



# マルチモジュールニューラルネットとその連想特性に関する研究

小澤, 誠一

---

(Degree)

博士 (工学)

(Date of Degree)

1998-03-11

(Date of Publication)

2007-08-27

(Resource Type)

doctoral thesis

(Report Number)

乙2214

(JaLCD0I)

<https://doi.org/10.11501/3141259>

(URL)

<https://hdl.handle.net/20.500.14094/D2002214>

※ 当コンテンツは神戸大学の学術成果です。無断複製・不正使用等を禁じます。著作権法で認められている範囲内で、適切にご利用ください。



神戸大学博士論文

マルチモジュールニューラルネットと  
その連想特性に関する研究

平成10年1月

小澤 誠一

# 概 要

人工ニューラルネットの研究は，生物の情報処理メカニズムの解明やモデル化に重点を置いた「生物指向型」の研究と，生物の情報処理メカニズムを参考にして新たな情報処理様式の提案やシステムの高性能化を目指す「応用指向型」の研究とに大別される．工学の分野においては主に後者のアプローチが取られ，これまでパターン認識や制御，予測，信号処理，最適化など数多くの問題に適用されてきた．しかしながら，この応用事例の多くは単一の情報処理系として構成されたものが中心であり，「ネットワークの部位によって異なった情報処理様式をもったニューラルネット」，つまり複数の情報処理系(モジュール)が統合されたマルチモジュールニューラルネットを用いた事例はまだそれほど多くない．複数の階層型ニューラルネットを統合した静的なマルチモジュールニューラルネットを扱った事例はあるが，基本的に問題空間を分割して解く，いわゆる分割統治型のアプローチであって，モジュール間で行われる相互作用を積極的に利用するものではない．これまでの生物学的知見からも明らかなように，生物においては複数の情報処理系で構成されたニューラルネットによる動的な情報処理が中心である．応用指向型のニューラルネット研究であっても，新たな情報処理様式の提案は生物システムを参考にすることから生まれる．この意味で，複数個の動的な情報処理系で構成されたニューラルネット(動的マルチモジュールニューラルネット)を研究対象とすることは重要でかつ興味深い．そこで，本論文では動的マルチモジュールニューラルネットのモデル化手法を提案し，このモデルで実現される動特性の考察を行うことにした．また，本モデルの実用性に対する評価を行うために連想記憶に適用し，その連想特性および性能について調べることにした．

まず，第1章では本研究を行うに至った背景と目的について詳しく論じた．

第2章では，まず本モデルにおけるモジュールの情報処理と相互作用の機能について定義し，多様な相互作用を導入するためのクラスを定義した．動的マルチモジュールニューラルネットの情報処理をエネルギー関数で記述する方法について述

## II 概要

---

べ、これからネットワークに漸近安定性を保証する動作方程式 (ユニットの状態と結合荷重の更新則) を導出する方法について述べた。また、実際に 2 種類の相互作用が導入された動的マルチモジュールニューラルネットの動作方程式を導出した。さらに、モジュールの最終状態間に一対多の関係が生じる可能性について述べ、これを回避するために導入される自己参照型相互作用について述べた。

第 3 章では、マルチモジュールモデルに導入された相互作用の役割について調べるため、ホップフィールドネットに誤差エネルギー (相互作用に対するエネルギー関数) を付加した 1 モジュールモデルを導出した。そして、この 1 モジュールモデルと等価な情報処理を行う自己想起型連想記憶モデルを導き、この結合行列を固有値解析することでネットワークの動特性を考察する方法を提案した。この解析手法を用いて動特性を考察した結果、記憶パターン (モジュールの最終状態) が張る空間と直交するノイズ空間において、相互作用を導入することの効果が明確に現われることがわかった。また、このノイズ空間のダイナミクスはネットワークの状態からノイズ成分を除去する役割をもち、連想性能を従来手法に比べて飛躍的に向上させる。このことは、記憶パターンがランダムに選ばれた場合とクラスター構造をもつ両方の場合において確認された。また、ネットワーク全体の情報処理に対する誤差エネルギーの貢献度を変化させることで、記憶パターンへの引き込み領域が拡大縮小することがわかった。これは情報検索の柔軟性を本モデルに与えるものであり、連想記憶モデルとして重要な性質の一つである。

第 4 章では、マルチモジュールニューラルネットの基本モデルである 2 モジュールモデルを導出し、モジュール間に相互作用を導入したニューラルネットの基本動特性について調べた。状態空間におけるエネルギー関数の形状や引き込み領域などを調べた結果、モジュールは他方のモジュールの状態に依存して自己の情報処理様式をダイナミックに変化させることがわかった。このときモジュール間には競合が生じており、その競合を解消することで特定の最終状態を導く様子が観測された。また、記憶パターンの構造に関係なく、優れた連想性能を 2 モジュールモデルが有することをシミュレーション実験により確認した。また相互作用に非線形性を加えることで、連想性能がさらに向上することを示した。また、モジュールの最終状態に一対多の対応関係がある一例として文字パターン対の連想を取り上げ、自己参照型相互作用を取り入れたモデルが正しく動作することを確認した。

第 5 章では、異なる 2 種類の自己参照型相互作用 (大局的相互作用と局所的相互作用) を導入したマルチモジュールモデルが導出され、その動特性の考察を行った。第 3 章で提案した動特性の解析手法をここでも適用するため、マルチモジュールモデル

に対する等価モデルを導出した。等価モデルの結合行列を固有値解析し、動特性の考察を行った結果、2種類の相互作用が同時に導入されることで動特性が改善されることを確認した。また、この動特性の改善がノイズ空間における望ましいダイナミックスの形成に起因することを示した。また、(優れた連想性能) = (多様な引き込み特性の実現) という観点から、マルチモジュールモデルの連想記憶システムとしての評価が行われた。この結果、貢献度パラメータやモジュール数、さらに導入される相互作用の種類を変更することで、様々な連想特性(引き込み特性) が実現されることを示した。

第6章では、前章までに得られた結果をまとめ、本研究における結論と今後の課題について述べた。



# 目 次

1	序 論	1
1.1	本研究の背景	1
1.2	本研究の目的	7
2	マルチモジュールニューラルネットとその動作方程式	11
2.1	まえがき	11
2.2	マルチモジュールニューラルネットのモデル化	14
2.2.1	ネットワーク構造と情報処理機能	14
2.2.2	エネルギー関数による情報処理の記述	21
2.2.3	マルチモジュール CCHN の動作方程式	24
2.3	2種類の相互作用を導入したマルチモジュール CCHN	27
2.3.1	モデルの概要	27
2.3.2	エネルギー関数と動作方程式	31
2.3.3	動作方程式の取扱い	35
2.4	まとめ	38
3	誤差エネルギーをもつホップフィールドモデル — 1モジュールモデル	41
3.1	まえがき	41
3.2	1モジュールモデルから導出される自己想起型連想記憶(1)	
	— 相関モデル	43
3.2.1	1モジュールモデル	43
3.2.2	自己想起型連想記憶モデルの導出	45
3.2.3	シミュレーションの方法と連想特性の評価方法	47
3.2.4	シミュレーション実験および考察	49

---

3.3	1 モジュールモデルから導出される自己想起型連想記憶 (2)	
	— 最適連想写像モデル . . . . .	55
3.3.1	自己想起型連想記憶モデルの導出 . . . . .	55
3.3.2	基本動特性の考察 . . . . .	57
3.3.3	シミュレーション実験および連想特性に対する考察 . . . . .	58
3.4	まとめ . . . . .	65
4	マルチモジュールニューラルネットの基本モデル — 2 モジュールモデル	69
4.1	まえがき . . . . .	69
4.2	基本動特性の考察 . . . . .	70
4.2.1	エネルギー関数の形状に対する考察 . . . . .	70
4.2.2	想起過程におけるエネルギー変動の考察 . . . . .	78
4.3	連想記憶への適用と性能評価 . . . . .	83
4.3.1	文字パターン対の連想 . . . . .	83
4.3.2	ランダムパターンに対する連想性能 . . . . .	86
4.3.3	クラスタ状パターンに対する連想性能 . . . . .	88
4.4	まとめ . . . . .	90
5	マルチモジュールニューラルネットの動特性に関する考察および連想記憶への適用	93
5.1	まえがき . . . . .	93
5.2	異種の相互作用を導入する効果 . . . . .	94
5.2.1	マルチモジュール CCHN-LAGI に導入される相互作用 . . . . .	94
5.2.2	連想性能の評価 . . . . .	95
5.2.3	結合行列の固有値解析 . . . . .	98
5.3	連想特性の多様性に対する考察 . . . . .	105
5.3.1	記憶パターンとシミュレーション方法 . . . . .	106
5.3.2	連想特性の多様性に対する考察 . . . . .	108
5.4	まとめ . . . . .	117
6	結論	121
A.1	マルチモジュール CCHN-LAGI の動作方程式	127
A.2	モデルの分類と略記方法	129



---

A.3 HN-COR における結合ダイナミックスの解	131
A.4 CCHN-LAGI/L( $M$ ) と等価な自己想起型連想記憶の結合行列	133
参考文献	135
本論文に関する発表論文	141



# 第1章 序 論

## 1.1 本研究の背景

1940年代における電子計算機の実用化とほぼ時期を同じくして、人工ニューラルネットの研究が始まったとされる [1]。この時期に行われた先駆的研究には、McCulloch & Pitts によるニューロンの数理モデル [2] や Hebb によるシナプスの可塑性に対する仮説 [3]、そして Hodgkin & Huxley によるニューロンの膜電位モデル [4] などがある。これらはともに、生物の神経系で起こっている現象を簡単な数理モデルによって表現しようと試みたものである。1950年代に入ると、Rosenblatt による文字識別機械 (パーセプトロン) [5]、Widrow らによる適応システム (アダライン) やデルタ則 [6] など、上記の数理モデルを工学システムとして発展させようとする試みが活発となる。これらでは、生物学的な意味合いや妥当性が必ずしも重要視されるわけではなく、高性能なシステムを構築するために、上記の数理モデルに対して様々な拡張が試みられる。よって、前者のアプローチでは生物で起こっている現象を説明するためにモデル化が行われるのに対して、後者のアプローチでは生物の情報処理をヒントにすることで従来の手法にない新たな情報処理様式の提案や性能の向上などが試みられる。例えば、階層型ニューラルネットにおいて最も多用されている逆誤差伝搬法 [7] (一般化デルタ則と呼ぶ場合もある) やその原型となった Widrow らによるデルタ則は、高速かつ正確な学習を可能とするようにヘップ則 (上記の Hebb 仮説をもとに考案されたシナプスの増強ルール) を改良したものである。また、Hopfield らは上記の McCulloch & Pitts 型ニューロンモデルに時間特性をもたせたモデルを相互結合型ニューラルネットに適用し、連想記憶 [8] や最適化問題 [9] などにおいて従来手法にない新しい情報処理様式を提供した。

このように人工ニューラルネットの研究は、生物の情報処理メカニズムの解明やモデル化に重点を置いた「生物指向型」の研究と、生物の情報処理メカニズムを参考にして新たな情報処理様式の提案やシステムの高性能化を目指す「応用指向型」

の研究とに大別される．もちろんこの中間に位置付けられる研究もある．高次機能を有するシステムを構築する際，生物学的知見がある部分に関してはそれを積極的に導入し，ない部分に対しては生物システムとしてなるべく矛盾が生じないような仮説(メカニズム)を導入することで全体を構成する「構成論的アプローチ」と呼ばれる手法がこれに相当する．この手法において最も成功した研究としては，福島によるネオゴグニトロン [10] が挙げられるであろう．

