遺伝的アルゴリズムを用いたモジュール型 動的ニューラルネットの構造決定法[†]

小澤 誠^{一*}・堤 一 義^{**}・馬場 則 夫^{***}

An Architecture Design Method of Modular Dynamical Neural Networks Using Genetic Algorithms[†]

Seiichi OZAWA*, Kazuyoshi TSUTSUMI** and Norio BABA***

In this paper, we propose an evolutionary approach to architecture design of modular dynamical neural networks. As one of modular dynamical neural networks, we adopt Cross-Coupled Hopfield Nets (CCHN) in which plural Hopfield networks are coupled to each other. The architecture of CCHN is represented by some structural-parameters such as the number of modules, the numbers of units per module, the module connectivity, and so forth. In the proposed design method, these structural-parameters are treated as phenotype of an individual, and suitable modular architecture is searched through the evolution of its genetic representation (genotype) by using genetic algorithms. Based on a simple direct coding method, the order of length of genetic representation for the structural-parameters can be estimated to be $O(N^2)$ where N is the total number of units. On the other hand, the order of genetic representation proposed here is $O(N)$. To verify the usefulness of proposed method, we apply a CCHN to associative memories. Here, the fitness of an individual is defined so as to be larger when a CCHN has a simpler architecture as well as when the association performance is higher. As the result of simulations, we certify that the proposed design method can find high-performance CCHN with simple modular architectures.

Key Words: modular neural network, architecture design, genetic algorithms, associative memories

1. はじめに

ニューラルネットを用いて問題を解く際、階層型および相互結合型のどちらのタイプであっても初期段階では隠れユニットを多めに用意し、ユニット間を完全結合した状態で学習を始めることが多い。これは問題を解くのに適したニューラルネットの構造やパラメータ数があらかじめわかっていないためであるが、このような結合荷重のホモニアス性や冗長性は学習時間の増大や局所解へのトラップを引き起こす可能性がある。これを回避する方法としては、(1) 不要な結合や隠れユニットを積極的に削除するメカニズムを取り入れる方法¹⁾、(2) 入出力ユニット間の結合関係に何らかの構造を

あらかじめ導入する方法、が挙げられる。ここでは後者のアプローチをモジュール型ニューラルネットと呼び、その構造を効率的に決定する方法について議論する。

これまで提案されてきたモジュール型ニューラルネットは、入力空間における特定の部分領域を受けもつ階層型ニューラルネットを複数個用意し、これらの出力を統合して解を求める、いわゆる分割統治型のものが多い^{2), 3)}。このタイプは振る舞いが静的であり、主にパターン認識や予測問題に多く適用されてきた。しかし、ニューラルネットに行わせたい情報処理は静的なものだけに限らず、問題によっては動的な振る舞いを利用することが有利な場合もある⁴⁾。このような動的な特性をもつモジュール型ニューラルネット(以後、モジュール型動的ニューラルネットと呼ぶ)として、著者らはクロス結合ホップフィールドネット (Cross-Coupled Hopfield Nets, CCHN) を提案してきた^{5)~7)}。CCHNは複数のホップフィールドネットが相互結合されたニューラルネットであり、構成されるモジュール数やモジュール間結合のトポロジーに依存せずネットワーク全体として漸近安定性を保証できるモデルである。CCHNモデルで扱えるのは漸近安定な系だけであるが、モジュール数やモジュール間結合などのモジュール構造を変更することで多様な動特性が得られることを示している⁸⁾。著者らは、このようなCCHNを主に連想記憶に適用してきたが、どのようなモジュール構造

[†] 第6回インテリジェント・システム・シンポジウムで一部発表(1996.10)

* 神戸大学大学院自然科学研究科 神戸市灘区六甲台町 1-1

** 龍谷大学理工学部 大津市瀬田大江町横谷 1-5

*** 大阪教育大学教育学部 柏原市旭ヶ丘 4-698-1

* Graduate School of Science and Technology, Kobe University, 1-1 Rokkodai, Nada, Kobe

** Faculty of Science and Technology, Ryukoku University, 1-5 Yokotani, Ohe-cho, Seta, Otsu

*** Faculty of Education, Osaka Kyoiku University, 4-698-1 Asahigaoka, Kashiwara, Osaka

(Received June 4, 1999)

(Revised December 17, 1999)

