

法政大学学術機関リポジトリ

HOSEI UNIVERSITY REPOSITORY

PDF issue: 2024-12-22

簡素な進化的アルゴリズムと ヒステリシス ニューラルネットの合成

山岡, 慧 / Yamaoka, Kei

(出版者 / Publisher)

法政大学大学院理工学・工学研究科

(雑誌名 / Journal or Publication Title)

法政大学大学院紀要. 理工学・工学研究科編 / 法政大学大学院紀要. 理工学・工学研究科編

(巻 / Volume)

58

(開始ページ / Start Page)

1

(終了ページ / End Page)

3

(発行年 / Year)

2017-03-31

(URL)

<https://doi.org/10.15002/00014198>

簡素な進化的アルゴリズムと ヒステリシスニューラルネットの合成

A SIMPLE EVOLUTIONARY ALGORITHM AND SYNTHESIS OF HYSTERESIS NEURAL NETWORKS

山岡 慧

Kei YAMAOKA

指導教員 斎藤利通

法政大学大学院理工学研究科電気電子工学専攻修士課程

This paper presents the growing greedy search algorithm and its application to associative memories of hysteresis neural networks in which storage of desired memories are guaranteed. In the algorithm, individuals correspond to cross-connection parameters, the cost function evaluates the number of spurious memories, and the set of individuals can grow depending on the global best. Performing basic numerical experiments, the algorithm efficiency is investigated.

Key Words : *greedy search, hysteresis neural nets, associative memories*

ヒステリシスニューラルネット (Hysteresis Neural Networks 以下 HNN [1]) は 2 値ヒステリシス活性化関数によって特徴付けられる連続時間再帰型ネットワークである。HNN はパラメータに依存し、共存平衡点、同期、カオス、分岐 [2] など様々な現象を呈す。また、ダイナミクスは区分線形微分方程式で記述されるため、厳密な解析が可能となる。HNN は連想メモリ、AD コンバータ、組み合わせ最適化問題などへの適用されている [1]-[5]。連想メモリにおいて、所望記憶の埋め込みと安定性については理論的に可能である。しかし、偽記憶を抑制することは困難であり、問題の一つである。

本論文では貪欲法 [6] に基づく簡素な進化的アルゴリズム (Simple Evolutionary Algorithm 以下 SEA) を提案する。SEA において、進化個体は HNN の相互結合パラメータに対応する。また、偽記憶の数を評価関数として評価する。初期個体は所望の記憶を保障する相関学習 [1] によって与えられる。個体 (相互結合パラメータ) に bit 変換とエリート戦略が適用され、次世代個体候補が作成され、評価値によって評価される。SEA は評価によって個体の数が変わる可能性があるアルゴリズムである。この個体数が変化する構造は従来の貪欲法との大きな相違点である。本研究では、連想メモリの基本的な例題について数値実験を行い、SEA が偽記憶の抑制に有効であることを確認した。本章では SEA を HNN に適用したが、SEA は他の様々なシステムへ適用可能である。

1. ヒステリシスニューラルネット

ヒステリシスニューラルネット (HNN) のダイナミクスは以下のように記述される。

$$\begin{aligned} \dot{x}_i &= -x_i + \sum_{l=1}^N w_{il}y_l + d_i \equiv -(x_i - p_i), \quad i = 1 \sim N \\ y_i &= h(x_i) = \begin{cases} +1 & \text{for } x_i > -Th \\ -1 & \text{for } x_i < Th \end{cases} \end{aligned} \quad (1)$$

ここで、 $x \equiv (x_1, \dots, x_N)$ は内部状態のベクトルであり、 $y \equiv (y_1, \dots, y_N)$ 、 $y_i \in \{-1, 1\}$ は 2 値出力ベクトルである。 $h(x)$

をヒステリシス活性化関数であり、図. 1 に示す。 $h(x)$ は x が右側のしきい値 $Th > 0$ (もしくは、左側のしきい値 $-Th$) に達したとき -1 から 1 (もしくは、 1 to -1) に切り替わる。 w_{ij} は結合パラメータであり、 d_i はオフセットパラメータである。簡単の為、相互結合パラメータは以下のような 3 値であり対角線に対称なものをを用いる。相互結合パラメータ: $w_{ij} = w_{ji} \in \{-1, 0, 1\}$ 。また、 $d_i = 0$ and $Th = 0$ とし、システムの簡単化を図る。ここで HNN の安定平衡点は以下の式によって与えられる。

$$p(y) \equiv (p_1, \dots, p_N), \quad p_i = \sum_{l=1}^N w_{il}y_l \quad (2)$$

平衡点 $p(y)$ お安定性は以下の条件式によって分類される。

$$\begin{aligned} \text{任意の } i \text{ に対して } p_i y_i > -Th \text{ ならば } p(y) \text{ は安定である。} \\ \text{ある } i \text{ に対して } p_i y_i \leq -Th \text{ ならば } p(y) \text{ は不安定である。} \end{aligned} \quad (3)$$