上記の分類に従うと，本研究は応用指向型の範疇に属する．そこで，以下では応用指向型の人工ニューラルネット(単にニューラルネットと呼ぶ)を前提に話を進めていくことにする．なお，人工ニューラルネットにおけるニューロンを生物のものと区別するため，以下では「ユニット」と呼ぶことにする．

1980年代以降におけるニューラルネットの研究には，

1. 状態が静的であるモデルから動的に変化するモデルへ
2. 単一の情報処理系で構成されたモデルから複数の情報処理系で構成されたモデルへ

という流れの変化がみられる [11]．1. の静的なモデルとはパーセプトロンや Kohonen の自己組織化モデル [12] などに代表される階層型ニューラルネットを指し，入力情報およびその出力の関係が静的な写像で表されるタイプである．このモデルは，パターン認識や予測，連想記憶，関数近似，故障診断など数多くの問題に適用され，現在でも最も多用されているタイプのニューラルネットである．しかし，入出力情報が動的な特性をもっている場合や逐次的な処理を通して解が求まるような場合には，静的なモデルよりもネットワークにダイナミクスをもたせた動的モデルが用いられる．このような動的モデルの適用は，制御や音声処理，連想記憶，最適化などで活発に行われている．

一方，2. の単一の情報処理系で構成されたモデルとは，ニューラルネットの構成要素(ユニットや結合荷重など)に対し，同一の情報処理様式に基づいて動作・学習が行われるものを意味する．つまり，階層型・相互結合型のニューラルネットに限らず，同一の学習アルゴリズムによってすべての結合荷重が変更され，同一の動作アルゴリズムによってユニットの状態が変更されるタイプがこれに相当する．この典型的な例は，逆誤差伝搬法などで学習される階層型ニューラルネットやホップフィールドネット [13] などである．これに対し，複数の情報処理系で構成されたニューラルネットモデルとは，

ネットワークにおける異なる部位が独立した情報処理系を実現し、これら部位が相互に影響を及ぼし合うことのできるニューラルネット

を意味する。ここで「部位」とは、ネットワーク全体の局所的な情報処理を担っている部分であり、これ自身も幾つかのユニットと結合荷重で構成されたニューラルネットである。このような部位は特定の機能を実現している処理単位と考えられ、以下では「モジュール」と呼ばれる。

神経科学において明らかにされている生理学的・解剖学的知見は、このモジュールという処理単位を導入することを前提としたものが多い [14][15][16][17]。例えば、聴覚抹消系から中枢に至るまでの情報処理系は、蝸牛神経核、上オリブ核、下丘、内側膝状体という複数の部位における情報処理に分離され、それぞれで統合・抽出された情報が脳皮質に伝わり、さらに高次の情報処理がなされる [15]。また脳皮質においても、視覚野、聴覚野、運動野など、さらにこれら領域をさらに細分化した部位に分けられている。脳の情報処理機能に対する理解は、主にこれら部位の情報処理を統合的に取り扱うことで行われているのが現状である。例えば、Hubel & Wiesel により示されたネコの視覚野におけるコラム構造 [18] などは、ネットワークにモジュールという構造を導入することで、その機能の理解に成功したよい例であるといえる。

複数の情報処理系で構成されたニューラルネットにおいて、モジュール間の信号のやりとり、すなわち「相互作用」を考慮することは重要である。脳における左右の脳半球は脳梁と呼ばれる神経線維の束を介して結合されているが、脳梁を取り去った患者に対する多くの解剖学的・心理学的実験から、右脳と左脳の相互作用を断ち切ることでいくつかの機能が失われることが示されている [17]。これは逆にいうと脳梁を取り去っても失われない機能があることを示しており、右脳と左脳という2つのモジュールが独立して実現している機能と両者の相互作用によってしか実現されない機能とがあることを意味する。このような相互作用の必要性を示す事例は、さらに小規模なモジュールである視覚野でも報告されている [16]。Zeki らは、視覚野の V1~V5 における結合関係について次のような特徴を指摘している。

- I. 最も重要な結合関係は divergent と convergent な結合である。
- II. convergent な結合は、さらに topical と confluent な convergence に分けられる。
- III. 他の重要な結合関係には、forward と backward の結合がある。

I. の divergent な結合とは，あるモジュールの信号が他の複数モジュールに伝達されるような経路を意味しており，convergent な結合とは，複数モジュールの信号が1つのモジュールに統合されるような経路を意味する．II. の topical convergence は同一経路内において信号が空間的に統合されることを表しており，confluent convergence は異なる経路における属性の異なる信号が統合されることを表している．III. は，I. と II. の結合経路が低次から高次の部位へという forward 方向だけでなく，高次から低次の部位へという backward 方向にも存在することを表している．これらの生理学的知見は，モジュール間相互作用を通して様々なかたちの情報統合が行われ，ダイナミックに信号の伝搬が行われることを示唆している．

以上から「複数個の動的な情報処理系で構成されたニューラルネット」は単一の静的な情報処理系で構成されたものに比べて，生物が取り入れている構造をより忠実に反映しているといえ，生物システムを参考に新たな情報処理様式の提案を目指す応用指向型の本研究者にとって非常に魅力的な対象となる．以下では，複数個の情報処理系で構成されたニューラルネットのことを「マルチモジュールニューラルネット」と呼ぶことにする．

マルチモジュールニューラルネットに関するいくつかの研究を紹介する．動的なマルチモジュールニューラルネットのモデルとして，Grossberg らは，外界からの入力パターンを受ける下位層とそのパターンのカテゴリを出力する上位層で構成された ART(Adaptive Resonance Theory) モデルを提案した [19]．このモデルの特徴は，下位から上位へのボトムアップ信号だけでなく，上位から下位へのトップダウン信号との相互作用を考慮して出力が決定される点にある．すなわち，このモデルは上位と下位の2つのモジュールとそれらの相互作用を導入した動的マルチモジュールニューラルネットの範疇に属する．このモデルのボトムアップ経路では外部入力に対応するカテゴリーパターンが競合によって上位層に生成され，トップダウン経路ではカテゴリパターンから下位層の入力パターンに対する推論結果が与えられる．両者の結果に矛盾が解消されるまで，下位 → 上位 → 下位という信号伝搬が繰り返される．また福島らは，特徴の局所的な位置ずれや歪みを吸収できる文字認識装置であるネオゴグニトロンにバックワードの経路を加えた選択的注意モデル [20] を提案した．このモデルの特徴は，入力層に同時に与えられたパターンに対して順に選択的注意を与えることで，例えば漢字の偏と旁を切り出すような認識を行えるところにある．選択的注意のメカニズムは，ある時刻において認識されたパターンに対し，その出力に関与するフォワード経路の信号をバックワード経路で抑制することで実現される．なお，静的なマルチモジュールニューラルネットの研究は数多く報告さ

れている。これらの多くは、独立して学習を行った複数の階層型ニューラルネットを階層的に結合し、その出力情報を統合するものである。つまり与えられた問題を部分問題(サブタスク)に分割し、それぞれを小規模なネットワークに学習させた後、それらの結果を統合して元の問題の解を出力する。このようにモジュール化することで、大規模な問題に対しても高速な学習を可能にしたり、ローカルミニマムに捕捉される可能性を小さくすることができる [21]。このアプローチの応用事例は、音声認識 [22][23] や文字認識 [24][25]、診断 [26]、回帰問題 [27] など多岐にわたっている。

以上からわかるように、現在のところ動的マルチモジュールニューラルネットの研究は、静的なものに比べて、かなり数が少ない。この主な原因には、ニューラルネットに行わせたい動作が静的なものに比べて一般に複雑となることが挙げられる。上述のように静的なモデルを適用する目的は、1つの問題を部分問題に分割し、それらを複数のネットワークに分担させることで問題を簡単にかつ高速に行う点にある。つまり基本的に分割統治型のアプローチとして利用されることがほとんどである。これに対して、動的なモデルでは、異なる情報処理を行う複数のモジュールが相互に競合や協調などを行うことで、単モジュールでは実現困難な情報処理を実現しようと試みられる。この場合、モジュールや相互作用をどのような情報処理様式で記述したらよいかは、試行錯誤的に決められることが殆どである。このことから、動的マルチモジュールニューラルネットに関するこれまでの研究の問題点は以下のようにまとめられる。

- (i) ある機能をマルチモジュールニューラルネットで実現するためには、適した情報処理様式(ユニットや結合荷重などの変更則)を試行錯誤的に見つけ出す必要がある。
  - (ii) (i) が原因となって、多数のモジュールで構成されたものや複雑な相互作用をもったものをモデル化しにくい。
- (ii) における複雑な相互作用とは、例えば、p.3で紹介した視覚野のモジュール間結合に関する解剖学的知見に基づくようなものである。

さて、人間がもつ最も基本的な機能として「連想」や「記憶」が挙げられる。生物の記憶メカニズムは生理学や解剖学だけでなく、心理学、計算機科学といった様々な分野において研究が進んでいる。連想記憶の応用事例には、例えば以下のようなものが挙げられる。

- (1) 高速なデータアクセスを可能とするコンピュータの記憶装置

(2) 高速な情報検索を可能とするマルチメディア情報データベースシステム

(3) ノイズの付加や欠落によって失われた情報の復元を行うシステム

(3) のシステムは、例えばパターン認識装置や画像処理装置などの前処理システムとして使われる。

ニューラルネットを用いた連想記憶モデルは、大きく相互想起型と自己想起型に分かれる。入力パターンを  $x$  とし、これと関連付けたい出力パターンを  $y$  としたとき、 $x$  と  $y$  が同一のパターンで表される  $x \rightarrow x$  の変換は自己想起型の連想と呼ばれる。これに対して、 $x$  と  $y$  が異なる  $x \rightarrow y$  の変換は相互想起型の連想と呼ばれる。相互想起型連想記憶には、 $x \rightarrow y$  の1回の変換で出力を求めるもの [12][28][29][30][31] と  $x \leftrightarrow y$  の双方向の変換を繰り返すことで出力を求めるもの [32][33][34] がある。一方、自己想起型連想記憶では出力がフィードバック結合を介して新たな入力とされることが多く、この再帰的な信号伝達を通して状態が安定化され、最終出力が求められる。自己想起型連想記憶としてよく用いられるのは、ユニットが完全相互結合され、その結合が対称行列で表されるホップフィールドネット (HN と略す) である [8]。HN の結合行列には相関行列および直交行列を用いることが多く、特に前者を結合行列とした自己想起型連想記憶モデルについては、記憶容量 [8][35][36] やダイナミクス [37][38][39][40] に関する解析などが盛んに行われてきた。また、記憶容量や引き込み領域の拡大を目的とした連想特性の改善手法も数々提案されている [41][42][43]。

このようにニューラルネットを用いた連想記憶モデルは、かなり古くから精力的に行われているものの、前述した応用事例のうち (1) や (2) のように高い信頼性が問われるものには、ルックアップテーブル型の連想記憶がよく用いられる。この理由はニューラルネットモデルの記憶容量がかなり小さいことと、不完全パターンからの想起能力がそれほど高くないことが挙げられる [44]。上記のルックアップテーブル型連想記憶をニューラルネットによって効率よく実現した研究事例もあるが [45]、本来、ニューラルネットを用いた連想記憶モデルに対して求められているものは、単に記憶容量や想起能力だけでなく