を与えれば所望する連想特性が得られるかは試行錯誤的に決めるしかなかった。

一般に、モジュール型ニューラルネットの構造を決定するパラメータには、モジュール数や各モジュールのユニット数、モジュール間の結合形態や相互作用の種類など多数考えられる(以後、これらを構造パラメータと呼ぶ)。よって、これらの組み合わせで表現できるモジュール構造のバリエーションは膨大となり、試行錯誤的に決定するには限界がある。ニューラルネットの構造を自動的に決定する方法はこれまでいくつか提案されており、それらの多くは進化型探索、つまり遺伝的アルゴリズムや遺伝的プログラミングなどを用いる方法である。前者のアプローチは「直接コーディング法」と「文法コーディング法」に大きく分けられるが、文法コーディング法は取り扱いが複雑であるため階層型ニューラルネット以外ほとんど用いられていない。これに対し、直接コーディング法は構造パラメータと遺伝子型との対応が直接的で適用が容易であるため、階層型ニューラルネットだけでなくモジュール型ニューラルネットにも適用されている。このうちモジュール型動的ニューラルネットへの適用としては、Happel らのアプローチが挙げられる⁹⁾。このアプローチでは、遺伝子型の前半部にモジュール数と各モジュールのユニット数の情報が直接的にコーディングされ、その長さはすべてのモジュールに含まれる総ユニット数の最大値に等しくなる。また、後半部にはモジュール間結合の情報がコーディングされるが、直接コーディング法に基づいているためモジュールの最大個数の2乗に等しい長さの遺伝子型が必要であり、モジュール数が多いと遺伝子列がかなり長くなる。そこで、Happel らは想定するモジュールの最大個数を制限してモジュール間結合部の遺伝子長があまり大きくないようにしている。遺伝的プログラミングを用いた後者のアプローチのうちモジュール型動的ニューラルネットへ適用した事例には、モジュール構造に対する遺伝子型を木で表した Cho らの研究がある¹⁰⁾。このアプローチでは、木の各ノードをモジュールに対応づけ、ノード間結合がそのままモジュール間結合に対応づけられる。この方法は、木の階層性を利用して複雑なモジュール構造を表現しやすいという利点をもつが、モジュール間の結合数は枝数に等しく、枝数が最大でモジュール数の2乗オーダーとなる点で Happel らのアプローチと同様の問題をもつ。

そこで本論文では、モジュール型動的ニューラルネットの構造パラメータを効率よくコーディングできる遺伝子型表現として、直接コーディング法をベースとしながらもモジュール間結合の決定に文法コーディング法を一部取り入れた表現方法を提案する。また、このような遺伝子型をもつ個体集団に対して遺伝的アルゴリズムを適用し、問題に適したモジュール構造の探索が容易になることを示す。第2章で CCHN モデルを簡単に説明したあと、第3章で提案する遺伝子型表現と表現型への変換手順について述べる。また、遺伝的アルゴリズムを用いたモジュール構造の探索手順を示す。第4章では、CCHN を連想記憶に応用し、提案手法によって優れた連

Fig. 1 Schematic diagram of (a) 4-module CCHN and (b) a two-layer internetwork from the n th module to the m th module, $INT^{(n,m)}$

想性能が得られるかを検証する。第5章では本論文で得られた結果をまとめ、今後の課題について述べる。

2. クロス結合ホップフィールドネット

クロス結合ホップフィールドネット(以後、CCHNと略す)は、複数個のホップフィールドネットがインターネットワークと呼ばれる階層型ニューラルネットで相互結合されたモデルである。Fig. 1(a)は、4つのモジュール(ホップフィールドネット)が互いに結合し合っている様子を概念的に表したものである。モジュール間にある矢印は2モジュール間に何らかの望ましい状態関係があることを示しており、その一つ一つは Fig. 1(b)で示した2層インターネットワークの写像として実現される。なお、モジュール内結合は完全相互結合であるが、モジュール間結合はすべて結合される必要はない。第 n モジュールから第 m モジュールの状態間写像を与えるインターネットワークを $INT^{(n,m)}$ と表すと、これらの入出力関係はつぎのように与えられる(注1)。

[インターネットワーク $INT^{(n,m)}$ の入出力関係]

$$o_i^{(n,m)(2)} = \sum_j w_{ij}^{(n,m)} o_j^{(n,m)(1)} \quad (1)$$

$$o_i^{(n,m)(1)} = \begin{cases} v_i^{(n)} & (i \leq N^{(n)}) \\ v_{i-N^{(n)}}^{(m)} & (N^{(n)} < i \leq N^{(n)} + N^{(m)}) \end{cases} \quad (2)$$

ここで、 $o_i^{(n,m)(l)}$ ($l = 1, 2$) は $INT^{(n,m)}$ の第 l 層出力、 $w_{ij}^{(n,m)}$ は結合荷重、 $N^{(m)}$ は第 m モジュールのユニット数を表す。Fig. 1(b) および (2) 式からわかるように、 $o_i^{(n,m)(1)}$ は $INT^{(n,m)}$ の入力であり、第 m モジュールと第 n モジュールの状態 $v_i^{(m)}$, $v_i^{(n)}$ がそのまま入力される。これら2モジュールの状態が同時に参照されるのは、モジュールの状態間に多対多の対応関係があっても適切な相互作用が与えられるようにするためである⁷⁾。

(注1) CCHN モデルには、インターネットワークに隠れ層を導入したり、(1)~(2)式以外の入出力関係をもつものもある。しかし、ここでは簡単のため、3節で取り扱うものだけに説明を限定しておく。他のモデルについては文献5)~7)を参照されたい。

CCHNの情報処理はエネルギー関数という一種の目的関数を用いて記述される。\$M\$ 個のモジュールで構成されたCCHNのエネルギー関数は、第 \$m\$ モジュールにおける \$j\$ 番目のユニットから \$i\$ 番目のユニットへの結合荷重を \$T_{ij}^{(m)}\$ としたとき以下で与えられる。

$$E = \sum_{m=1}^M \alpha^{(m)} \left[-\frac{1}{2} \sum_i \sum_j T_{ij}^{(m)} v_i^{(m)} v_j^{(m)} + \sum_i \frac{1}{r_i^{(m)}} \int_0^{v_i^{(m)}} g^{-1}(v) dv - \sum_i I_i^{(m)} v_i^{(m)} \right] + \frac{1}{2} \sum_{m=1}^M \sum_{\substack{n=1 \\ n \neq m}}^M \beta^{(n,m)} \sum_i (v_i^{(m)} - o_i^{(n,m)(2)})^2 \quad (3)$$

