H-009

アンサンブル自己生成ニューラルネットワークの効率的なオンライン枝刈り法

An Efficient Online Pruning Method for Ensemble Self-Generating Neural Networks

井上 浩孝 † 成久 洋之 ‡
Hirotaka Inoue Hiroyuki Narihisa

1. まえがき

我々は訓練データの提示順をランダムに入れ換えることで複数の自己生成ニューラルネットを作成し，それらの出力を統合して未学習テストデータに対する汎化誤差を改善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）のための効率的枝刈り法を提案し，汎化能力を枝刈り前より改善し，かつ記憶容量を大幅に削減可能であることを示した [1]．本研究では，より効率的な処理を行うための ESGNN のオンライン枝刈り法を提案し，従来の手法である k-近傍法（k-nearest neighbor: k-NN）と比較検討する．

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．SGNT アルゴリズムを記述する前に，表記法を以下に示す．

- 入力データベクトル: $\boldsymbol{e}_i \in \mathbb{R}^m$.
- SGNT の根, 葉, 節点: n_j.
- n_j の重みベクトル: $\boldsymbol{w}_j \in \mathbb{R}^m$.
- n_j に含まれる葉の数: c_j.
- 距離測度: $d(\boldsymbol{e}_i, \boldsymbol{w}_j)$.
- $\boldsymbol{e}_i$ に対して勝者となる SGNT の葉: n_{win}.

SGNT アルゴリズムの疑似 C 言語によるコードを図 1 に，表 1 に SGNT アルゴリズムの副手続きと仕様をそれぞれ示す．

生成過程において，競合学習により訓練実例ベクトル $\boldsymbol{e}_i$ 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．

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

全訓練データが葉として SGNT に挿入された後，各節点の重みはその節点内に含まれる全ての葉の重みの平均値となる．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．本研究では効果的に SGNT の枝刈りを実行するため，記憶容量と汎化能力のドレードオフの関係がある距離に基づいて葉の数を制御するために文献 [1] の SGNT アルゴリズムで用いているしきい値を削除している．上記の問題を回避しより効率的な処理を行うため，我々は新たな枝刈り法を副手続き `prune(n_win)` に実装する．冗長な枝を刈りとるためにクラスラベルを使用する．n_{win} に結合した葉に対して，もしそれら全ての葉が同一クラスを持つ場合，その親ノードに葉のクラスを与え，全ての葉を刈り取る．次節で ESGNN のオフライン枝刈り法について述べる．

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．本研究では，K 個の識別器からなる ESGNN の出力の多数決を取る．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が，計算コストは SGNT の数が増加するにつれて増大する．そこで，記憶容量を削減するため，識別問題に対する ESGNN の枝刈り法を提案する．本手法はマージフェーズと評価フェーズの二つの部分からなる．マージフェーズは冗長な葉を削減するための枝刈りアルゴリズムとして実装される (図 2)．マージフェーズ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しきい値 α を用いる．同じ親を持つ葉および節点において，最多クラスの占める割合がしきい値 α 以上であるならば，それらの葉および節点を刈り取り，親節点にその最多クラスを与える．

Input :
- A set of training examples $E = \{\boldsymbol{e}_i\}, i = 1, \ldots, N$.
- A distance measure $d(\boldsymbol{e}_i, \boldsymbol{w}_j)$.

Method :

```
1  copy(n1,e1);
2  for  (i = 2, j = 2; i <= N; i++) {
3    n_win = choose(e_i, n1);
4    if (leaf(n_win))  {
5      copy(n_j, w_win);
6      connect(n_j, n_win);
7      j++;
8    }
9    copy(n_j, e_i);
10   connect(n_j, n_win);
11   j++;
12   prune(n_win);
13 }
```

Output : Constructed SGNT by E.

図 1 SGNT アルゴリズム.

表 1 SGNT アルゴリズムで用いられる副手続き.