(a) 並列・分散処理による情報検索の高速性

(b) 柔軟な学習・適応能力

(c) システムの自己組織性

(d) 情報検索の柔軟性



といったものがあると思われる．このように柔軟でかつ高速な連想記憶システムを構築する上で，ルックアップテーブル型だけでなくニューラルネットを用いた連想記憶モデルの研究は，今なお重要かつ不可欠といえる．

## 1.2 本研究の目的

本研究の目的は大きく次の2つに分けられる．

1. 多様な情報処理を可能とする動的マルチモジュールニューラルネットの提案とその動特性の考察
2. 連想記憶への適用とその連想性能の評価

前述したように，動的マルチモジュールニューラルネットの多くでは，ネットワークの動作や学習の方法，さらには各種パラメータを何度となく調整した上で，試行錯誤的にその情報処理様式が決定される．また，このような試行錯誤的アプローチは，多数のモジュールで構成されたネットワークや様々な結合様式が導入されたネットワークのモデル化を困難にし，より高度な情報処理を実現する上での妨げとなり得る．より効率的なモデル化を行うためには，ネットワーク構成や各部位の情報処理様式などになるべく依存しない設計手法を考案することが重要である．また，できるだけ少ない試行錯誤で適切なネットワークの動作方程式 (ユニットの状態と結合荷重の変更則) を導出できるものが望ましい．

上記 1. の主な目的は，このような動的マルチモジュールニューラルネットに対するモデル化手法の提案にある．このモデル化手法では，モジュールや相互作用の情報処理様式をエネルギー関数として個別に記述し，それを組み合わせることでネットワーク全体の情報処理を記述する．これは堤のクロス結合ホップフィールドネット [46] で行われているモデル化手法を参考としており，以下の点でマルチモジュールニューラルネットとしての拡張が行われている．

- 任意のモジュール数をもつネットワークに対応できる．
- 様々なモジュール間相互作用の導入が可能となっている．

また 2. の目的は，提案した動的マルチモジュールニューラルネットの実用的な問題に対する有効性を示すことにある．よって，連想記憶の研究でよく用いられているランダムパターンだけでなく，現実的な記憶パターンの構造を反映したクラスタ

状パターン対しても性能を評価する．さらに p.6(a)-(d) で示した連想記憶特性のうち，本モデルにおける「情報検索の柔軟性」についても議論を試みる．

本論文では，動的マルチモジュールニューラルネットとしての特徴が明らかになるよう，構造の単純なものから複雑なものへと順に議論を展開していく．各章の内容について，以下に簡単にまとめておく．

第2章では，まず本モデルにおけるモジュールの情報処理と相互作用の機能について定義し，多様な相互作用を導入するためクラスを定義する．これに基づいて，動的マルチモジュールニューラルネットの情報処理をエネルギー関数で記述する方法について述べる．また，このエネルギー関数からネットワークに漸近安定性を保証する動作方程式が簡単に導出できることを示し，実際に2種類の相互作用が導入された動的マルチモジュールニューラルネットの動作方程式を導出する．さらに，モジュールの最終状態間に一対多の関係が生じる可能性について述べ，これを回避するために導入される自己参照型相互作用について述べる．

第3章では，マルチモジュールモデルに導入された相互作用の役割について調べるため，ホップフィールドネットに誤差エネルギー（相互作用に対するエネルギー関数）を付加した1モジュールモデルを導出する．また，1モジュールモデルと等価な情報処理を行う自己想起型連想記憶モデルを導き，この結合行列に固有値解析を行うことでネットワークの動特性を解析する方法を提案する．この解析手法を用いて1モジュールモデルの動特性を考察し，考察結果をシミュレーション実験によって確認する．1モジュールモデルを連想記憶に適用し，記憶パターンがランダムである場合とクラスタ状に分布する場合の両方について，その連想性能を調べる．さらに1モジュールモデルにおける引き込み特性の可変性についても議論を行う．これは情報検索の柔軟性に関する議論につながると考えられる．

第4章では，マルチモジュールニューラルネットの基本モデルである2モジュールモデルを導出し，モジュール間に相互作用を導入したニューラルネットの基本動特性について調べる．状態空間におけるエネルギー関数の形状や引き込み領域などを調べ，相互作用がモジュールに与えるダイナミックな影響について考察する．2モジュールモデルを連想記憶に適用し，記憶パターンがランダムである場合とクラスタ状に分布する場合の両方について，その連想性能を調べる．また，モジュールの最終状態に一対多の対応関係がある一例として文字パターン対の連想を取り上げ，自己参照型相互作用を取り入れたモデルが正しく動作することを確認する．

第5章では、異なる2種類の自己参照型相互作用を導入したマルチモジュールモデルを導出し、その動特性を第3章と同様の方法によって考察する。このとき、2種類の相互作用が同時に導入されることの効果について主に議論していく。また、これまで調べられた1モジュールモデルと2モジュールモデルに対する動特性と照らし合わせ、マルチモジュールモデルの動特性に関する考察を深める。また、貢献度パラメータやモジュール数、導入される相互作用の種類を変更し、これらパラメータに対する連想特性の多様性(情報検索の柔軟性)についても調べる。

第6章では、前章までに得られた結果をまとめ、本研究における結論と今後の課題について述べる。



## 第2章 マルチモジュールニューラルネットとその動作方程式

### 2.1 まえがき

ニューラルネットワークに動的な特性をもたせるには，これまで次のような方法が用いられてきた．

1. ユニットの状態にダイナミクスを導入する．
2. ユニット間の信号伝搬に時間遅れ特性を導入する．

1. は，例えばユニットの特性として Hodgkin-Huxley 方程式や連続時間モデルを適用するものを指し，この場合，階層型ニューラルネットであっても動的な特性をもたせることが可能である．2. は動的な特性をもたせる最も一般的な方法であり，例えば相互結合型ニューラルネットにおいて，各ユニット間の信号伝播に1単位時間の遅延を与えるようなものに相当する（ユニット間で異なる遅延時間をもたせてもよい）．当然のことながら，1. と2. の両方を導入することも可能である．

これらの方法によって実現されるニューラルネットの動特性は，大きく次のように分類される．

- (i) 安定平衡点に収束するもの
- (ii) 周期振動を発生するもの
- (iii) 非周期振動を発生するもの

(i) の動特性をもつ代表的なネットワークはホップフィールドネット (Hopfield Network: HN)[47] である．HN は対称な結合荷重をもち，Fig.2.1(a) に示すように入力・出力ユニットを共通とする相互結合型ネットワークである．これまでに連想記憶 [8] や最適化問題 [9]，ロボットの軌道生成問題 [48][49] など多くの問題に適用されている．このモデルの特徴は情報処理がエネルギー関数という一種の評価関数で記述される点

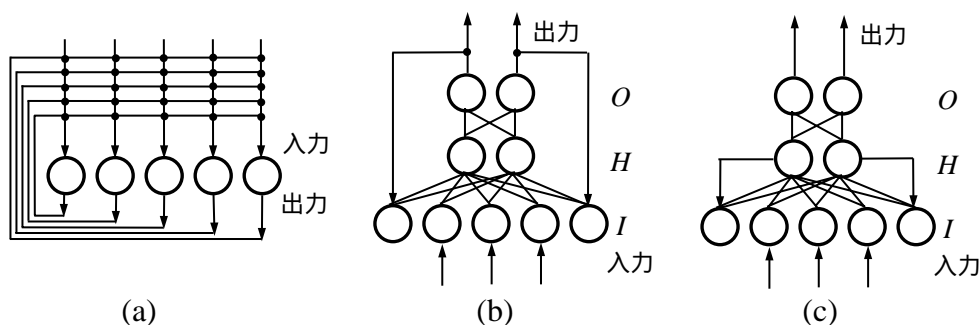


Figure 2.1: Three major types of recurrent neural networks: (a) Hopfield network, (b) Jordan network, (c) Elman network.  $I$ ,  $H$ , and  $O$  respectively mean input, hidden and output layers.

にあり，これによってネットワークの情報処理はエネルギー関数の最小化問題に帰着される．一般に，最終的に求めたい解がエネルギー関数の最小点に対応付けられる．但し，問題によっては最終的な解が前もって明示的には与えられないものもある．連想記憶の場合，最終的に想起したい記憶パターンはあらかじめ既知であることが多く，記憶パターンを最小点とするようなエネルギー関数を定義すればよい．しかしながら，例えば巡回セールスマン問題などの組み合わせ最適化問題では最適解はあらかじめわかっていない．このような場合は，解の評価を行う目的関数と解が満たすべき制約条件をエネルギー関数の形で記述する．

(ii)(iii) の動特性は結合を非対称としたネットワークで実現されることが多く，この場合，所望の情報処理を得るためにどのようなネットワーク構造をもたせ，どのような学習アルゴリズムを採用するかが問題となる．最も簡単なものは，階層型ニューラルネットの中間層出力または出力層出力を次時刻の入力として使う Figs.2.1(b)(c) のタイプである [50][51]．このようなネットワークでは，フィードバック結合に時間遅れもった固定結合を用いることが多く，フィードフォワード結合に対してのみ逆誤差伝搬法 [7] による学習が適用される．また，一般的な相互結合型ニューラルネットでは，任意のユニット間結合に 1 単位時間の遅れをもたせ，誤差逆伝搬法を拡張した Back Propagation Through Time や Real Time Recurrent Learning[52] などの学習アルゴリズム [53][54][55][56] が適用される．しかしながら，相互結合型ニューラルネットは学習時間や必要なメモリ量，解の精度など問題点も多く，実用的な問題への応用はそれほど多くはない [1][57]．

以上のニューラルネットでは、構成されるユニットや結合荷重に対して、同一の情報処理様式が適用される。この意味において、上記のニューラルネットは単一の情報処理系で構成されたモデルといえる。これに対し、複数個の情報処理系で構成されたニューラルネット(マルチモジュールニューラルネットと呼ぶ)は、単一情報処理系のモデルを複数個結合したものと見え、実現できる動特性の自由度も一般に大きい。そこで、本研究で提案するモデルがどのようなタイプの動特性をもったものであるかを明確にしておく必要がある。

本研究では、前述 (i) の動特性をもった動的マルチモジュールニューラルネットに限定する。つまり、ネットワーク全体が漸近安定な系で表されるものを取り扱う。このモデルでは、(ii) や (iii) の動特性を取り扱えるモデルに比べて実現できる動特性がかなり限定されてしまうことになるが、ネットワーク構成や各部位の情報処理様式などに依存しないモデル化が可能となる。これはネットワークの情報処理をエネルギー関数によって記述することで実現され、このエネルギー関数を単調に減少させる動作方程式(ユニットの状態や結合荷重の変更則)は一定の手順のもとで容易に導出できる。エネルギー関数の最小点は望ましい最終結果を表しているため、この意味で導出された動作方程式は適切なダイナミクスを与えるといえる。さらに、動作方程式はエネルギー関数を常に減少させるように選ばれることから、ネットワークの動特性がエネルギー関数の幾何学的特性として表されるという特徴ももつ。この性質は、3章以降において本モデルの動特性を考察する上で好都合となる。

本論文で提案する動的マルチモジュールニューラルネットモデルは、堤が提案したクロス結合ホップフィールドネット(Cross-Coupled Hopfield Nets: CCHN)[46]を拡張したものである。Fig.2.2に示すように、CCHNモデルは2つのHNを「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したものである。本モデルにおいて拡張されている点について、以下にまとめておく。

- 任意のモジュール数をもつネットワークに対応できる。
- 様々なモジュール間相互作用の導入が可能となっている。

2.2では、本モデルにおけるモジュールや相互作用の機能について定義し、導入される相互作用の種類(クラス)を定義する。これらに基づいて、様々な相互作用を導入したマルチモジュールニューラルネットのモデル化を行う。また、ネットワークに漸近安定性を保証する動作方程式をエネルギー関数から導く。2.3では、実際に2種類の相互作用を導入したマルチモジュールニューラルネットの動作方程式を求める。

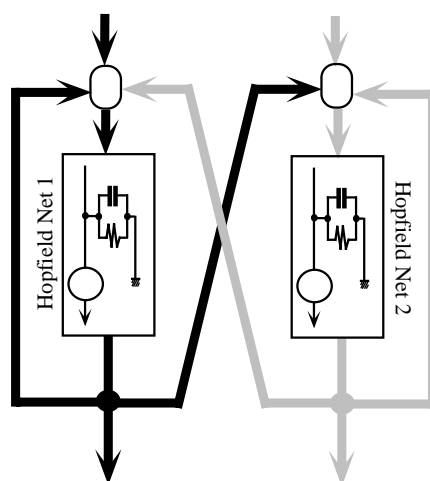


Figure 2.2: A schematic diagram of Cross-Coupled Hopfield Nets (CCHN).