ここで、\$r_i^{(m)}\$ および \$I_i^{(m)}\$ はそれぞれ第 \$m\$ モジュール \$i\$ 番目のユニットにおけるエネルギー損失を決定する係数と外部入力を表す。また、\$g(\cdot)\$ はユニットの出力関数である。\$\alpha^{(m)}\$ および \$\beta^{(n,m)}\$ は各エネルギー項の「貢献度」を決定する正の定数である。

(3) 式右辺のエネルギー関数は大きく二つの部分からなり、第1項はモジュールの情報処理を記述するエネルギーに対応し、第2項はモジュール間相互作用のエネルギーに対応する。(3) 式右辺第1項はアナログ型ホップフィールドネットのエネルギー関数に等しく、よく知られるように、その最小点は \$T_{ij}^{(m)}\$ や \$I_i^{(m)}\$ によって変化する。また第2項は第 \$m\$ モジュールの状態 \$v_i^{(m)}\$ と \$INT^{(n,m)}\$ の出力 \$o_i^{(n,m)(2)}\$ の二乗誤差で与えられ、モジュール間で定義された望ましい状態関係がすべて満たされたとき0となり最小化される。

ネットワークの状態更新則、モジュールおよびインターネットワークの結合荷重に対する学習則は、(3) 式のエネルギーを単調減少させる動作方程式の導出を行った結果として求められる^(注2)。結果的には、モジュール結合の学習則として相関学習が、インターネットワークの学習則として誤差逆伝播学習が微分方程式のかたちで得られる⁵⁾。これらは連立微分方程式となるが、状態更新のスピードが学習のスピードに比べて十分速いという一種の断熱近似を導入し、実際には学習をオフラインで行う。つまり、各モジュールで安定化させたい状態^(注3)を教師データとしてモジュールおよびインターネットワークを学習し、その後、状態更新則に基づいてCCHNを動作させる。ここで、CCHNモデルでは望ましい状態関係を与えたいモジュール間にのみ相互作用があるものとして(3) 式右辺第2項の誤差エネルギーを定義し、インターネットワークの学習を行うことに注意する。

3. 遺伝的アルゴリズムを用いた構造決定法

3.1 CCHNの表現型

CCHNを個体とする集団に対して遺伝的アルゴリズムによる探索を行い、モジュール構造を自動的に決定する方法を提案する。この場合、表現型はCCHNの構造パラメータとなり、それらを二進列でコーディングしたものが遺伝子型に対応する。まずどのような構造的特徴を表現型とするかを決める必要がある。

CCHNモデルの構造パラメータは(a)モジュール数 \$M\$、(b)各モジュールのユニット数 \$N^{(m)}\$、(c)モジュール間の結合形態、(d)相互作用の種類、(e)インターネットワークの隠れ層数、(f)モジュール内情報処理の貢献度 \$\alpha^{(m)}\$、(g)モジュール間相互作用の貢献度 \$\beta^{(n,m)}\$、などが挙げられる。これら構造パラメータの組み合わせによって表されるCCHNは無数にあり、すべてを探索の対象とすることはあまり現実的でない。そこで本論文では、これらの中から(a)~(c)を表現型として採用し、これら以外の構造パラメータはあらかじめ固定しておく。具体的には、(d)は(1)-(2)式で示されるタイプの相互作用のみを扱うこととし、(e)の隠れ層数は零とする(つまりインターネットワークの入出力間写像は線形に限定)。また(f)および(g)の貢献度は \$\alpha^{(m)} = 1\$、\$\beta^{(n,m)} = 0.8\$ とする。

3.2 遺伝子コーディングに対する考え方

ニューラルネットの構造を遺伝的アルゴリズムで探索するアプローチは、その遺伝子型へのコーディング法により「直接コーディング法」と「文法コーディング法」に基づく方法に分けられる⁹⁾。前者はニューラルネットのユニットや結合に関する情報(ユニット数や結合の有無など)を表現型とし、それらの情報が直接的に遺伝子列にコーディングされる方法である。この方法は遺伝子型から表現型への変換が容易である反面、ユニット数や結合数が多いと遺伝子列が長くなり、探索空間が大きくなりやすい。また、ネットワークサイズがあらかじめ決まっているものにしか適用できない。一方、後者はネットワークを生成するルールを表現型とし、このルールを構成する基本要素が遺伝子列にコーディングされる。ネットワークは、遺伝子型から表現型への変換によって生成されたルールに基づいて生成される。基本要素の組み合わせによって多様なルールが生成されるので、様々なネットワーク構造の実現が可能である。しかし、無意味なルールを生じないための工夫が必要であり、またルールを構成する基本要素の選び方にも注意を払う必要がある。文法コーディング法は、構造が単純な階層型ニューラルネットであっても取り扱いが容易でないため⁹⁾、本研究では直接コーディング法を基本に考えていくことにする。

まず、モジュール型動的ニューラルネットに最も単純な直接コーディング法を適用したときの遺伝子長を見積もってみる。ネットワーク全体の総ユニット数が \$N\$ であるとする、モジュール数および各モジュールのユニット数は \$1 \sim N\$ の値

