGNN には適しているといえる．

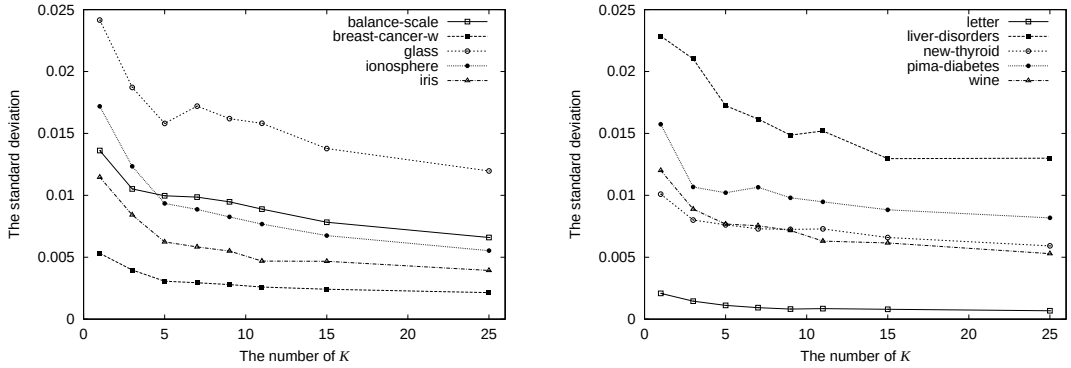

図 9 shuffling を用いた ESGNN の K と標準偏差との関係
Fig. 9 The relation between the number of K and the standard deviation of ESGNN by shuffling.

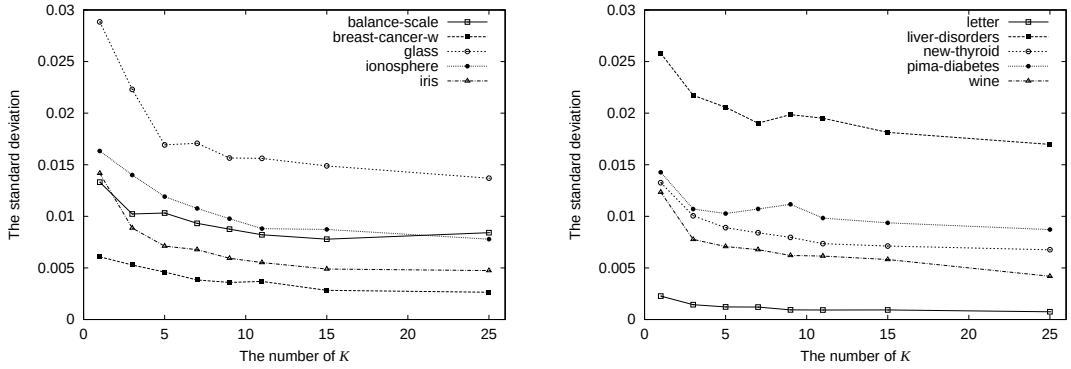

図 10 bagging を用いた ESGNN の K と標準偏差との関係
Fig. 10 The relation between the number of K and the standard deviation of ESGNN by bagging.

ESGNN において，アンサンブル学習に使用する SGNN の数 K に対する解の精度の改善特性を検討する．具体的には，K を 1，3，5，7，9，11，15，25 とし，各問題に対する K の増加にともなう解の精度の標準偏差の減少傾向から改善特性を検討する．なお，標準偏差は，10-fold cross-validation で解の精度を評価する作業を 1 回とし，訓練データの提示順をランダムに入れ換えてこの作業を 100 回行うことで算出する．**図 9** に shuffling を用いた ESGNN の K と標準偏差の関係を，**図 10** に bagging を用いた ESGNN の K と標準偏差の関係を示す．ここで，図 9 (a)，図 10 (a) に表 3 の前半 5 つのデータセット（balance-scale，breast-cancer-w，glass，ionosphere，iris）に対する結果を，図 9 (b)，図 10 (b) に表 3 の後半 5 つのデータセット（letter，liver-disorders，new-thyroid，pima-diabetes，wine）に対する結果を示す．これらの結果より，アンサンブル学習に使用する SGNN の数 K が増加するにつれて，標準偏差が減少することが分かる．また，K は 3，5，7 のような小さな値で大幅に標準偏差が減少し，K が増えるにつれて減少の幅が小さくなっている．したがって，K は大きい場合に高い解の精度が得られるが，K が小さい場合においても解の精度を改善する効果が確実に得られるため，扱うデータセットの規模，計算機の性能に合わせて，K を柔軟に決定することができる．

6. ま と め

本論文では，ESGNN のための新しい枝刈り法を提案し，計算コストと解の精度を算出した．さらに，2 つのサンプリング法において既存の最近傍法，k-最近傍法と比較検討した．実験結果より，記憶容量が大幅に削減され，解の精度は最近傍法よりも高い精度が得られることを示した．そして，shuffling と bagging を用いた ESGNN の比較では，記憶容量の削減率と解