Sub procedure	Specification
`copy`(n_j,$\boldsymbol{e}_i$)	Create n_j and copy $\boldsymbol{e}_i$ as $\boldsymbol{w}_j$ in n_j.
`choose`($\boldsymbol{e}_i$,n_1)	Decide n_{win} for $\boldsymbol{e}_i$.
`leaf`(n_{win})	Check n_{win} whether it is a leaf.
`connect`(n_j, n_{win})	Connect n_j as a child leaf of n_{win}.
`prune`(n_{win})	Prune leaves if leaves have a same class.

```
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 枝刈りアルゴリズム.

†呉工業高等専門学校電気情報工学科
‡岡山理科大学工学部情報工学科

表 2 100 回の試行における枝刈り前，枝刈り後のアンサンブルモデルの平均記憶容量と識別精度．識別精度欄の括弧内に標準偏差を示す（$\times 10^{-3}$）．

	memory requirement			classification accyracy		
Dataset	pruned	unpruned	ratio	pruned	unpruned	ratio
balance-scale	107.68	861.18	12.5	0.861(6.33)	0.837(7.83)	+2.4
breast-cancer-w	30.88	897.37	3.4	0.97(2.41)	0.966(2.71)	+0.4
glass	104.33	297.75	35	0.714(13.01)	0.709(14.86)	+0.5
ionosphere	50.75	472.39	10.7	0.891(6.75)	0.862(7.33)	+2.9
iris	15.64	208.56	7.4	0.962(6.04)	0.955(5.45)	+0.7
letter	6197.5	27028.56	22.9	0.956(0.77)	0.955(0.72)	+0.1
liver-disorders	163.12	471.6	34.5	0.648(12.89)	0.636(13.36)	+1.2
new-thyroid	49.45	298.21	16.5	0.958(7.5)	0.957(7.49)	+0.1
pima-diabetes	204.4	1045.03	19.5	0.749(7.05)	0.728(7.83)	+2.1
wine	15	238.95	6.2	0.976(4.41)	0.972(5.57)	+0.4
Average	684.69	3181.96	16.9	0.868	0.857	+1.1

表 3 最良の識別精度が得られた枝刈りアンサンブルモデルと k-NN の識別精度，記憶容量，処理時間の比較．

	classification acc.		memory requirement		computation time (s)	
Dataset	SONG	k-NN	SONG	k-NN	SONG	k-NN
balance-scale	0.878	**0.888**	**109.93**	562.5	**0.82**	1.14
breast-cancer-w	**0.974**	0.969	**26.8**	629.1	**1.18**	1.25
glass	**0.758**	0.701	**91.33**	192.6	0.36	**0.08**
ionosphere	**0.912**	0.866	**51.38**	315.9	1.93	**0.2**
iris	**0.973**	0.96	**11.34**	135	0.13	**0.05**
letter	0.958	**0.96**	**6208.03**	18000	**208.52**	503.14
liver-disorders	**0.685**	0.653	**134.17**	310.5	**0.54**	0.56
new-thyroid	**0.972**	**0.972**	**45.74**	193.5	0.23	**0.05**
pima-diabetes	**0.764**	0.751	**183.57**	691.2	**1.72**	2.49
wine	**0.983**	0.977	**11.8**	160.2	0.31	**0.15**
Average	**0.885**	0.869	**687.41**	2119.1	**21.57**	50.91

各識別問題に対する最適なしきい値 α はそれぞれ異なっているので，評価フェーズでは 10 群交差検定を導入することにより最良のしきい値を決定する．

4. 実験結果

機械学習リポジトリ [2] の 10 のベンチマーク問題に対して，バギング [3] を用いて生成した ESGNN の計算コスト（記憶容量と計算時間）と汎化能力を検討する．10 のベンチマーク問題に対して 10 群交差検定を用いて汎化能力を評価する．本実験に関して，SGNN の生成のための距離測度に修正ユークリッド距離を用いる．最良なしきい値 α を選択するため，0.5 から 1 まで 0.05 毎に異なるしきい値 α を設定する．ESGNN における SGNT の数 K を 25 とし，各訓練セットのサンプリング順序を変化させることで 100 回試行する．全ての計算は SunBlade2000 (CPU: UltraSPARC III Cu 900MHz, Memory: 1GB, OS: Solaris 8) 上で実行する．なお，全てのアルゴリズムは C 言語で実装し，gcc（ver. 3.3.2）に最適化オプション `-O2` を付けてコンパイルした．

100 回の試行における枝刈り前，枝刈り後の ESGNN の平均記憶容量と識別精度を表 2 に示す．ここで記憶容量として，SGNT の根，節点，葉の合計を掲載している．記憶容量は枝刈り法を適用することで 10 のデータセットに対して 6596.6% 削減されているのがわかる．特に glass に対して関連研究 [1] では 57.6%の削減率だったものが，65%と 7.4%の削減が達成された．また，10 のデータセットに対して枝刈り前の ESGNN よりもより高い精度が得られていることがわかる．以上の結果より，本手法は計算コスト削減と識別精度改善の両方の観点から有効であるといえる．

本手法の有効性を検証するため，同一環境における k-NN との識別精度，記憶容量，処理時間の比較を行う．ここでは，枝刈り ESGNN の 100 回の試行における最良の識別精度が得られた結果と，k-NN で k を 1,3,5,7,9,11,13,15,25 と変化したときに最良の識別精度が得られた結果とを比較する（表 3）．なお，k-NN において，距離測度は本手法で用いた修正ユークリッド距離を使用した．枝刈り後の ESGNN は k-NN に比べて識別精度で 10 問中八つの問題でより高い結果が得られており，記憶容量は全ての問題でより小さな記憶容量を達成しており，処理時間は特に大規模な letter の問題に対して約 1/2.4 となっている．以上の結果より，本手法は実践的な大規模データセットを扱うデータマイニングに対して有効な手法であるといえる．

5. むすび

本研究では，ESGNN のための新しいオンライン枝刈り法を提案し，計算コストと汎化能力を評価した．実験結果より，枝刈り SGNT をアンサンブルモデルの基本識別器に用いることで記憶容量が大幅に減少し，かつ識別精度が向上することがわかった．以上の結果より，本手法は大規模データセットを識別するための簡便で実践的な手法であるといえる．今後，本手法の識別器の数と汎化能力改善特性および，並列特性を検討する予定である．

謝辞

本研究の一部は文部科学省科学研究費補助金若手研究 (B)(No.15700206) によるものである．

参考文献

[1] 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ための高速枝刈り法,” 情報処理学会論文誌：数理モデル化と応用, vol. 43, no. SIG(TOM6), pp. 59–69, 2003.

[2] C. Blake and C. Merz, “UCI repository of machine learning databases,” University of California, Irvine, Dept of Information and Computer Science, 1998. Datasets is available at `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[3] L. Breiman, “Bagging predictors,” Machine Learning, vol. 24, pp. 123–140, 1996.