(注2) 具体的には、時間微分 \$dE/dt \leq 0\$ が常に成り立つよう動作方程式を求める。詳細は文献7)を参照されたい。

(注3) 連想記憶に適用する場合、モジュールで想起したい記憶パターンに対応する。

をとることになる。このとき、各モジュールのユニット数は $\log_2 N$ ビットの二進値で表せるので、モジュール数とユニット数をコーディングするのに $N \log_2 N$ ビットあればよい。一方、すべてのモジュール間結合に対してその有無を二進値でコーディングすると、モジュール数の最大が N であることより遺伝子長は N^2 となる。以上より、最も単純な直接コーディング法を適用した場合、遺伝子型の長さは $N \log_2 N + N^2$ となる。これは、探索空間の大きさが $O(2^{N^2})$ となることを意味しており、総ユニット数が多いと探索が困難になると予想される。また、このコーディング法はモジュール数が少ないと利用される情報が少なく、かなり無駄の多い表現になっている。

上記の直接コーディング法の問題は、一つ一つのモジュール間結合の有無を異なる遺伝子で表現していることと、モジュール間結合数が可変パラメータであるモジュール数に依存することにある。そこで、効率のよいコーディングを行う方法として、構造パラメータの可変性に依存しない何らかのルールを遺伝子型から表現型への変換手順に導入することが考えられる。しかし、すべてのルールを遺伝子列から生成すると、今度は前述した文法コーディング法の欠点の問題となってくる。よって、いくつかのルールは遺伝子列に記述されない固定的なものとして扱うのがよいと思われる。このような考えを実現する遺伝子型から表現型への変換手順を以下で提案する。

3.3 遺伝子型から表現型への変換手順

本節では、CCHNのユニットに属性(以下、ユニット属性と呼ぶ)を導入し、これに基づいた遺伝子型から表現型への変換方法を提案する。ユニット属性は表現型への変換過程で参照される情報であり、本提案手法では3.1で述べた3種類の構造パラメータを決定するために用いられる。CCHNの遺伝子型は、このユニット属性とモジュール間結合の情報がコーディングされた2つの異なる二進列で構成される。前者は各ユニットの属性値が直接コーディングされたもので、以下ではユニット属性部 X_u と呼ぶ。また、後者はモジュール間結合を決定するルールに対応し、モジュール間結合部 X_c と呼ぶことにする。

まずユニット属性部 X_u の遺伝子表現について述べる。いまネットワーク全体の総ユニット数を N とし、各ユニットに p 種類のユニット属性 $\{A(1), \dots, A(p)\}$ のいずれかが割り当てられるものとする。第 i 番目ユニットのユニット属性 U_i は長さ q の二進列 $b_i = \{b_{i1}, \dots, b_{iq}\}$ ($i = 1, \dots, N$) でコーディングされ(つまりユニット属性部の長さは qN となる)、これらの対応関係は以下の R_1 によってあらかじめ定義されているものとする。

$$U_i = A(R_1(b_i)) \quad (4)$$

b_i の長さは q であるため最大 2^q 個のユニット属性を表せるが、異なる二進列が同一の属性に重複して対応づけられても構わないものとする。ユニット属性部の遺伝子表現 X_u は上

Table 1 An example of correspondence between unit attributes and binary strings

binary strings, b_i	unit attributes, U_i
000, 011, 100	$A(1)$
001, 101	$A(2)$
010, 110	$A(3)$
111	$A(4)$

記の R_1 に基づいて、以下のように N 個のユニット属性の並び Y_u に変換される。

$$X_u : b_1, \dots, b_i, \dots, b_N$$

→

$$Y_u : U_1, \dots, U_i, \dots, U_N \\ = A(R_1(b_1)), \dots, A(R_1(b_i)), \dots, A(R_1(b_N))$$

[例題1] $p = 4$ としてユニット属性 $\{A(1), A(2), A(3), A(4)\}$ を仮定する。二進列の長さ q を3として、合計8種類の二進列 b_i がたとえば **Table 1** に示すようにユニット属性 U_i と R_1 によって対応づけられるとする。総ユニット数 N が10であるCCHNに対し、ユニット属性部 X_u が

$$011001100111111010111110101110$$

で表されたとき、上記の R_1 に基づいて、つぎのようにユニット属性の並び Y_u に変換される。

$$A(1)A(2)A(1)A(4)A(4)A(3)A(4)A(3)A(2)A(3)$$

つぎに、 i 番目と $i+1$ 番目のユニット属性に対して変数 v_i ($i = 1, \dots, N-1$) を以下のように定義する。

$$v_i = R_2(U_i, U_{i+1}) \quad (5)$$

R_2 は i 番目と $i+1$ 番目のユニット属性の値によってモジュールに分割するかどうかを決定するルールに相当し、分割する場合は1を与え、分割しない場合は0を与えるものとする。よって $v_i = 1$ となる場所を調べれば遺伝子列 X_u からモジュール数および各モジュールのユニット数が一意に決定される。ここで分割されたモジュール数を M 、各モジュールのユニット数を $N^{(m)}$ ($m = 1, \dots, M$)、各モジュールにおけるユニット属性の列を $y_u^{(m)}$ で表すことにする。

[例題2] 以下の R_2 を考える。

$$R_2(U_i, U_{i+1}) = \begin{cases} 1 & ; U_i = U_{i+1} = A(4) \\ 0 & ; \text{otherwise} \end{cases}$$