さらに、モジュールの最終状態間に一対多の関係が生じる可能性について述べ、これを回避するために導入される自己参照型相互作用について述べる。

## 2.2 マルチモジュールニューラルネットのモデル化

### 2.2.1 ネットワーク構造と情報処理機能

本論文では、堤が提案したクロス結合ホップフィールドネット (Cross-Coupled Hopfield Nets: CCHN) モデルを拡張し、複数個のホップフィールドネット (Hopfield Network: HN) が相互結合された Fig.2.3 のようなネットワークを考える。

以下では、Fig.2.3 におけるそれぞれの HN を「モジュール」とし、ネットワーク全体を「マルチモジュール CCHN (multi-module CCHN)」と呼ぶことにする。モジュールは設計者によって与えられた情報処理を独立して行えるものとするが、他のモジュールから影響を受けるため、単に自己に与えられた情報処理を行うだけではない。このようにモジュール間で与え合う影響を、以下では「相互作用」と呼ぶ。なお各モジュールや相互作用の機能、さらにはネットワーク全体がもつべき機能は、設計者によってあらかじめ与えられるものと仮定する。但し、これらの機能が実現されるためには、何らかの方法 (通常は学習) によってモジュール内結合 (内部結合) およびモジュール間結合 (外部結合) を適切に調整する必要があるものとする。

注意 2.1 設計者がネットワーク全体の機能だけを与え、各モジュールや相互作用の



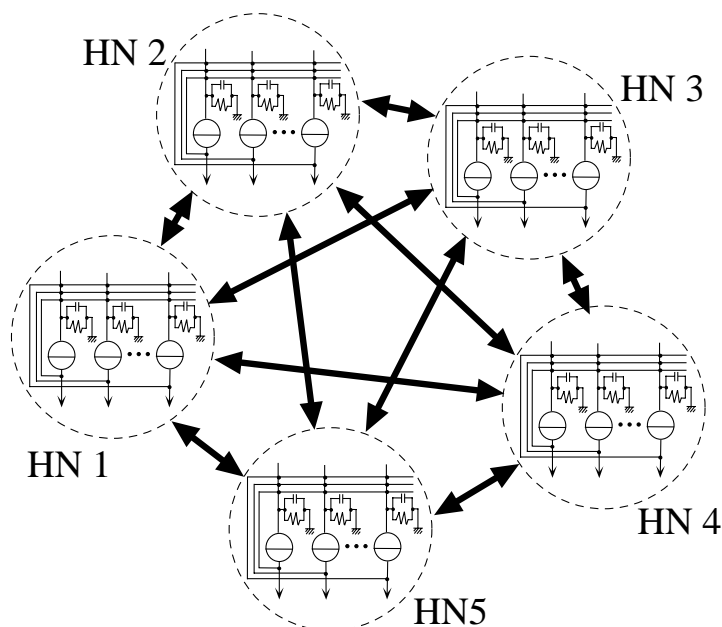


Figure 2.3: A schematic diagram of a multi-module CCHN

機能自体が適応的に獲得されるようなマルチモジュールニューラルネットを想定することも可能である。しかし、これにはモジュールと相互作用の機能、さらに処理するタスクを自動的に獲得するためのメカニズムを必要とし、この実現は一般に困難である。このようなアプローチも存在するが [58]、まだ研究として初歩的な段階にあり、本論文では扱わないこととした。

モジュールの情報処理は連続時間連続値型 HN の動作方程式 [47] に基づいて行われる。第  $m$  モジュールの動作方程式は HN で用いられる変数に  $(m)$  を付けることで、以下のように表される。

$$c^{(m)} \frac{du_i^{(m)}}{dt} = \sum_{j=1}^{N^{(m)}} T_{ij}^{(m)} v_j^{(m)} - \frac{u_i^{(m)}}{\tau^{(m)}} + J_i^{(m)} \quad (2.1)$$

$$v_i^{(m)} = g(ku_i^{(m)}) \quad (2.2)$$

$$(i = 1, \dots, N^{(m)}; m = 1, \dots, M)$$

ここで、 $M$  はマルチモジュール CCHN を構成するモジュールの個数、 $v_i^{(m)}$  および  $u_i^{(m)}$  は第  $m$  モジュールにおける第  $i$  ユニットの状態出力と内部ポテンシャルを表す。  $T_{ij}^{(m)}$  は第  $j$  ユニットから第  $i$  ユニットへの結合荷重、 $J_i^{(m)}$  は外部入力である。  $c^{(m)}$  は正の定数であり、ネットワークの状態変更の速度を決定する時定数に相当す

る．また， $r^{(m)}$  も正の定数で，これはネットワークから漏出する信号量を決定する抵抗値に相当する．式 (2.2) の  $g(\cdot)$  は有界かつ微分可能な単調増加関数であり，ユニットの「出力関数」と呼ばれる．Fig.2.4にモジュールの情報処理を行う連続時間型 HN の等価回路を示す．

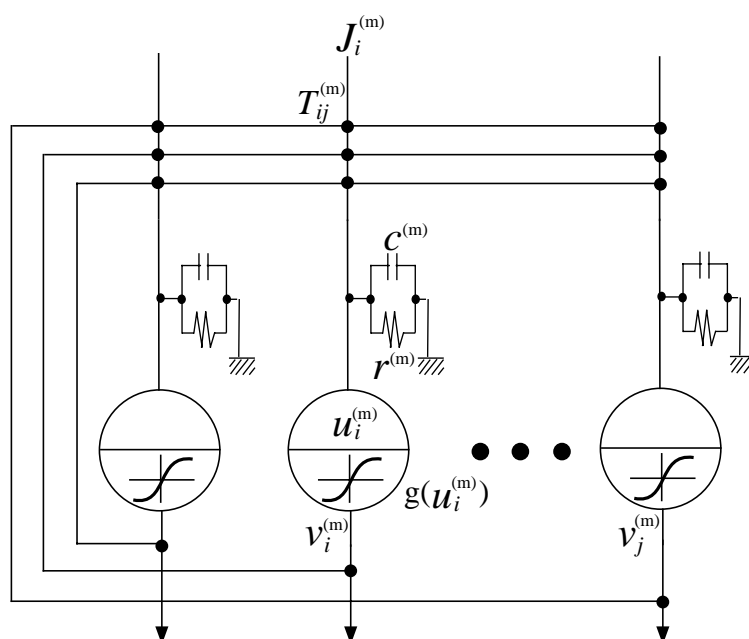


Figure 2.4: The equivalent electrical circuit which simulates the continuous time version of Hopfield network. This network is introduced as a module network.

内部結合の行列が対称 (つまり  $T_{ij}^{(m)} = T_{ji}^{(m)}$ ) であるときには，式 (2.1) の動作方程式は漸近安定な力学系 (ダイナミックス) を形成し，その安定平衡点はエネルギー関数と呼ばれる一種の評価関数の最小点として表される．つまり，モジュールの情報処理はエネルギー関数の最小点への収束プロセスとして表現される．ここでは，この最小点のことをモジュールの「最終状態」と呼ぶことにする．

本論文では，モジュールの情報処理機能を次のように定義する．

**定義 2.1 (モジュールの情報処理機能)** モジュールの情報処理を通して，モジュールの状態は設計者によって与えられた最終状態の一つに収束する．

但し，これはモジュールに与えられたメタレベルの機能を指しており，設計者が最終状態にどのような意味付けを行うかによって，その収束プロセスの意味付けが異なることに注意する．例えば巡回セールス問題の場合，最終状態は巡回経路に対応

し，上記の情報処理機能は最適な巡回経路の探索という意味付けがなされる．また連想記憶の場合，最終状態は記憶パターンに対応し，情報処理機能は記憶パターンの想起という意味付けがなされる．

次に相互作用の情報処理機能について考える．前述のように，個々のモジュールは漸近安定な特性をもっているが，与える相互作用の性質によってはネットワーク全体を漸近安定でない系にすることも可能である．しかし本論文ではこのような系は取り扱わず，マルチモジュール CCHN 全体が漸近安定な系で表されるものに限定する．さらに，マルチモジュール CCHN(ネットワーク全体)に望まれる最終状態は構成要素であるモジュールの最終状態の組合せで表されるものとする．

さて，以下の例を取り上げて，相互作用を導入することの意義について考えてみる．

例 2.1 3モジュールで構成されたネットワークを考え，各モジュールに次のような最終状態が与えられているとする．

$$\text{module 1 : } \boldsymbol{x}_1, \boldsymbol{x}_2 \quad \text{module 2 : } \boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3 \quad \text{module 3 : } z_1$$

ここでモジュール 2 および 3 に注目すると，これらのモジュールが取り得る最終状態の組合せは最大で以下の 3 通りある．

$$(\boldsymbol{y}_1, z_1) \quad (\boldsymbol{y}_2, z_1) \quad (\boldsymbol{y}_3, z_1)$$

モジュール 2 および 3 の最終状態が上記の 3 つの状態をとるようにする最も簡単な方法は，2 つのモジュール間に相互作用を導入せず，それぞれに独立した動作を行わせることである．このことから，設計者がモジュール間に相互作用を導入するのは，特定モジュールがとる最終状態の組合せを限定するためと考えることができる．このような限定は設計者にとって最終状態の組合せに何らかの意味付けを行っていることに相当し，以下では，この組合せを「ローカル・コンテキスト」と呼ぶことにする．いうまでもなく，設計者がネットワークにどのような情報処理機能を実現したいかによって，モジュール間に導入されるローカル・コンテキストは異なる．

また，モジュールの最終状態間にローカル・コンテキストが与えられれば，ネットワーク全体がとる最終状態も限定される．本例ではネットワーク全体の最終状態として計  $2 \times 3 \times 1 = 6$  通り考えられるが，モジュールの最終状態間にローカル・コンテキストが与えられることによって，結果的にその数は限定される．このように，モジュールの最終状態間にローカル・コンテキストを与えることで限定されるネットワーク全体の最終状態の組合せを「グローバル・コンテキスト」と呼ぶ．

さて、次にモジュールの最終状態間に与えられたローカル・コンテキストから相互作用をどのように実現するかについて述べる．モジュール間相互作用は、いうまでもなく、あるモジュールから他のモジュールへの信号伝搬を総称したものであり、信号の発生元と伝搬先が存在する．ここで、信号の発生元の違いによって、相互作用に対する次のようなクラスを導入する．

