G-018

自己生成ニューラルネットワークを用いたアンサンブル学習法
Ensemble Learning Using Self-Generating Neural Networks

松林　真司† Shinji Matsubayashi
井上　浩孝‡ Hirotaka Inoue

1. まえがき

自己生成ニューラルネットワーク (Self-Generating Neural Network : 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ことができる識別器である．しかし，SGNN の値には多くの汎化誤差が含まれる．そこで，我々は，SGNN を複数作成することで，SGNN の問題点である汎化誤差を改善した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 Network : ESGNN) を提案し，ESGNN の有効性を示した [1]．本研究では，パターン認識において従来の手法である最近傍法 (Nearest Neighbor method: NN 法) [2],[3] を含め，ESGNN，SGNN，最近傍法の識別精度の比較検討をする．また，ESGNN と SGNN の処理時間を求め，枝刈りによる効果を比較検討をする．

2. SGNN のオンライン枝刈り法

SGNN は，自己生成ニューラル木 (Self-Generating Neural Tree : SGNT) として実装される．そこで，SGNN の生成アルゴリズムを定義する上で，いくつかの表記を定義する．

- 入力データ：$e_i \in \mathbb{R}$.
- SGNT の根, 葉, ノード：n_j.
- n_j の重み：$w_j \in R^m$.
- n_j の葉の数：c_j.
- 計測距離：c_j.
- 勝ち葉：n_{win}.

表 1 に SGNN を生成するために用いた関数と，その詳細を示す．図 1 に疑似 C 言語による SGNN の生成アルゴリズムを示す．

生成過程において，競合学習により訓練実例ベクトル e_i に対する SGNT 内の勝者となる葉 n_{win} を決定する．根と，根から n_{win} に至る節点に存在する節点 n_j の重み $w_{jk}(k = 1, 2, \ldots, m)$ は次式を用いて更新する．

$$w_{jk} \leftarrow w_{jk} + \frac{1}{c_j} \cdot (e_{ik} - w_{jk}), \quad 1 \leq k \leq m. \ (1)$$

表 1 関数とその詳細

複手順	詳細
copy(n_j,e_i/w_{win})	n_j を造り n_j の重み e_i か w_{win} をコピー
choose(e_i,n_1)	e_i に対する n_{win} を選ぶ
leaf(n_{win})	n_{win} が SGNT 内の葉かどうかを判断する
connect(n_j, n_{win})	n_j を n_{win} の子供として繋げる
prune(n_{win})	n_{win} に同じクラスが存在すれば葉を刈る

3. ESGNN のオフライン枝刈り法

アンサンブル学習は，有限個の訓練データを用いて複数の予測器を生成し，未学習テストデータに対する汎化能力を向上する手法である，ESGNN は，識別器である SGNN を複数個並べ，各 SGNN が出力した値を多数決の要領で最も多い値を選び，選ばれた値を ESGNN の解として 1 つの値を出力するものである．ESGNN の汎化能力は従来の手法より優れているか同等である．しかし，計算コスト (記憶容量, 探索時間) は SGNN の数が増加するつれて増大するという欠点がある．そこで，計算コスト (記憶容量，探索時間) を削減するため，識別問題に対する ESGNN の枝刈り法を提案する．本手法では，ESGNN の生成にマージフェーズと評価フェーズの 2 つのアルゴリズムを用いた．マージフェーズではクラス情報と部分木の葉を刈り取るかどうかを決定するための閾値 α を用いる．同じ親を持つ葉および節点においては，最多クラスの占める割合が閾値 α 以上であるならば，それらの葉および節点を刈り取ることにより，親節点にその最多クラスを与える．マージフェーズのアルゴリズムを図 2 に示す．評価フェーズは，最適な閾値 α は各識別問題により異なるため，評価フェーズを用いて 10 群交差検定 (10-fold Cross-validation : 10-CV)[4] によりマージフェーズで用いる閾値 α の最適な値を決定するものである．評価フェーズのアルゴリズムを図 3 に示す．

```
Input :
  A set of training examples E = {e_i},
 i = 1, ... , N.
  A distance measure d(e_i,w_j).

Program Code :
  copy(n_1,e_1);
  for(i = 2, j=2; i<=N; i++) {
    n_win = choose(e_i,n_1);
    if (leaf(n_win)){
      copy(n_j, w_win);
      connect(n_j,n_win);
      j++;
    }
    copy(n_j, e_i);
    connect(n_j,e_win);
    j++;
    prune(n_win);
  }

Output :
  Constructed SGNT by E.
```

図 1 SGNN の生成アルゴリズム

4. 実験方法

本研究で用いた訓練データは，UCI に保管されている 10 問の識別問題 [5] を用いた．識別器は，K=25 に設定した ESGN と K=1 に設定した SGNN，そして，最近傍法の 3 つの識別器を用いた．3 つの識別器に，それぞれバギング [6] を用いて学習させ，各訓練セットのサンプリング順序を変化させることにより，それぞれ 100 回実験を行い，その時の識別精度の平均を測定した．また，ESGNN と SGNN の