の精度改善特性より，bagging を用いた ESGNN の方が優れていることが分かった．ESGNN は各 SGNN を独立に構築可能であるため，並列分散処理との親和性が高い[7]．したがって，本手法はデータマイニングに関して簡便かつ汎用性の高いモデルであるといえる．今後，ESGNN の効率化のため，記憶容量を削減し，汎化能力を枝刈り前より改善する新たな枝刈り法を検討し，既存の予測器である $C4.5$ [12] の bagging，boosting [11] との汎化能力改善特性，および計算コストの比較検討を行う予定である．

参 考 文 献

1) Blake, C. and Merz, C.: UCI Repository of machine learning databases, University of California, Irvine, Dept. of Information and Computer Sciences (1998). Datasets available at http://www.ics.uci.edu/˜mlearn/MLRepository.html.
2) Breiman, L.: Bagging Predictors, *Machine Learning*, Vol.24, pp.123–140 (1996).
3) Freund, Y. and Schapire, R.：ブースティング入門，人工知能学会誌，Vol.14, No.5, pp.771–780 (1999).
4) Haykin, S.: *Neural Networks: A comprehensive foundation*, 2nd edition, Prentice-Hall, Upper Saddle River, NJ (1999).
5) Inoue, H. and Narihisa, H.: Performance of Self-Generating Neural Network Applied to Pattern Recognition, *5th International Conference on Information Systems Analysis and Synthesis*, Orlando, FL, Vol.5, pp.608–614 (1999).
6) 井上浩孝，成久洋之：アンサンブル自己生成ニューラルネットワークの性能分析，電子情報通信学会論文誌（A），Vol.J83-A, No.10, pp.1227–1230 (2000).
7) 井上浩孝，成久洋之：アンサンブル自己生成ニューラルネットワークによる並列分散データマイニング，人工知能学会全国大会講演論文集，島根，pp.3D1–04 (2001).
8) Kohonen, T.: *Self-Organizing Maps*, Springer-Verlag, Berlin (1995).
9) 松嶋敏泰：情報論的学習理論での数理モデル，人工知能学会誌，Vol.16, No.2, pp.252–255 (2001).
10) Perrone, M.P. and Cooper, L.N.: When networks disagree: Ensemble methods for hybrid neural networks, *Neural Networks for Speech and Image Processing*, Mammone, R.J. (Ed.), pp.126–142, Chapman-Hall (1993).
11) Quinlan, J.R.: Bagging, Boosting, and C4.5, *Proc. 13th National Conference on Artificial Intelligence*, Portland, OR, pp.725–730 (1996).
12) Quinlan, J.R.: *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA (1993).
13) Rumelhart, D.E., Hinton, G.E. and Williams, R.J.: Learning Internal Representations by Error Propagation, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Rumelhart, D.E., McClelland, J.L. and the PDP Research Group (Eds.), chapter 8, pp.318–362, The MIT Press, Cambridge, MA (1986).
14) Stone, M.: Cross-validation: A review, *Math. Operationsforsch. Statist., Ser. Statistics*, Vol.9, No.1, pp.127–139 (1978).
15) Wen, W.X., Jennings, A. and Liu, H.: Learning a Neural Tree, *International Joint Conference on Neural Networks*, Beijing, China, Vol.2, pp.751–756 (1992).
16) Wen, W.X., Pang, V. and Jennings, A.: Self-Generating vs. Self-Organizing, What's Different?, *Neural Networks Theory, Technology, and Applications*, Simpson, P.K. (Ed.), pp.210–214, IEEE Technology Update Series, 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 (1999).

付 録

データセットの説明

Name	Cases	Classes	Attributes
balance-scale	625	3	4
breast-cancer-w	699	2	9
glass	214	6	9
ionosphere	351	2	34
iris	150	3	4
letter	20,000	26	16
liver-disorders	345	2	7
new-thyroid	215	3	5
pima-diabetes	768	2	8
wine	178	3	13

Name：データセット名，Cases：データ数，Classes：クラス数，Attributes：属性数（入力次元数）

(平成 13 年 8 月 16 日受付)
(平成 13 年 10 月 13 日再受付)
(平成 13 年 11 月 28 日採録)

井上 浩孝（正会員）

平成 9 年岡山理科大学工学部情報工学科卒業．平成 11 年同大学大学院修士課程修了．平成 14 年同大学院博士課程修了．博士（工学）．同年呉工業高等専門学校電気情報工学科助手．ニューラルネットワークの効率的な学習法に関する研究に従事．電子情報通信学会，人工知能学会，IEEE-CS 各会員．

成久 洋之（正会員）

昭和 45 年京都大学大学院博士課程修了．工学博士．現在，岡山理科大学工学部情報工学科教授．昭和 58 年～平成 6 年岡山理科大学情報処理センター所長．OR およびシステムの最適化に関する研究に従事．電子情報通信学会，日本 OR 学会各会員．