定義 2.2 (相互作用のクラス) あるモジュールは、モジュール間相互作用を介して  $k$  個のモジュール  $a_1, \dots, a_k$  に影響を受けているとする．このとき、この相互作用のクラスは  $\mathcal{I}_k$  であるという．

相互作用のクラスは、あるモジュールに相互作用を介して与えられる影響がどの程度グローバルな情報に基づいたものかを表している．つまり、 $k$  が大きいほどマルチモジュール CCHN のグローバル・コンテキストを反映した相互作用ということになる．以下に相互作用のクラスに対する一例を示す．

例 2.2 Fig.2.5 のような 3 モジュール CCHN を考える．モジュール間の信号伝搬経

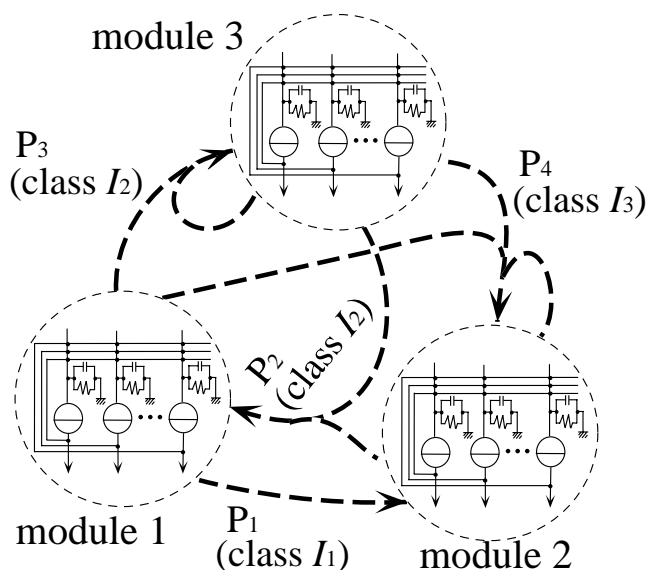


Figure 2.5: An example of a 3-module CCHN in which four interaction influences are introduced.

路は 4 種類あり、それぞれ  $P_i (i = 1, \dots, 4)$  の矢印で表されている．矢印の終点は影響が与えられるモジュールを指し、始点は信号発生元のモジュールを指している．例えば  $P_2$  の矢印は、第 2 モジュールと第 3 モジュールからの信号伝搬によって第 1

モジュールが影響を受けることを表している。定義 2.2 より、このような相互作用のクラスは  $\mathcal{I}_2$  である。第 3 モジュールが受ける影響もクラス  $\mathcal{I}_2$  であるが、自己の状態も参照している点に注意する。このような相互作用は「自己参照型相互作用」と呼ばれ、その特徴は次節において述べられる。第 2 モジュールは 2 種類の影響を受けており、それぞれクラス  $\mathcal{I}_1$  と  $\mathcal{I}_3$  に相当する。ここで、 $P_4$  の経路ではマルチモジュール CCHN 全体の状態が参照されており、最もグローバルな情報に基づく相互作用を表している。

上記の例のように、1 つのモジュールは異なるクラスの相互作用を同時に受けることも可能である。よって、あるモジュールが他のモジュールから受ける影響は、最大で  $\sum_{k=1}^M \binom{M}{k}$  種類考えられることになる。

いま、第  $m$  モジュールが相互作用を介して  $k$  個のモジュール  $a_1, \dots, a_k$  から影響を受けており（つまり、クラス  $\mathcal{I}_k$  に属する）、これらモジュールの最終状態ベクトル  $\xi^{(m)}, \xi^{(a_i)}$  ( $i = 1, \dots, k$ ) には、次のようなローカル・コンテキストが設計者によってあらかじめ与えられているとする。

$$(\xi^{(m)}, \xi^{(a_1)}, \dots, \xi^{(a_k)})$$

通常、上記のローカル・コンテキストは複数個存在するが、説明の簡単のため 1 つだけ考慮する。このとき、影響を与えられる第  $m$  モジュールと信号の発生元であるモジュール  $a_1, \dots, a_k$  の最終状態には、ローカル・コンテキストに基づいた次のような対応関係が存在する。

$$\xi^{(m)} \longleftarrow (\xi^{(a_1)}, \dots, \xi^{(a_k)})$$

この対応関係を一般的に表現するため、モジュール  $a_1, \dots, a_k$  の状態ベクトル  $v^{(a_i)}$  の関数として  $f^{(k,n,m)}(v^{(a_1)}, \dots, v^{(a_k)})$  を定義する。この関数は次式を満足する任意のベクトル関数である。

$$\xi^{(m)} = f^{(k,n,m)}(\xi^{(a_1)}, \dots, \xi^{(a_k)}) \quad (2.3)$$

関数  $f^{(k,n,m)}$  は単に式 (2.3) を満足するだけでなく、モジュール  $a_1, \dots, a_k$  が最終状態にないときにはローカル・コンテキストをなるべく反映した第  $m$  モジュールの状態を与えることが望ましい。このような関数の形状を設計者があらかじめ与えることも可能であるが、以下では汎化能力が期待できる階層型ニューラルネット (Fig.2.6 を参照) を用いて適応的に獲得する。つまり、関数  $f^{(k,n,m)}$  の値を階層型ニューラルネット (以下、インターネットワークと呼ぶ) の出力  $o^{(k,n,m)(L+2)}$  ( $L$ : 中間層数) で実現する。

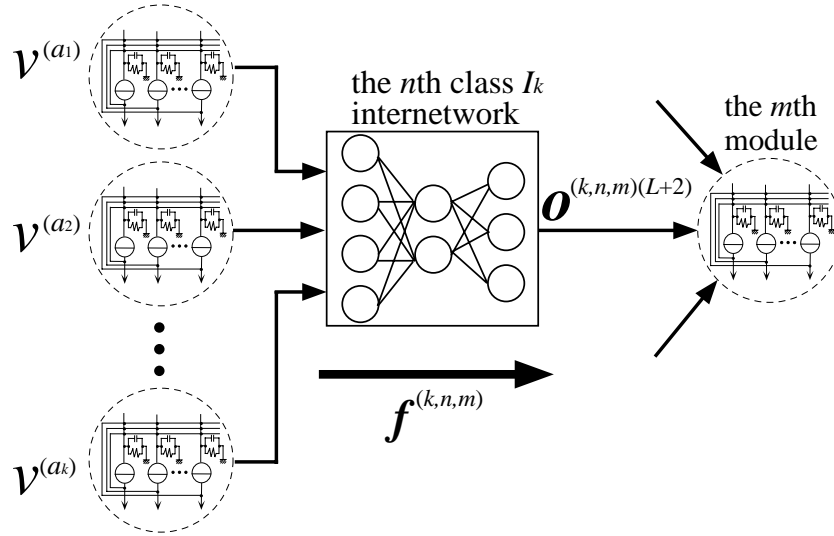


Figure 2.6: A conceptual diagram of the  $n$ th class  $\mathcal{I}_k$  internetwork.

注意 2.2  $f^{(k,n,m)}$  の上付き記号のうち,  $k$  は参照されるモジュールの個数 (つまり, クラス  $\mathcal{I}_k$ ) を表し,  $m$  は関数によって望ましい状態が与えられるモジュールの番号を表す. 第  $m$  モジュールに望ましい状態を与えるクラス  $\mathcal{I}_k$  の関数は最大  $\binom{M}{k}$  通り考えられるため,  $n$  はこれらを区別するためのインデックスである.

注意 2.3 Fig.2.6のインターネットワークにより  $f^{(k,n,m)}$  が獲得されるには, 連想記憶のケースのようにモジュールの最終状態が明示的に与えられる必要がある. 本論文では連想記憶への応用しか扱わないため, インターネットワークに式 (2.3) の対応関係を獲得させることにした. 最終状態が明示的に与えられないケース (例えば, 巡回セールスマン問題など) には, インターネットワークがどのような出力を行うのかをよく考える必要がある. また, この場合は以下で定義する相互作用のエネルギー関数も異なることに注意する.

インターネットワーク出力  $o^{(k,n,m)(L+2)}$  が第  $m$  モジュールの望ましい状態を与えるのであれば, 第  $m$  モジュールの状態は  $o^{(k,n,m)(L+2)}$  に近付くことが望まれる. つまり, モジュールの最終状態間に与えられたローカル・コンテキストを満足するよう, 第  $m$  モジュールの状態は変更されるべきである. 本論文では, この様な状態変化が相互作用によって行われると考える. そこで, モジュール間相互作用の機能を次のように定義する.

定義 2.3 (モジュール間相互作用の機能) モジュール間相互作用を通して, モジュー

ルの状態は設計者によって与えられたローカル・コンテキストの一つを満足するよう変化する。

但し、 $M$  モジュールモデルにおけるクラス  $\mathcal{I}_M$  の相互作用は、ローカル・コンテキストでなくグローバル・コンテキストを満足するように作用する。モジュールがクラス  $\mathcal{I}_M$  相互作用の影響を受けていない場合、マルチモジュール CCHN の最終状態に関するグローバル・コンテキストは明示的には与えられない。よって、モジュールはローカル・コンテキストを反映したいいくつかの相互作用を介して間接的にグローバル・コンテキストの情報を得る必要がある。

### 2.2.2 エネルギー関数による情報処理の記述

前述したように、マルチモジュール CCHN とその構成要素であるモジュールに漸近安定性を仮定している。このような場合、ネットワークの情報処理様式 (状態の時間発展) をエネルギー関数で記述するのが有効である。

第  $m$  モジュールのエネルギー関数  $E^{(m)}$  は式 (2.1) のリアプノフ関数として与えられ、各変数に  $(m)$  を付けて以下のように表される。

注意 2.4 ユニット状態  $v_i^{(m)}$  や出力関数  $g(\cdot)$  が有界であることから、 $E^{(m)}$  が下に有界であることは容易に示される。

[第  $m$  モジュールの情報処理]

$$E^{(m)} = -\frac{1}{2} \sum_{i=1}^{N^{(m)}} \sum_{j=1}^{N^{(m)}} T_{ij}^{(m)} v_i^{(m)} v_j^{(m)} + \sum_{i=1}^{N^{(m)}} \frac{1}{r^{(m)}} \int_0^{v_i^{(m)}} g^{-1}(v) dv - \sum_{i=1}^{N^{(m)}} J_i^{(m)} v_i^{(m)} + \frac{1}{4} \sum_{i=1}^{N^{(m)}} \sum_{j=1}^{N^{(m)}} T_{ij}^{(m)2} \quad (2.4)$$

ここで、右辺第 1 項はモジュール内部の情報処理に関するエネルギー項であり、フィードバック信号がモジュールの状態に影響しなくなったとき最小となる (つまり、 $\sum_j T_{ij}^{(m)} v_j^{(m)} = v_i^{(m)}$  が満たされ、ネットワークが安定化されたとき)。第 2 項はユニットのポテンシャルを発散させないためのエネルギー項である。第 3 項は外部入力に対するエネルギー項であり、やはり外部入力がモジュールの状態に影響しなくなったとき最小となる。第 4 項はモジュールの結合荷重を発散させないためのエネルギー項である。このエネルギー関数の最小点は結合荷重  $T_{ij}^{(m)}$  および外部入力  $I_i^{(m)}$  によって変化する。そこで、モジュールの最終状態がこの最小点に対応付けられるよう、何らかの方法で結合荷重や外部入力を調整する必要がある。

定義 2.3 より, 相互作用を通してモジュールの状態はあるローカル・コンテキストを満足するよう変化する. つまり, 第  $m$  モジュールの状態  $v^{(m)}$  がインターネットワーク出力  $o^{(k,n,m)(L+2)}$  に近づくよう, 相互作用のエネルギー関数を定義すればよい. そこで, 第  $m$  モジュールに影響を与えるクラス  $\mathcal{I}_k$  相互作用のうち,  $n$  番目のものに対するエネルギー関数  $E^{(k,n,m)}$  を次式で定義する.