このとき前述の例で得られたユニット属性の列から、モジュール分割を決定する変数 v_i が以下のように求められる。

$$v_i = 000100000$$

これより4番目と5番目のユニットで2つのモジュールに分割されることになり、これらモジュールにおけるユニット属性の列はつぎのように表される。

$$y_u^{(1)} : A(1)A(2)A(1)A(4)$$

$$y_u^{(2)} : A(4)A(3)A(4)A(3)A(2)A(3)$$

		to			
		A(1)	A(2)	...	A(p)
from	A(1)	x_{c1}	x_{c2}	...	x_{cp}
	A(2)	$x_{c(p+1)}$	...		
	$\vdots$	$\vdots$			$\vdots$
	A(p)			...	x_{cp^2}

Fig. 2 Rule of connectivity generated from binary string X_c

この例では, $M = 2$, $N^{(1)} = 4$, $N^{(2)} = 6$ となることから. ■

モジュール間の結合形態は, 前述したモジュール間結合部の遺伝子列 X_c から抽出されたルールに基づいて決定される. モジュール間結合部は, ユニット属性の個数を p としたとき p^2 の長さを持ち, $X_c = \{x_{c1}, \dots, x_{ck}, \dots, x_{cp^2}\}$ で表すものとする. この遺伝子列 X_c は, Fig. 2 に示すように $p \times p$ の行列状に並べ替えられる. ここで行および列はモジュール属性 $\{A(1), \dots, A(p)\}$ を表し, j 行 k 列の値が 1(0) のとき, モジュール属性が $A(j)$ であるモジュールから $A(k)$ であるモジュールへの結合がある(ない)ことを表す. なお, モジュール属性はユニット属性の列 $y_u^{(m)}$ から以下で表される R_3 によって決められるものとする.

$$C^{(m)} = A(R_3(y_u^{(m)})) \quad (6)$$

ここで, $C^{(m)}$ は第 m モジュールの属性を表す.

[例題 3] $p = 4$ としてモジュール属性 $\{A(1), A(2), A(3), A(4)\}$ を仮定する. ここで R_3 として, 各モジュールのユニット属性の列 $y_u^{(m)}$ のうち最も多いユニット属性をモジュール属性とするルールを考えてみる. このとき, 前の例で得られた 2 つのモジュールの属性は

$$C^{(1)} = A(1) \quad C^{(2)} = A(3)$$

となる. いまモジュール間結合部 X_c としてつぎのものを考える.

$$1010111100101100$$

この場合, モジュール 1 から 3 への結合はあるが, モジュール 3 から 1 への結合はないことになる. ■

以上からわかるように, 提案する遺伝子型は X_u および X_c を合わせた長さ $qN + p^2$ の遺伝子列となり, その遺伝子長はユニット数 N に関して $O(N)$ となる. また, 表現型への変換ルールのうちモジュール間結合の有無を決定するルール (つまり X_c) のみ GA の探索対象となり, その他のルール $R_1 \sim R_3$ はあらかじめ決められたもので固定されることに注意する. これによって構造的に意味のない表現型が生成されることを防ぎつつ, $O(N)$ の遺伝子長でも自由度の高いモジュール構造の探索が可能になっている.

3.4 GA を用いたモジュール構造の探索

前節で提案した遺伝子型の個体集団を考え, これに GA を

適用して優れたモジュール構造を探索する. ここではエリートを用いた適応度比例戦略に基づく選択を行い, GA オペレータには交叉, 突然変異, 逆位を採用した. 提案する構造決定法の手順を以下に簡単にまとめておく.

- (1) 長さ $qN + p^2$ である遺伝子列 $X (= X_u + X_c)$ を S 個ランダムに選んで初期集団 $Z_0(0)$ とする.
- (2) 3.3 に基づき, 各個体の遺伝子型を表現型 (構造パラメータ) に変換する.
- (3) (2) で決定されたモジュール構造をもつ CCHN の性能を評価し, 適応度を計算する.
- (4) 最も適応度の高いものをエリートとして選択し, 残りの $S - 1$ 個は集団 $Z_0(t)$ から適応度比例戦略に基づいて選択する. この $S - 1$ 個の集団を $Z_1(t)$ とする.
- (5) 集団 $Z_1(t)$ に対し, 交叉, 突然変異, 逆位を適用し, 新たな集団 $Z_2(t)$ を求める.
- (6) 世代数 t があらかじめ設定した最大世代数 G より大きいとき探索を終了する. そうでなければ, $Z_2(t)$ にエリートを加えた S 個の集団を新たに $Z_0(t+1)$ とし, さらに $t = t + 1$ として (2) に戻る.

4. シミュレーション実験

4.1 連想記憶への適用

ここでは CCHN を連想記憶に適用することで, 提案した構造決定法の評価を行う. 記憶パターンには要素が ± 1 の 2 値で表される 100 次元ベクトルを用い, これらが状態空間内でクラスター状に分布した記憶パターン集合を CCHN に記憶する. いま μ 番目の記憶パターンを ξ_μ , 遺伝子型から変換されたモジュール数とユニット数をそれぞれ M と $N^{(m)}$ ($m = 1, \dots, M$) とする. このとき, 100 次元の記憶パターンは各モジュールのユニット数に等しい次元をもつ M 個の部分ベクトル $\xi_\mu^{(m)} \in \{-1, 1\}^{N^{(m)}}$ ($m = 1, \dots, M$) に分割される. ここで, 記憶パターンと部分ベクトルの間には

$$\xi_\mu = \{\xi_\mu^{(1)}, \dots, \xi_\mu^{(M)}\}'$$

が成り立つものとする. なお, $'$ はベクトルの転置を表し, ξ_μ と $\xi_\mu^{(m)}$ はそれぞれ縦ベクトルとしている. 各モジュールでは安定平衡状態がこれら部分ベクトルと一致するように相関学習を行い, モジュール間のインターネットワークでは部分ベクトル間の対応関係を逆誤差伝播法で学習する. 以下では, 1 つのクラスターを構成する記憶パターン数を 4 とし, クラスター数を 3 または 9 とする (つまり, 記憶パターン数 P は 12 または 36). また, 同一クラスター内における記憶パターン間の相関は約 0.42 とし, 各クラスターのセントロイドはほぼ直交するように選んだ.