HNN が平衡点を持つ場合、以下の条件を満たすことでおおよそ全ての解はいずれかの平衡点に収束する。

$$w_{ii} + Th > 0 \text{ for } i = 1 \sim N \quad (4)$$

以上の条件式 (2)-(4) は [1] によって証明されている。以下では、HNN の連想メモリへの応用を考える。まず、所望の記憶を以下のように定義する。

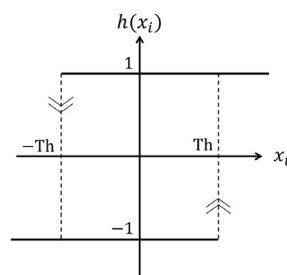


図 1 2 値ヒステリシス活性化関数

$$s^1, \dots, s^M, s^l \equiv (s^l_1, \dots, s^l_N), l = 1 \sim M$$

所望の記憶に対応する平衡点が安定になるパラメータを決定することができれば、所望の記憶はHNNに埋め込まれているといえる。すべての所望の記憶は以下のように定義された相関学習で保証されている [1]。まず、相互結合パラメータは相関行列を3値化することによって与えられる。

$$w_{ij} = \begin{cases} -1 & \text{for } c_{ij} \leq 0 \\ +1 & \text{for } c_{ij} > 0 \end{cases}, c_{ij} = \sum_{l=1}^M s^l_i s^l_j, i \neq j \quad (5)$$

自己相関結合パラメータ w_{ii} は以下の条件式を満たす最小の整数で与えられる。

$$0 < w_{ii} + Th + Q_i, Q_i = \min_k \sum_{i \neq j} w_{ij} s^k_i s^k_j \quad (6)$$

ある出力ベクトルの平衡点が安定であるが所望のメモリでない場合、出力ベクトルは偽記憶と呼ばれる。所望の記憶が全て記憶されていても、通常は偽記憶が存在している。偽記憶の抑制は、連想メモリにおける重要な問題である。

2. 簡素な進化的アルゴリズム

本稿では新しい進化的アルゴリズムとして貪欲法に基づく、簡素な進化的アルゴリズム (SEA) を提案する。まず、基本的な定義を行う。進化のステップ数を t とする。また、 $P^i(t) \equiv (p^i_1(t), \dots, p^i_M(t))$ をステップ $t=0000$ における i 番目の個体と定義する。ここで M は個体における次元であり、 $i = 1 \sim k(t)$ である。また、個体数 $k(t)$ は可変である。本研究で個体は以下のようにHNNにおける結合パラメータの半分を用いる。

$$P^i(t) \equiv \{w_{ij}\}, i > j, i, j = 1 \sim N$$

ここで、次元数 M は $M = N(N-1)/2$ となる。他の結合パラメータは以下のように行列の対角線に対称な値を用いる、 $w_{ji} = w_{ij}$ 。また、SEAの評価関数は以下のように設定する。

$$F(P^i(t)) = \text{偽記憶数}$$

$P^i(t)$ を w_{ij} に代入し、出力ベクトルを式 (2)-(4) に代入することで、評価関数は w_{ii} と式 (6) によって得られる。本研究で用いるSEAは、以下の2つのサブプログラムで構成されている。

(1) SEA1: ビット変換

Step 1: 初期化。 $t = 0$ とし、 $k(t) = 1$ とする。初期個体を $P^i(t)$, $i = 1 \sim k(t)$ は、式 (??) に示した相関学習によって与えられる。評価関数も初期化され、初期個体を用い、 $F(P^i(t))$ となる。また、 G_b をグローバルベストとし、 $G_b(t) = F(P^i(t))$ で与える。

Step 2: 1ビット変換。 $P^i(t)$, $i = 1 \sim k(t)$ である場合、個体の要素の1つを反転する。反転することで、 $N \times k(t)$ における個体 $Q^j(t)$, $j = 1 \sim N \times k(t)$ を得る。

Step 3: 個体の評価。個体群 $\{Q^j(t)\}$ は評価関数 F によって評価される。ここで $F(Q_b(t))$ を $F(Q^j(t))$ for $j = 1 \sim N \times k(t)$ に対する最良の値とすることでグローバルベストが更新される。

$$G_b(t) \leftarrow \begin{cases} F(Q^j(t)) & \text{if } F(Q_b(t)) < G_b(t) \\ G_b(t) & \text{otherwise} \end{cases} \quad (7)$$