[第  $m$  モジュールに影響を与えるクラス  $\mathcal{I}_k$  相互作用]

$$E^{(k,n,m)} = \frac{1}{2} \sum_{i=1}^{N^{(m)}} \left( v_i^{(m)} - o_i^{(k,n,m)(L+2)} \right)^2, \quad (2.5)$$

このエネルギー関数はモジュールの状態とインターネットワーク出力の誤差に基づいており, 以下では「誤差関数」とも呼ばれる. 当然ながら, モジュールの状態とインターネットワーク出力が一致したとき, このエネルギー関数は最小値をとる. インターネットワーク出力  $o^{(k,n,m)(L+2)}$  は具体的に以下の式で与えられる.

[インターネットワークの入力]

$$o_i^{(k,n,m)(1)} = \begin{cases} v_i^{(a_1)} & (i \leq \varphi(1)) \\ v_{i-\varphi(1)}^{(a_2)} & (\varphi(1) < i \leq \varphi(2)) \\ \vdots & \\ v_{i-\varphi(k-1)}^{(a_k)} & (\varphi(k-1) < i \leq \varphi(k)) \end{cases} \quad (2.6)$$

[インターネットワークの第  $l$  層出力]

$$o_i^{(k,n,m)(l)} = h(s_i^{(k,n,m)(l)}) \quad (2.7)$$

[インターネットワークの第  $l$  層内部ポテンシャル]

$$s_i^{(k,n,m)(l)} = \sum_{j=1}^{H^{(l)}} w_{ij}^{(k,n,m)(l-1)} o_j^{(k,n,m)(l-1)} \quad (2.8)$$

$$(m = 1, \dots, M; \quad l = 2, \dots, L+2)$$

但し,

$$\varphi(k) = \sum_{m=1}^k N^{(m)}$$

ここで,  $o_i^{(k,n,m)(l)}$  および  $s_i^{(k,n,m)(l)}$  はインターネットワーク第  $l$  層  $i$  番目のユニット出力および内部ポテンシャルを表し,  $w_{ij}^{(k,n,m)(l)}$  は第  $l$  層  $j$  番目のユニットから  $l+1$



層  $i$  番目のユニットへの結合荷重である．また  $M$  はモジュール数， $N^{(m)}$  は第  $m$  モジュールのユニット数， $h(\cdot)$  は微分可能なユニットの出力関数である． $L$  および  $H^{(l)}$  はインターネットワークの中間層数と中間ユニット数を表す．

マルチモジュール CCHN のエネルギー関数はそれぞれスカラーで表されるため，モジュールの情報処理や相互作用に対するエネルギー関数を自由に組合せてネットワークの情報処理を記述できる．本論文では，マルチモジュール CCHN のエネルギー関数がモジュールの情報処理と相互作用に対するエネルギー関数の線形和で表されるものとする．

[マルチモジュール CCHN のエネルギー関数]

$$\begin{aligned}
 E &= \sum_{m=1}^M \alpha^{(m)} E^{(m)} + \sum_{k=1}^M \sum_{m=1}^M \sum_{n=1}^{q^{(k,m)}} \beta^{(k,n,m)} E^{(k,n,m)}, \\
 &= \sum_{m=1}^M \alpha^{(m)} \left[ -\frac{1}{2} \sum_{i=1}^{N^{(m)}} \sum_{j=1}^{N^{(m)}} T_{ij}^{(m)} v_i^{(m)} v_j^{(m)} + \sum_{i=1}^{N^{(m)}} \frac{1}{r_i^{(m)}} \int_0^{v_i^{(m)}} g_i^{(m)-1}(v) dv - \sum_{i=1}^{N^{(m)}} J_i^{(m)} v_i^{(m)} \right. \\
 &\quad \left. + \frac{1}{4} \sum_{i=1}^{N^{(m)}} \sum_{j=1}^{N^{(m)}} T_{ij}^{(m)2} \right] + \sum_{k=1}^M \sum_{m=1}^M \sum_{n=1}^{q^{(k,m)}} \frac{\beta^{(k,n,m)}}{2} \sum_{i=1}^{N^{(m)}} \left( v_i^{(m)} - o_i^{(k,n,m)(L+2)} \right)^2 \quad (2.9)
 \end{aligned}$$

ここで  $q^{(k,m)}$  は，第  $m$  モジュールに影響を与えるクラス  $\mathcal{I}_k$  インターネットワークの個数を表す．また  $\alpha^{(m)}$  および  $\beta^{(k,n,m)}$  は正の定数で，モジュール情報処理とモジュール間相互作用が全体の情報処理に対してどの程度の重要度をもつのかを表す．以下では，このパラメータを「貢献度」と呼ぶことにする．なお， $E^{(m)}$  および  $E^{(k,n,m)}$  が下に有界であることから， $E$  も下に有界な関数であることに注意する．

右辺第 2 項からわかるように，このモデルでは様々なクラスの相互作用を同時に考慮でき，さらに貢献度パラメータを変化させることで膨大な数のバリエーションを表現できることになる．マルチモジュール CCHN モデルのバリエーションを与えることになるパラメータについて Table 2.1 にまとめておく．なお， $\beta^{(k,n,m)}$  によってモジュール間相互作用の貢献度が変更されるだけでなく，適当な  $\beta^{(k,n,m)}$  を零とすることでモジュール間結合をなくすこともできる点に注意する．つまり，これによって様々なモジュール間結合の形態を表現できることになる．以下では，上記のパラメータをマルチモジュール CCHN の「構造パラメータ」と呼ぶことにする．後述するように，この構造パラメータを様々に変化させることで，本モデルは多様な動的性質をもつことができる．

Table 2.1: Structural parameters of the proposed multi-module neural network.

number of modules	$M$
number of units in each module	$N^{(m)}$
interaction class	$\mathcal{I}_k$
number of hidden layers in internetworks	$L$
number of hidden units in internetworks	$H^{(l)}$
contribution of module information processing	$\alpha^{(m)}$
contribution of interactions connections among modules	$\beta^{(k,n,m)}$

### 2.2.3 マルチモジュール CCHN の動作方程式

モジュールと相互作用のエネルギー関数の線形和で記述されたマルチモジュール CCHN のエネルギー関数から，ネットワーク全体の漸近安定性を保証するような動作方程式を導く．上記のエネルギー関数は下に有界であるから，エネルギー関数の時間微分が負またはゼロとなるよう動作方程式を求めればよい．

エネルギー関数およびインターネットワーク出力は時間の関数であるが，式 (2.4), (2.5) および式 (2.6) ~ (2.8) から，状態や結合荷重，インターネットワーク出力の関数と考えることもできる．そこで，以下のように関数を書き換える．

[モジュールのエネルギー関数]

$$E^{(m)}(t) = \mathcal{E}^{(m)}(\mathbf{v}^{(m)}, \mathbf{T}^{(m)}) \quad (2.10)$$

[相互作用のエネルギー関数]

$$E^{(k,n,m)}(t) = \mathcal{E}^{(k,n,m)}(\mathbf{v}^{(m)}, \mathbf{o}^{(k,n,m)(L+2)}) \quad (2.11)$$

但し，

$$o_i^{(k,n,m)(L+2)}(t) = \mathcal{F}_i^{(k,n,m)}(\mathbf{v}^{(a_1)}, \dots, \mathbf{v}^{(a_k)}, \mathbf{w}^{(k,n,m)(1)}, \dots, \mathbf{w}^{(k,n,m)(L+1)}) \quad (2.12)$$

ここで， $\mathbf{T}^{(m)}$ ， $\mathbf{v}^{(m)}$ ， $\mathbf{w}^{(k,n,m)(l)}$ ， $\mathbf{o}^{(k,n,m)(L+2)}$  はすべて時間  $t$  の関数である．インターネットワークには多層ニューラルネットを用いることが可能であり，この場合， $o_i^{(k,n,m)(L+2)}$  は入力や結合荷重に関する非線形関数  $\mathcal{F}_i^{(k,n,m)}$  として表されることになる．

注意 2.5 モジュールの外部入力  $J$  は定数として取り扱う。

式 (2.10),(2.11) の時間微分は，式 (2.4),(2.5) のエネルギー関数を用いて次のように求められる。

$$\frac{dE^{(m)}}{dt} = \sum_{i=1}^{N^{(m)}} \frac{\partial \mathcal{E}^{(m)}}{\partial v_i^{(m)}} \frac{dv_i^{(m)}}{dt} + \sum_{i=1}^{N^{(m)}} \sum_{j=1}^{N^{(m)}} \frac{\partial \mathcal{E}^{(m)}}{\partial T_{ij}^{(m)}} \frac{dT_{ij}^{(m)}}{dt} \quad (2.13)$$

$$\begin{aligned} \frac{dE^{(k,n,m)}}{dt} &= \sum_{i=1}^{N^{(m)}} \frac{\partial \mathcal{E}^{(k,n,m)}}{\partial v_i^{(m)}} \frac{dv_i^{(m)}}{dt} - \sum_{i=1}^{N^{(m)}} \frac{\partial \mathcal{E}^{(k,n,m)}}{\partial o_i^{(k,n,m)(L+2)}} \frac{do_i^{(k,n,m)(L+2)}}{dt} \\ &= \sum_{i=1}^{N^{(m)}} e_i^{(k,n,m)} \left( \frac{dv_i^{(m)}}{dt} - \frac{do_i^{(k,n,m)(L+2)}}{dt} \right) \\ &= \sum_{i=1}^{N^{(m)}} e_i^{(k,n,m)} \left[ \frac{dv_i^{(m)}}{dt} - \sum_{q=1}^k \sum_{j=1}^{N^{(a_q)}} \frac{\partial \mathcal{F}_i^{(k,n,m)}}{\partial v_j^{(a_q)}} \frac{dv_j^{(a_q)}}{dt} \right. \\ &\quad \left. - \sum_{l=1}^{L+1} \sum_j \sum_p \frac{\partial \mathcal{F}_i^{(k,n,m)}}{\partial w_{jp}^{(k,n,m)(l)}} \frac{dw_{jp}^{(k,n,m)(l)}}{dt} \right] \end{aligned} \quad (2.14)$$