連想性能は, 記憶パターンの引込み領域がどの程度大きいかで評価する. つまり, 正しい想起が可能である初期状態の集合が大きいかほど連想性能が高いとする. 限られた計算時間なるべく正確な評価を行うため, 引込み領域の境界付近で 1 個体当たり 400 点の初期値をランダムに選んだ. いま, 初

期状態 $v(0)$ とこれにもっとも近い記憶パターン ξ_μ との方向余弦 d をつぎのように定義する.

$$d = \max_{\mu} \left\{ \frac{1}{N} v(0)' \xi_{\mu} \right\} \quad (7)$$

このとき、初期値の選択範囲は、クラスター数が3のときは $d = 0.15 \sim 0.35$ 、クラスター数が9のときは $d = 0.3 \sim 0.5$ と設定した。なお、初期状態の集合はすべての個体に対して同一のものを与えた。

4.2 適応度関数

CCHN で実現できるモジュール構造のバリエーションは非常に多く、構造は異なっても同様の性能を実現できるものも少なからず存在すると考えられる。そこで、ここでは個体の適応度として、単に連想性能だけでなく構造上の制約も同時に考慮できるものを定義する。以下では、なるべく単純なモジュール構造をもつニューラルネットが見出されるよう制約を与えることにする。モジュール構造の単純さを表す指標にはいくつか考えられるが、本論文では以下で定義されるモジュール間結合密度 σ_i を取り入れる。

$$\sigma_i = \frac{c_i}{M_i(M_i - 1)} \quad (8)$$

ここで、 i は個体の番号を表すインデックスであり、 M_i および c_i は第 i 個体の遺伝子型から表現型に変換して求められたモジュール数およびモジュール間結合数を表す。この定義から明らかなように、 σ_i はモジュール間結合数に比例して大きくなり、完全相互結合された状態で1となる。

i 番目の個体に対する適応度 f_i を以下で定義する。

$$f_i = \max \left(\frac{1}{K} \sum_{k=1}^K (1 - d_k) \theta_{ik} - \gamma \sigma_i, 0 \right) \quad (9)$$

ただし

$$\theta_{ik} = \begin{cases} 1 & ; \text{ 試行 } k \text{ の想起が成功} \\ 0 & ; \text{ それ以外} \end{cases}$$

ここで、 d_k は試行 k における初期状態の方向余弦を表す。なお、初期状態 $v(0)$ から想起を開始し、ネットワークの状態ベクトルと記憶パターンのすべての要素において、それらの符号が平衡状態で完全に一致したとき「想起が成功した」とする。(9) 式の右辺第1項は連想性能に関する項であり、想起の成功回数が多いほど適応度は高く、また記憶点から遠い初期状態から想起できたとき高い適応度が与えられる。第2項は結合密度の定義から明らかなように、複雑なモジュール構造をとることに対するペナルティーである。なお、 γ はこれら2項のバランスを決定する正の定数である。

4.3 実験結果および考察

各パラメータを以下のように設定し、3.3-3.4で述べた構造決定法を実行した。

個体数 $S = 50$, 最大世代数 $G = 50$

交差率 0.8, 突然変異率 0.001, 逆位確率 0.3

属性数 $p = 4$, 属性を表す二進列の長さ $q = 3$

なお、遺伝子型から表現型への変換に用いるルール $R_1 \sim R_3$ は例で取り上げたものを用いる。また比較のために、3.2で

Table 2 Percentage of success recall when $\gamma = 0$

	$P = 12$	$P = 36$
proposed coding method	73.5	65.1
direct coding method	73.7	53.5
2-module (50,50)	66.9	55.9
3-module (33,33,34)	60.4	39.9
4-module (25,25,25,25)	64.0	15.3
5-module (20,20,20,20,20)	32.8	1.7

取り上げた直接コーディング法に基づいた方法も検討した(注4)。この方法では、遺伝子型はモジュール数とユニット数をコーディングする前半部とモジュール間結合をコーディングする後半部に分けられ、それぞれを二進列で表現する。ユニット総数 N が100であるため、各モジュールのユニット数には7ビットの二進値が必要となり、それをモジュールの最大個数だけ直列につないで前半部の遺伝子列とした。なお、遺伝子列の先頭からモジュールのユニット数を順に読み出し、総ユニット数が100になった時点で前半部の読み出しをやめる。遺伝子列の後半部ではモジュール間結合が直接的にコーディングされるため、モジュール数の最大値の2乗、すなわち10000ビットの二進列が必要となる。以上より、以下のシミュレーションで用いる直接コーディング法の遺伝子長は10700となる。なお、直接コーディング法で用いられる遺伝的アルゴリズムおよび各種パラメータは提案手法と同じものを扱い、どちらの手法においても初期個体集団のすべての遺伝子型はモジュール数が1となるようにランダムに選んだ。