複数の個体 $Q^j(t)$ が評価関数 F における同じ最良値を有する場合、その個体は次世代の個体となり、個体数が変化する。最良の個体の数が個体数制限 M_a を超えた場合、最良の個体のうち M_a 個がランダムに選択される。例えば、 $Q^1(t)$, $Q^7(t)$, $Q^9(t)$ が同じ最良値を持つならば、これらの3つの個体は $P^1(t)$, $P^2(t)$, $P^3(t)$ であり、 $k(t) \leftarrow 3$ にする。

Step 4: G_b 更新の確認と終了判定。 $G_b(t)$ が更新制限 M_b 回更新されていない場合、SEA1を終了し、SEA2に切り替える。

Step 5: $t \leftarrow t + 1$ とし Step 2 に進み、最大時間制限 t_{max1} まで繰り返す。

(2) SEA2: ゼロ挿入

Step 1: 初期化。 $t \leftarrow t_1 + 1$ とする、ここで t_1 はSEA1における最終ステップ数とする。 $k(t) = 0$ とする。 $P^i(t)$, $i = 1 \sim k(t)$ はSEA1の最良の評価値をもつ個体とする。複数の最良の個体が存在する場合、それらのうちの1つがランダムに選択される。コスト関数と G_b は $P^i(t)$ に初期化する。

Step 2: ゼロ挿入。 $P^i(t)$, $i = 1 \sim k(t)$ に対して、個体の要素に1つゼロを挿入する。挿入することで、 $N \times k(t)$ における個体 $R^j(t)$, $j = 1 \sim N \times k(t)$ を得る。

Step 3: 個体の評価。個体群 $\{R^j(t)\}$ は評価関数 F によって評価される。ここで $F(R_b(t))$ を $F(R^j(t))$ for $j = 1 \sim N \times k(t)$ に対する最良の値とすることでグローバルベストが更新される。

$$G_b(t) \leftarrow \begin{cases} F(R^j(t)) & \text{if } F(R_b(t)) < G_b(t) \\ G_b(t) & \text{otherwise} \end{cases} \quad (8)$$

複数の個体 $R^j(t)$ が評価関数 F における同じ最良値を有する場合、その個体は次世代の個体となり、個体数が変化する。最良の個体の数が個体数制限 M_a を超えた場合、最良の個体のうち M_a 個がランダムに選択される。

Step 4: G_b 更新の確認と終了判定。 $G_b(t)$ が更新制限 M_b 回更新されていない場合、探索を終了する。

Step 5: $t \leftarrow t + 1$ とし Step 2 に進み、最大時間制限 t_{max2} まで繰り返す。

3. 数値実験

本項ではアルゴリズムの効率を解析するため、基本的な数値実験を行った。14個の所望の記憶を基に以下の例を考察する。Example 1 (s^8 から s^{14} はそれぞれ s^1 から s^7 の反転パターンである。)

$$\begin{aligned} s^1 &\equiv (-1, 1, -1, -1, 1, 1, -1, 1) = -s^8 \\ s^2 &\equiv (1, -1, 1, -1, 1, -1, 1, -1) = -s^9 \\ s^3 &\equiv (1, 1, -1, -1, 1, 1, -1, -1) = -s^{10} \\ s^4 &\equiv (1, 1, 1, -1, -1, -1, 1, -1) = -s^{11} \\ s^5 &\equiv (-1, 1, 1, -1, 1, 1, -1, -1) = -s^{12} \\ s^6 &\equiv (1, -1, 1, -1, 1, 1, -1, -1) = -s^{13} \\ s^7 &\equiv (-1, 1, 1, 1, -1, -1, -1, 1) = -s^{14} \end{aligned}$$

相関学習を適用すると、これらの所望の記憶は8次元HNN ($N = 8$) に保存することができる。表 1 に相関学習後の結合パラメータを示す。例 1 は88個の偽記憶を有す、その偽記憶を抑制するため、SEA1とSEA2を適用した。アルゴリズムのパラメータは以下のように設定した。

$$t_{max1} = t_{max2} = 25, M_a = 20, M_b = 5$$

表. 2 に SEA1 の後のパラメータ値を示す. 例 1 は 12 個の偽記憶を有す. 表. 3 に、SEA2 の後のパラメータ値を示す. 例 1 は 4 個の偽記憶を有す.

図. 2 に Gb、および #IDU(個体数)の推移を示す. SEA1 が偽記憶を十分に抑制していることが図から確認できる. #IDU は大きく変化し、その変化は探索に効果的であると考えられる. $t = 13$ では、Gb の改善が停滞し、SEA1 は SEA2 に切り替わる. $t = 21$ では、Gb の改善が停滞し、SEA2 が終了している.

4. まとめ

この論文では、簡素な進化アルゴリズム (SEA) を紹介した. SEA1 および SEA2 はそれぞれ、ビット反転およびゼロ挿入に基づいて動作する. SEA の個体は評価関数によって評価され、個体の集合は評価関数による評価に応じて変化することができる. このアルゴリズムをヒステリシスニューラルネットワークにおける偽記憶の抑制問題に適用した. 基本的な実験によって、アルゴリズムの効率を確認した.