但し，

$$e_i^{(k,n,m)} = v_i^{(m)} - o_i^{(k,n,m)(L+2)} \quad (2.15)$$

ここで，モジュールは対称な結合荷重をもつことに注意する（つまり  $T_{ij}^{(m)} = T_{ji}^{(m)}$ ）。式 (2.9) を時間微分したものに式 (2.13),(2.14) を代入し， $dv_i^{(m)}/dt$ ,  $dT_{ij}^{(m)}/dt$ ,  $dw_{jp}^{(k,n,m)(l)}/dt$  について整理すると，以下ようになる。

$$\begin{aligned} \frac{dE}{dt} &= \sum_{m=1}^M \sum_{i=1}^{N^{(m)}} D_i^{(m)}(\mathbf{v}^{(m)}, \mathbf{T}^{(m)}, \hat{\mathbf{v}}^{(m)}, \hat{\mathbf{e}}^{(m)}, \hat{\mathbf{w}}^{(m)}, \alpha^{(m)}, \hat{\beta}^{(m)}) \frac{dv_i^{(m)}}{dt} \\ &\quad + \sum_{m=1}^M \sum_{i=1}^{N^{(m)}} \sum_{j=1}^{N^{(m)}} \alpha^{(m)} \frac{\partial \mathcal{E}^{(m)}}{\partial T_{ij}^{(m)}} \frac{dT_{ij}^{(m)}}{dt} \\ &\quad - \sum_{k=1}^M \sum_{m=1}^M \sum_{n=1}^{q^{(k,m)}} \sum_{l=1}^{L+1} \sum_j \sum_p \beta^{(k,n,m)} \left[ \sum_{i=1}^{N^{(m)}} e_i^{(k,n,m)} \frac{\partial \mathcal{F}_i^{(k,n,m)}}{\partial w_{jp}^{(k,n,m)(l)}} \right] \frac{dw_{jp}^{(k,n,m)(l)}}{dt} \end{aligned} \quad (2.16)$$

ここで  $D_i^{(m)}$  における  $\hat{\mathbf{v}}^{(m)}$  は，第  $m$  モジュールに影響を与えているすべてのインターネットワークの入力（モジュールの状態ベクトル）をまとめたものである。 $\hat{\mathbf{e}}^{(m)}$  および  $\hat{\mathbf{w}}^{(m)}$ ,  $\hat{\beta}^{(m)}$  は，同じく第  $m$  モジュールに影響を与えているすべてのインターネットワークにおける誤差信号，結合荷重，貢献度パラメータをまとめたものである。このように  $D_i^{(m)}$  はモジュール間の結合形態や導入されているクラスの種類などに依存するため，ここではどのような変数の関数であるかのみを示している。

さてマルチモジュール CCHN の動作方程式を導出するには，式 (2.16) が常に 0 または負となればよい。これを保証する一つの方法は，モジュールの状態  $v^{(m)}$  および

モジュールの結合  $T^{(m)}$  , インターネットワークの結合  $w^{(k,n,m)}$  を以下のような微分方程式に基づいて変更することである .

$$c \frac{du_i^{(m)}}{dt} = -D_i^{(m)}(v^{(m)}, T^{(m)}, \hat{v}^{(m)}, \hat{e}^{(m)}, \hat{w}^{(m)}, \alpha^{(m)}, \hat{\beta}^{(m)}) \quad (2.17)$$

$$\varepsilon \frac{dT_{ij}^{(m)}}{dt} = -\frac{\partial \mathcal{E}^{(m)}}{\partial T_{ij}^{(m)}} \quad (2.18)$$

$$\gamma \frac{dw_{jp}^{(k,n,m)(l)}}{dt} = \sum_{i=1}^{N^{(m)}} e_i^{(k,n,m)} \frac{\partial \mathcal{F}_i^{(k,n,m)}}{\partial w_{jp}^{(k,n,m)(l)}} \quad (2.19)$$

ここで ,  $c, \varepsilon, \gamma$  はそれぞれ正の時定数である . 式 (2.17) は第  $m$  モジュール  $i$  番目のユニットにおける内部ポテンシャルのダイナミクスに相当し , 以下では「活性ダイナミクス」と呼ぶことにする . マルチモジュール CCHN における各モジュールの情報処理が , 自己の情報処理だけでなく相互作用を介して他のモジュールの情報処理にも影響を受けることは式 (2.17) から明らかである . また , 式 (2.18),(2.19) はそれぞれモジュールとインターネットワークの結合変更則 (学習則) を表しており , 以下では「結合ダイナミクス」と呼ぶことにする .

これらの動作方程式によって , 式 (2.16) が常に負または 0 になることは , 式 (2.17) ~ (2.19) を式 (2.16) に代入し ,  $v_i^{(m)} = g(ku_i^{(m)})$  の関係において  $g(\cdot)$  が単調増加関数であることに注意すれば明らかである .

$$\begin{aligned} \frac{dE}{dt} = & - \sum_{m=1}^M \sum_{i=1}^{N^{(m)}} ck g'(ku_i^{(m)}) \cdot \left( \frac{du_i^{(m)}}{dt} \right)^2 - \sum_{m=1}^M \sum_{i=1}^{N^{(m)}} \sum_{j=1}^{N^{(m)}} \alpha^{(m)} \varepsilon \cdot \left( \frac{dT_{ij}^{(m)}}{dt} \right)^2 \\ & - \sum_{k=1}^M \sum_{m=1}^M \sum_{n=1}^{q^{(k,m)}} \sum_{l=1}^{L+1} \sum_j \sum_p \beta^{(k,n,m)} \gamma \cdot \left( \frac{dw_{jp}^{(k,n,m)(l)}}{dt} \right)^2 \leq 0 \end{aligned} \quad (2.20)$$

ユニットの活性ダイナミクスを表す関数  $D_i^{(m)}$  は , モジュール間結合の形態や導入する相互作用のクラスなどによって様々である . そこで次節では , マルチモジュール CCHN の具体例を一つ挙げて , エネルギー関数から動作方程式の導出を実際に行ってみる . これによって , 活性ダイナミクスや結合ダイナミクスの特徴がより明確になるとと思われる .

## 2.3 2種類の相互作用を導入したマルチモジュール CCHN

### 2.3.1 モデルの概要

2.2では、様々なクラスの相互作用を導入できるマルチモジュール CCHN を提案した。また、このエネルギー関数の定義と動作方程式の導出方法について述べた。マルチモジュール CCHN が表現できるネットワーク構造のバリエーションが膨大であることから、動作方程式の一般形を具体的な式で記述することは難しい。そこで本節では、導入する相互作用のクラスを2種類に限定したマルチモジュール CCHN を一例として取り上げ、そのエネルギー関数や動作方程式を具体的な形で示していく。ここで取り上げるモデルはマルチモジュール CCHN が表現できるバリエーションのごく一例でしかないが、その導出を具体的に行ってみることは本モデルの特徴を理解する上で重要である。

導入する相互作用のクラスを2種類に限定した理由は、以下の点についてマルチモジュール CCHN の基本動特性を明らかにしたいためである。

1. 相互作用がモジュール (またはネットワーク全体) の情報処理に対して与える影響
2. 異種の相互作用が同時に導入されることの効果

これらの点について、モデルの本質を損なわずなるべく単純な形で考察するため、本節では性質が大きく異なると考えられる2種類の相互作用を導入する。このモデルはマルチモジュールモデルを取り扱う第5章で使われることになる。また第3,4章で取り扱われる1モジュールモデルと2モジュールモデルは、ここで求められるマルチモジュール CCHN の特別なケースに相当する。つまり、本節におけるマルチモジュール CCHN は単に一例を示すために取り上げられるのではなく、2.2で提案したマルチモジュール CCHN の動特性を次節以降で議論していために使われる。

導入する2種類の相互作用として、インターネットネットワーク入力が式(2.21),(2.22)で与えられるものを考える。これから明らかなように、導入される相互作用のクラスは  $\mathcal{I}_2$  と  $\mathcal{I}_M$  である。

[ $n$  番目のクラス  $\mathcal{I}_2$  インターネットワークの入力]

$$o_i^{(2,n,m)(1)} = \begin{cases} v_i^{(n)} & (i \leq N^{(n)}) \\ v_{i-N^{(n)}}^{(m)} & (N^{(n)} < i \leq N^{(n)} + N^{(m)}) \end{cases} \quad (n \neq m) \quad (2.21)$$

[クラス  $\mathcal{I}_M$  インターネットワークの入力]

$$o_i^{(M,1,m)(1)} = \begin{cases} v_i^{(1)} & (i \leq \varphi(1)) \\ v_{i-\varphi(1)}^{(2)} & (\varphi(1) < i \leq \varphi(2)) \\ \vdots & \\ v_{i-\varphi(M-1)}^{(M)} & (\varphi(M-1) < i \leq \varphi(M)) \end{cases} \quad (2.22)$$

但し,

$$\varphi(k) = \sum_{m=1}^k N^{(m)}$$

ここで,  $M$  はモジュール数である. 上記2種類のインターネットワークの構成を Figs.2.7(a)(b) に示しておく.

クラス  $\mathcal{I}_2$  と  $\mathcal{I}_M$  のインターネットワークでは, その定義から明らかなように, 前者は2つのモジュールの状態が参照され, 後者は構成されるすべてのモジュールの状態が参照される.  $M > 2$  のとき, クラス  $\mathcal{I}_2$  インターネットワークはネットワーク全体の局所的な情報を参照していることになり, その局所性は  $M$  が大きくなるにつれて高まる. これに対して,  $\mathcal{I}_M$  インターネットワークはネットワークの大域的な情報を常に参照している. そこで, 以下では前者のインターネットワークによって生成される相互作用を「局所的相互作用 (Local Interaction)」, 後者に対しては「大域的相互作用 (Global Interaction)」と呼ぶことにする.

注意 2.6  $M = 2$  のとき, クラス  $\mathcal{I}_2$  と  $\mathcal{I}_M$  のインターネットワークの入力情報は同じである. よって, これらから生成される相互作用も同じである.

第  $m$  モジュールが影響を受けるクラス  $\mathcal{I}_2$  インターネットワークは, 参照される入力情報の違いによって最大  $M(M-1)/2$  通り考えられる. しかしながら, そのすべてを導入すると様々な種類の局所的相互作用が生成されてしまい, モジュールへの影響を考察することが困難になる. そこで, 第  $m$  モジュールに影響を与えるクラス  $\mathcal{I}_2$  インターネットワークには, 第  $m$  モジュール自身の状態とその他1つのモジュールの状態を入力することとした(式(2.21)および図2.7(a)を参照). このように自己のモジュールの状態を参照するインターネットワークを「自己参照型」と呼ぶことにする. また, これら2種類の相互作用を導入したマルチモジュール CCHN モデルを Cross-Coupled Hopfield Nets with Local And Global Interactions (CCHN-LAGI) と呼ぶ.

注意 2.7 クラス  $\mathcal{I}_M$  インターネットワークも自己参照型である. すなわち, ここで導入される2つのインターネットワークはともに自己参照型といえる.