(9) 式において $\gamma = 0$ としたときの結果を Table 2 に示す。この結果は、構造決定で評価用に用いられた400点の初期状態に加えて新たに3600点の初期状態に対して想起を行ったときの成功率を表している。ただし、3種類の異なる初期状態集合に対して構造決定法をそれぞれ実行し、同様の想起実験を通して得られた結果を平均した。Table 2の1行目および2行目は、提案したコーディング手法と直接コーディング法を用いたときに見出された個体のうちで最も適応度の高いものの成功率を表している。また比較のために、3行目以降にモジュール数を2~5とし、各モジュールのユニット数をカッコ内の数値に固定したCCHNの成功率を示しておく(モジュール間結合は完全結合)。ユニット数が100であるとき、ランダムパターンに対するホップフィールドネットの記憶容量はほぼ $P=15$ とされているが¹¹⁾、ここで用いたパターンは相関性をもつので Table 2における $P=12$ のケースでも正しい想起は行えないことに注意する。

まず、提案手法と Table 2の3行目以降の結果との比較であるが、この結果より各モジュールにユニット数を均等に割り当てて方法では高い連想性能は望めないことがわかる。つまり、与えられた記憶パターンに適したモジュール構造は単

(注4) 文法コーディング法は、ネットワークを生成するルールの選択や致死遺伝子を防ぐ工夫などに関して、直接コーディング法に比べてはるかに定義の自由度が大きい。よって客観的な評価が難しいと考えられ、ここでは直接コーディング法のみを比較対象とした。

Fig. 3 Changes of (a) the number of modules, (b) the number of units per module, (c) connection density

純なものでなく、試行錯誤的に見出すことがそもそも難しい問題設定であったことがわかる。つぎに、直接コーディング法との比較であるが、記憶パターン数が少ない $P=12$ のケースでは連想性能に大きな差はないが、 $P=36$ のケースでは提案手法によって高い連想性能をもつ CCHN が見出されている。一般に記憶パターン数が多くなると記憶および想起が難しくなるため、連想性能がモジュール構造に大きく依存し、性能の高い CCHN を与える構造パラメータの集合が小さくなることは容易に想像できる。よってこの結果は、提案手法のほうが直接コーディング法に比べて、構造パラメータの空間を効率よく探索できたことを示している。適応度が最大であった CCHN のモジュール構造の一つを Table 3 に示す。

50 世代までに探索されたモジュール構造の多様性を検証するため、Fig. 3(a)-(c) にモジュール数、ユニット数、結合密度のそれぞれについて 50 個体の平均値、最大値、最小値の移り変わりを示す。なお Fig. 3(c) における結合密度の計算では、モジュール数 1 のケースを含めていない。よって、結合密度がゼロになっているのはモジュール数が 2 以上でありながらモジュール間結合をもたない CCHN が探索されていることを意味する。これらの図から、ここで取り上げた 3 種類の構造パラメータに対し、多様性の高い探索が行われているようすがわかる。本実験ではシミュレーション時間の関係上、

Table 3 Found modular architecture of individuals (CCHN) with maximum fitness. In 'connectivity', $n \rightarrow m$ means that there is a connection from the n th module to the m th module

	$P = 12$	$P = 36$
num. of modules	3	2
num. of units	66, 8, 26	92, 8
connectivity	$1 \rightarrow 2, 2 \rightarrow 1,$ $2 \rightarrow 3, 3 \rightarrow 1,$	$1 \rightarrow 2, 2 \rightarrow 1$

Table 4 Percentage of success recall for different γ s

	success rate	connect. density
$\gamma = 0.0$	73.5	72.2
$\gamma = 0.25$	71.8	41.7
$\gamma = 0.5$	66.4	30.6

Fig. 4 Changes of connection density for different γ

50 個体の集団を 50 世代にわたってしか探索を行わなかったが、これだけでも優れたモジュール構造が見出されたのは、ここで提案した遺伝子型がモジュール構造の探索に適した表現になっていることを示唆している。

つぎに、構造上の制約を考慮した場合について、 $P = 12$ に対する連想性能の結果を Table 4 に示す。前述の結果同様、3 種類の初期値集合に対して、それぞれ 4000 回の想起実験における成功率の平均を求めた。また、Table 4 には最良個体のモジュール間結合密度の平均も示している。これからわかるように、構造上の制約を考慮しない $\gamma = 0$ のとき最も性能が高いが、 $\gamma = 0.25$ のときに高性能性と疎結合性の両方を満たす優れたモジュール構造が見出されている。ただし、高性能性と疎結合性がうまくバランスされた個体を見出すには、今のところ γ を試行錯誤的に決めるしかない。これについては、パラメータの最適化も含めたメタ GA の導入などを今後検討する必要があると思われる。最後に、パラメータ γ が探索過程において適切に構造上の制約を課しているようすを確認するため、Fig. 4 にモジュール間結合密度の移り変わりを示しておく。 γ が大きくなるにつれ、結合密度の値が最終的に小さくなるようすがわかる。

5. まとめ

動的な特性をもつモジュール型ニューラルネットの一つであるクロス結合ホップフィールドネット (CCHN) に対し、そのモジュール構造を遺伝的アルゴリズムにより自動的に決定する方法を提案した。モジュール構造を決定するパラメータには、モジュール数、各モジュールのユニット数、そしてモジュール間結合の形態を取り上げた。最も単純な直接コーディング法では、遺伝子長が総ユニット数 N に対して $O(N^2)$ となるのに対して、本論文で提案した遺伝子型表現では $O(N)$