今後の課題は、探索プロセスの分析、アルゴリズムパラメータの効果的な設定、様々な離散問題への適用などがあげられる.

参考文献

- 1) Jin'no, K., Saito, T.: Analysis and synthesis of a continuous-time hysteresis neural network. In: Proc. IEEE/ISCAS, 471-474 (1992)
- 2) Jin'no, K., Nakamura, T., Saito, T.: Analysis of bifurcation phenomena in a 3 cells hysteresis neural network. IEEE Trans. Circuit Syst. I, 46, 7, 851-857 (1999)

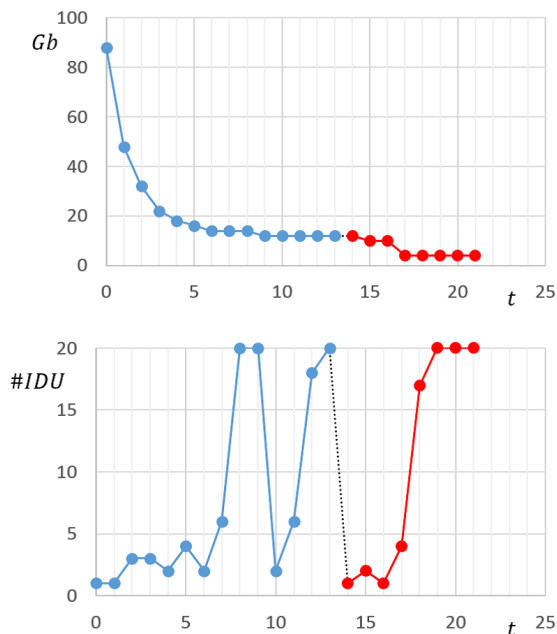


図 2 Gb と #IDU の進化過程 (1)

表 1 相関学習後の結合パラメータ

j	1	2	3	4	5	6	7	8
w_{1j}	+3	-1	-1	-1	+1	+1	+1	-1
w_{2j}	-1	+1	-1	-1	-1	-1	-1	+1
w_{3j}	+1	-1	+3	-1	-1	-1	+1	-1
w_{4j}	-1	-1	-1	+1	-1	-1	-1	+1
w_{5j}	+1	-1	-1	-1	+3	+1	-1	-1
w_{6j}	+1	-1	-1	-1	+1	+1	-1	+1
w_{7j}	+1	-1	+1	-1	-1	-1	+3	-1
w_{8j}	-1	+1	-1	+1	-1	+1	-1	+1

表 2 SEA1 の結合パラメータ

j	1	2	3	4	5	6	7	8
w_{1j}	+4	+1	+1	-1	-1	-1	-1	-1
w_{2j}	+1	+6	+1	-1	-1	-1	+1	-1
w_{3j}	+1	+1	+2	+1	-1	-1	-1	-1
w_{4j}	-1	-1	+1	0	-1	-1	-1	+1
w_{5j}	-1	-1	-1	-1	0	+1	-1	-1
w_{6j}	-1	-1	-1	-1	+1	0	-1	+1
w_{7j}	-1	-1	-1	-1	+1	-1	+4	-1
w_{8j}	-1	-1	-1	+1	-1	+1	-1	0

表 3 SEA2 の結合パラメータ

j	1	2	3	4	5	6	7	8
w_{1j}	+3	0	+1	0	-1	-1	0	-1
w_{2j}	0	+4	0	-1	-1	-1	+1	-1
w_{3j}	+1	0	+2	0	-1	-1	+1	-1
w_{4j}	0	-1	0	0	-1	-1	-1	+1
w_{5j}	-1	-1	-1	-1	+1	+1	0	-1
w_{6j}	-1	-1	-1	-1	+1	0	-1	+1
w_{7j}	0	+1	+1	-1	0	-1	+2	-1
w_{8j}	-1	-1	-1	+1	-1	+1	+1	0

- 3) Jin'no, K., Tanaka, M.: Hysteresis quantizer, In: Proc. IEEE/ISCAS, 661-664 (1997)
- 4) Nakaguchi, T., Jin'no, K., Tanaka, M.: Hysteresis Neural Networks for N-Queens Problems. IEICE Trans. Fundamentals., E82-A, 9, 1851-1859 (1999).
- 5) Nakaguchi, T., Isome, S., Jin'no, K., Tanaka, M.: Box Puzzling Problem Solver by Hysteresis Neural Networks. IEICE Trans. Fundamentals., E84-A, 9, 2173-2181 (2001)
- 6) Couvreur, C., Bresler, Y.: On the optimality of the backward greedy algorithm for the subset selection problem. SIAM J. Matrix Anal. Appl., 21, 3, 797-808 (2000)