†呉工業高等専門学校専攻科機械電気工学専攻
‡呉工業高等専門学校電気情報工学科

表 2 100 回の試行における識別精度の平均と処理時間

Dataset	classification accuracy ESGNN	SGNN	NN	computation time [s] ESGNN	SGNN
breast-cancer	**0.97**	0.96	0.96	**1.79**	0.09
balance-scale	**0.87**	0.78	0.78	**1.61**	0.07
glass	**0.72**	0.64	0.67	**0.81**	0.04
ionosphere	**0.89**	0.85	0.86	**2.70**	0.16
iris	**0.97**	0.95	0.95	**0.18**	0.01
liver-disorders	**0.62**	0.58	0.61	**1.43**	0.07
thyroid-disease	**0.95**	0.94	**0.95**	**0.50**	**0.02**
pima-diabetes	**0.75**	0.70	0.70	**3.24**	0.13
wine	**0.96**	0.95	0.95	**0.45**	0.02
letter	**0.96**	0.88	**0.96**	**444.40**	18.04

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

図 2 マージフェーズのアルゴリズム

```
1 begin   initialize α = 0.5
2   do for each α
3     evaluate the marge phase with 10-fold CV
4     if the best classification accuracy is obtained,
5     then record the α as the optimal
            threshold value
6     α ← α + 0.05
7   until α = 1
8 end.
```

図 3 評価フェーズのアルゴリズム

処理速度を測定した．また，ESGNN と SGNN の処理時間は，学習時間，最適化時間，テスト時間を合計したものである．

また，各識別問題に対して最適な閾値 α は異なるので，各問題ごとに α を 0.5 から 1 まで 0.05 ずつ変化 $[\alpha = 0.5, 0.55, 0.6, \cdots, 1]$ させ，評価フェーズにより最適な α を見つけ出し，マージフェーズを用いて枝刈りを行うことにより，メモリ使用量と処理時間の削減を行った．本研究では，全ての計算は IBM PC-AT 互換機 (CPU: Intel Pentium 4 2.4GHz，Memory:256MB) 上で実装する．また，全てのアルゴリズムは C 言語で実装したものであり，gcc に最適化オプション-O2 を付けてコンパイルした．

5. 実験結果

表 4 に 10 問の識別問題に対する K=25 に設定した ESGNN と K=1 に設定した SGNN，そして，最近傍法との各 100 回の試行における識別精度の平均値と ESGNN，SGNN の処理時間を示す．

表 2 より ESGNN の識別精度の平均値は，SGNN と比べると全ての識別問題において，向上していることが分かる．特に，10 問の識別問題の内の balance-scale，glass，letter の識別精度の平均値の結果より，ESGNN の識別精度は，SGNN の識別精度よりそれぞれ 9%，8%，8%の識別精度の大幅な向上が見られた．また，最近傍法と比較すると，十問中八つの問題で ESGNN の識別精度が勝っている．残り二つの問題においては識別精度は同等であった．また，処理時間については，K=25 とした ESGNN と K=1 とした SGNN とでは，SGNN の方が処理時間は少ないが，ESGNN の K は 25 と設定しているため，SGNN の処理時間を 25 倍とした時，thyroid-disease の処理時間は同等であるが，他の識別問題では ESGNN の方が処理時間が早い．これは，ESGNN の処理速度が SGNN の処理速度より早く，ESGNN のオフライン枝刈り方による効果といえる．以上の結果より，本手法は，データマイニングにおいて有効な手段であるといえる．また，ESGNN のオフライン枝刈り法による処理速度の向上も確認した．

6. むすび

本研究では，自己生成ニューラルネットワーク (Self-Generating Neural Network : SGNN) を用いた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 Network : ESGNN) を生成し，ESGNN と SGNN，さらに従来の手法である最近傍法 (NN 法) に実際に識別問題を与えることにより，ESGNN，SGNN，最近傍法の識別精度，また，ESGNN，SGNN の処理時間の比較検討を行った．

実験結果より，ESGNN は SGNN や最近傍法と比べ全ての識別問題において同等，もしくは，それ以上のより高い識別精度が得られることが分かった．また，ESGNN のオフライン枝刈り法により処理速度を速めることができ，処理時間の削減も確認できた．ゆえに，ESGNN は，SGNN や最近傍法と比べ，より優れた識別器であり，SGNN に比べ処理速度も優れている．以上のことから，本手法において，自己生成ニューラルネットワークを用いたアンサンブル学習は有効であるといえる．今後，我々は本手法の特性を考慮し，ESGNN のさらなる計算コストの削減，識別精度の改良を行うと共に，k-近傍法 (k-Nearest Neighbor method: k−NN 法) との識別精度，処理時間の比較検討を行う予定である．

参考文献

[1] 佛圓和之，井上浩孝，“自己生成ニューラルネットワークを用いたアンサンブル学習に関する研究，”平成 19 年度電気・情報関連学会中国支部連合大会講演論文集，pp.152-153，2007

[2] Richard O.Duda,Peter E.Hart, David G.Stork,"Pattern Classification, " 2001.

[3] 安居院猛，長尾智晴，" C 言語による画像処理入門，" 2000.

[4] M. Stone, “Cross-validation: A review, " Math.Operationsforsch.Statist.,Ser.Statistics, vol.9,no. 1, pp. 127-139, 1978.

[5] C.Blake, C.Merz, “UCI repository of machine learning databases, " 1998. http://www.ics.usi.edu/ mlearn/MLRepository.html

[6] L. Breiman, “Bagging predictors," Machine Learning, vol. 24, pp. 123-140, 1996