でコーディングできる。CCHNを連想記憶に適用した結果、提案した方法によって高い連想性能をもつものが見出されただけでなく、構造上の制約(ここではモジュール間結合の疎結合性)を同時に考慮したCCHNが見出された。このことは、提案した遺伝子型が単に表現型の情報をコンパクトにコーディングしているだけでなく、優れたモジュール構造の探索を容易にする表現になっていることを示唆している。

ただし、この結論がどの程度の問題に対して成り立つかは今のところ不明であり、これについて今後さらに検討を要する。特に、提案した遺伝子型によって任意の構造パラメータを同時に探索できるわけではない点に注意する必要がある。モジュール数およびモジュールのユニット数は任意のものを探索できるが、モジュール間結合形態の種類(つまりモジュール間結合形態に対する探索空間の大きさ)はユニット属性の数やそれを表す二進列の長さ、そしてルール $R_1 \sim R_3$ の定義に依存して決まる。様々な問題に対する本提案手法の有効性を示すには、モジュール間結合形態に対する探索空間を適切に設定できることが望ましく、この探索空間と上記のパラメータ(およびルール)との関係について理論的考察が必要である。また、本論文ではモジュール間に相互結合を許すネットワークに対して提案手法を適用したが、遺伝子型から表現型への変換ルールを工夫することでモジュールが階層的に結合されたネットワークにも適用できるものと思われる。これらについては今後の課題としたい。

参 考 文 献

- 1) 西川, 北村編著: ニューラルネットと計測制御, 朝倉出版 (1995)
- 2) M. I. Jordan and R. A. Jacobs: Hierarchical mixtures of experts and the EM algorithm, *Neural Computation*, **6**-2, 181/214 (1994)
- 3) B.-L. Lu and M. Ito: Decomposition and parallel learning of imbalanced classification problems by min-max modular neural network, *Proc. The Fifth Int. Conf. on Neural Information Processing (ICONIP98-Kitakyushu)*, **1**, 199/202 (1998)
- 4) 馬場, 小島, 小澤: ニューラルネットの基礎と応用, 共立出版 (1994)
- 5) K. Tsutsumi: Cross-Coupled Hopfield Nets via generalized-delta-rule-based internetworks, *Proc. Int. Joint Conf. on Neural Networks (IJCNN90-San Diego)*, **II**, 259/265 (1990)
- 6) K. Tsutsumi: Higher degree error backpropagation in Cross-Coupled Hopfield Nets, *Proc. Int. Joint Conf. on Neural Networks (IJCNN91-Seattle)*, **II**, 349/355 (1991)
- 7) S. Ozawa, K. Tsutsumi, and N. Baba: An artificial modular neural network and its basic dynamical characteristics, *Biological Cybernetics*, **78**-1, 19/36 (1998)
- 8) 小澤, 堤, 馬場: モジュール構造ニューラルネットから導出される自己想起型連想記憶モデルとその連想特性の多様性, システム制御情報学会論文誌, **10**-12, 668/678 (1997)
- 9) B. L. M. Happel and J. M. J. Murre: Design and evolution of modular neural network architectures, *Neural Networks*, **7**, 985/1004 (1994)
- 10) S.-B. Cho and K. Shimohara: Modular neural networks evolved by genetic programming, *Proc. Int. Conf. on Evolutionary Computation (ICEC96-Nagoya)*, 681/684 (1996)
- 11) J. J. Hopfield: Neural networks and physical systems with

emergent collective computational abilities, *Proc. Natl. Sci. USA*, **79**, 2254/2558 (1982)

[著 者 紹 介]

小 澤 誠 一 (正会員)

1965年1月8日生。89年神戸大学大学院工学研究科計測工学専攻修士課程修了。同年奈良工業高等専門学校情報工学科助手, 93年大阪教育大学教育学部教養学科情報科学専攻助手, 同講師を経て, 2000年1月より神戸大学大学院自然科学研究科助教授となり, 現在に至る。ニューラルネットによる知的情報処理の研究に従事。博士(工学)。電子情報通信学会, 日本神経回路学会, システム制御情報学会などの会員。

堤 一 義

1954年7月22日生。85年神戸大学大学院自然科学研究科システム科学専攻博士課程修了。学術博士。同大学院文部教官助手を経て, 90年より龍谷大学理工学部講師, 93年より同助教授, 99年より同教授となり, 現在に至る。その間94年10月より95年9月まで米国メリーランド大学客員助教授。主として, 人工神経ネットワーク, 知能ロボットなど, 脳の情報処理機能の解明とその応用に関する研究に従事。電子情報通信学会, 日本神経回路学会, システム制御情報学会, IEEE, International Neural Network Society 会員。

馬 場 則 夫 (正会員)

1974年京都大学大学院博士課程単位修了。同年徳島大学工学部情報工学科助手, 88年知能情報工学科教授, 90年大阪教育大学教育学部教授となり, 現在に至る。この間約2年間, フランスINRIA研究所およびオーストリアIIASA研究所客員研究員。学習オートマトン, ゲーミング, 数理計画法, ロボット工学, ニューラルネット等の研究に従事。工学博士。著書: *New Topics in Learning Automata Theory and Applications* (Springer-Verlag), *ニューラルネットの基礎と応用* (共立出版), *PC-9801 パソコンゲーミング手法* (日刊工業新聞), *数理計画法入門* (共立出版)。IEEE, INNS, システム制御情報学会, 日本シミュレーション&ゲーミング学会, 日本OR学会会員。