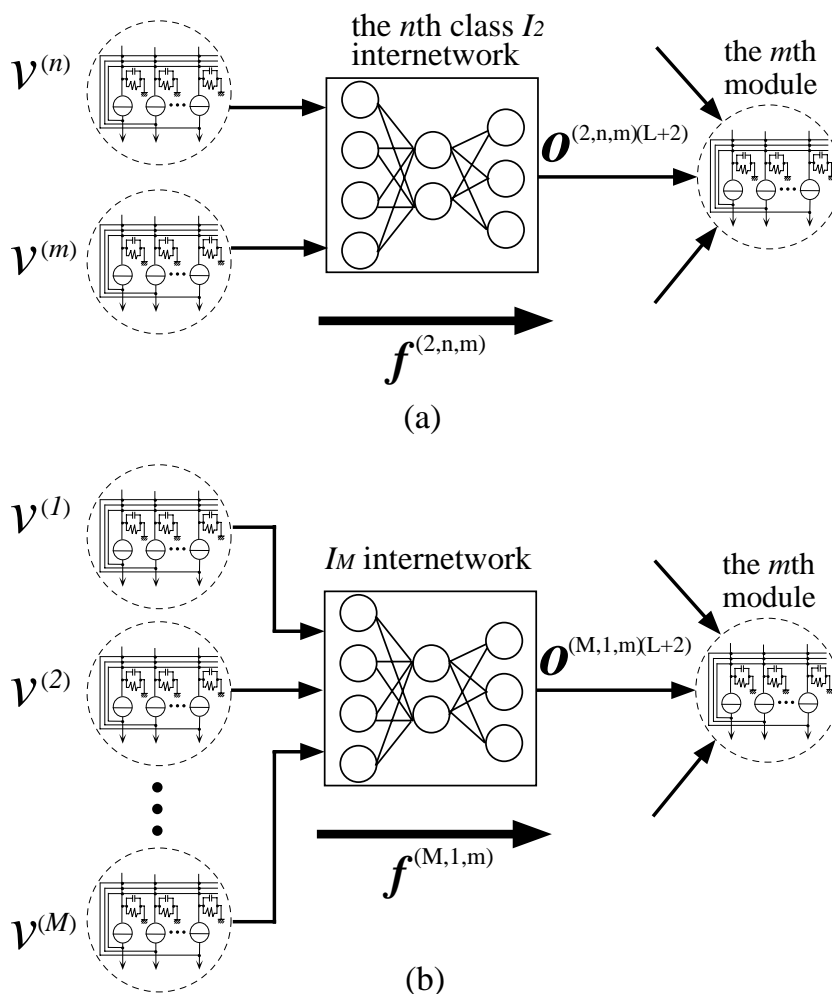


Figure 2.7: A schematic diagram of the introduced class (a)  $\mathcal{I}_2$  internetwork and (b)  $\mathcal{I}_M$  internetwork.

注意 2.8 クラス  $\mathcal{I}_2$  インターネットワークでは、自己のモジュールの状態以外に他の1つのモジュールの状態を参照する。式 (2.21) からわかるように、このもう一方のモジュールの番号とインターネットワークのインデックス  $n$  とを対応付けることにする。つまり  $o^{(2,n,m)(L+2)}$  は、第  $n$  番目と第  $m$  番目のモジュールの状態を入力として参照し、第  $m$  モジュールに影響を与えるインターネットワーク出力を意味する。また、自己のモジュールの状態だけを入力として参照するインターネットワークは考慮しないため (つまり  $n \neq m$ )、第  $m$  モジュールに影響を与えるクラス  $\mathcal{I}_2$  インターネットワークの個数は  $M - 1$  となる。一方、クラス  $\mathcal{I}_M$  インターネットワークは1つだけである。

さて，ここで上記のような自己参照型のインターネットワークに限った理由を説明する必要がある．まず次の例について考えてみる．

例 2.3 いま Fig.2.8 に示すような3モジュールのネットワークを考える．次に示すよ

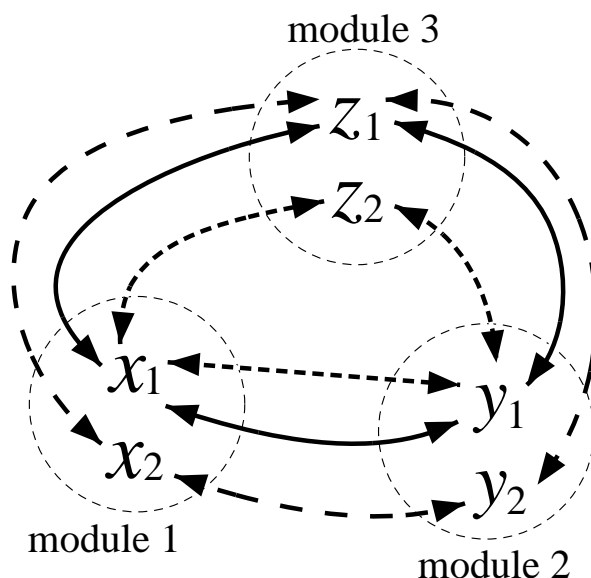


Figure 2.8: An example of a 3-module network and the relationships between the final states in modules.

うに，各モジュールは最終状態をそれぞれ2つもっているものとする．

$$\text{module 1 : } \mathbf{x}_1, \mathbf{x}_2 \quad \text{module 2 : } \mathbf{y}_1, \mathbf{y}_2 \quad \text{module 3 : } \mathbf{z}_1, \mathbf{z}_2$$

ネットワーク全体の最終状態間に次のような3種類の望ましい対応関係(グローバル・コンテキスト)があるものと仮定する．

$$(\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_1) \quad (\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_2) \quad (\mathbf{x}_2, \mathbf{y}_2, \mathbf{z}_1)$$

これらの対応関係は，Fig.2.8において3種類の矢印で示されている．

ネットワーク全体が上記のようなグローバル・コンテキストを満足するよう，マルチモジュールCCHNを設計するケースを考える．いま，module 3に影響を与えるクラス  $\mathcal{I}_2$  インターネットワークの入力を module 1 と module 2 の状態としたとき，インターネットワークの入出力関係は次のように表される．

入力                  出力



$$\begin{aligned} (\mathbf{x}_1, \mathbf{y}_1) &\longrightarrow z_1 \\ (\mathbf{x}_1, \mathbf{y}_1) &\longrightarrow z_2 \\ (\mathbf{x}_2, \mathbf{y}_2) &\longrightarrow z_1 \end{aligned}$$

これから分かるように，入出力関係に一对多の対応関係が含まれてしまう．当然ながら，このような対応関係をインターネットワークによって獲得することは不可能である．上記の例は非常に稀なケースを取り上げたのではなく，モジュールの最終状態の関係に少しオーバーラップがあるだけで，一对多の対応関係を容易に作ることができる．このような一对多の対応関係を常に回避する一つの方法は，自己参照型のインターネットワークを導入することである．上記の例の場合，例えば module 3 に影響を与える クラス  $\mathcal{I}_2$  インターネットワークとして module 1 と module 3 の状態を入力するものを導入すればよい（これは p.18の Fig.2.5における  $P_3$  に相当）．次節以降で行うシミュレーション実験のなかには，記憶パターン間にかなりオーバーラップをもったものが含まれる．そこで，一对多の対応関係が生じないように，ここでは自己参照型のものだけに限定することにした．

### 2.3.2 エネルギー関数と動作方程式

まず，クラス  $\mathcal{I}_2$  と  $\mathcal{I}_M$  の2種類のインターネットワークを導入したマルチモジュール CCHN-LAGI のエネルギー関数を定義する．これは，式 (2.9) のエネルギー関数に対して

$$k = 2, M \quad q^{(2,m)} = M - 1 \quad q^{(M,m)} = 1$$

とし，インデックス  $n$  に対する注意 2.8 を考慮すれば以下のように表される．

$$\begin{aligned} E &= \sum_{m=1}^M \alpha^{(m)} E^{(m)} + \sum_{m=1}^M \sum_{\substack{n=1 \\ n \neq m}}^M \beta^{(2,n,m)} E^{(2,n,m)} + \sum_{m=1}^M \beta^{(M,1,m)} E^{(M,1,m)} \\ &= \sum_{m=1}^M \alpha^{(m)} \left[ -\frac{1}{2} \sum_{i=1}^{N^{(m)}} \sum_{j=1}^{N^{(m)}} T_{ij}^{(m)} v_i^{(m)} v_j^{(m)} + \sum_{i=1}^{N^{(m)}} \frac{1}{r_i^{(m)}} \int_0^{v_i^{(m)}} g_i^{(m)}(v) dv - \sum_{i=1}^{N^{(m)}} J_i^{(m)} v_i^{(m)} \right. \\ &\quad \left. + \frac{1}{4} \sum_{i=1}^{N^{(m)}} \sum_{j=1}^{N^{(m)}} T_{ij}^{(m)^2} \right] + \frac{1}{2} \sum_{m=1}^M \sum_{\substack{n=1 \\ n \neq m}}^M \beta^{(2,n,m)} \sum_{i=1}^{N^{(m)}} \left( v_i^{(m)} - o_i^{(2,n,m)(L+2)} \right)^2 \\ &\quad + \frac{1}{2} \sum_{m=1}^M \beta^{(M,1,m)} \sum_{i=1}^{N^{(m)}} \left( v_i^{(m)} - o_i^{(M,1,m)(L+2)} \right)^2 \end{aligned} \quad (2.23)$$

右辺第1項はモジュールの情報処理に対するエネルギー関数で，第2,3項はそれぞれクラス  $\mathcal{I}_2$  と  $\mathcal{I}_M$  の相互作用に対するエネルギー関数である．

式 (2.23) を時間微分し, 2.2.3 で述べた方法に基づいてマルチモジュール CCHN-LAGI の動作方程式を求めると以下ようになる (導出の詳細は付録 A.1 を参照) .

[モジュールの活性ダイナミクス]

$$\begin{aligned}
 c \frac{du_i^{(m)}}{dt} &= \alpha^{(m)} \cdot \left( \sum_j T_{ij}^{(m)} v_j^{(m)} - \frac{u_i^{(m)}}{r} + J_i^{(m)} \right) \\
 &\quad - \sum_{\substack{n=1 \\ n \neq m}}^M \beta^{(2,n,m)} \cdot \left( v_i^{(m)} - o_i^{(2,n,m)(L+2)} \right) \\
 &\quad + \sum_{\substack{n=1 \\ n \neq m}}^M \sum_j \left( \beta^{(2,m,n)} w_{ji}^{(2,m,n)(1)} \delta_j^{(2,m,n)(2)} + \beta^{(2,n,m)} w_{j,i+N^{(n)}}^{(2,n,m)(1)} \delta_j^{(2,n,m)(2)} \right) \\
 &\quad - \beta^{(M,1,m)} \cdot \left( v_i^{(m)} - o_i^{(M,1,m)(L+2)} \right) \\
 &\quad + \sum_{n=1}^M \beta^{(M,1,n)} \cdot \sum_j \left( w_{j,i+\varphi(m-1)}^{(M,1,n)(1)} \delta_j^{(M,1,n)(2)} \right)
 \end{aligned} \tag{2.24}$$

[モジュールの結合ダイナミクス]

$$\varepsilon \frac{dT_{ij}^{(m)}}{dt} = -T_{ij}^{(m)} + v_i^{(m)} v_j^{(m)} \tag{2.25}$$

[インターネットワークの結合ダイナミクス]

$$\gamma \frac{dw_{ij}^{(2,n,m)(l)}}{dt} = o_j^{(2,n,m)(l)} \delta_i^{(2,n,m)(l+1)} \quad (n \neq m) \tag{2.26}$$

$$\gamma \frac{dw_{ij}^{(M,1,m)(l)}}{dt} = o_j^{(M,1,m)(l)} \delta_i^{(M,1,m)(l+1)} \tag{2.27}$$

但し,

$$\delta_i^{(k,n,m)(L+2)} = h'(s_i^{(k,n,m)(L+2)}) \cdot \left( v_i^{(m)} - o_i^{(k,n,m)(L+2)} \right) \tag{2.28}$$

$$\delta_i^{(k,n,m)(l+1)} = h'(s_i^{(k,n,m)(l+1)}) \cdot \left( \sum_j w_{ji}^{(k,n,m)(l+1)} \delta_j^{(k,n,m)(l+2)} \right) \tag{2.29}$$

$$(k = 2, M; l = 1, \dots, L)$$

$$\varphi(k) = \sum_{m=1}^k N^{(m)}$$

ここで,  $c, \varepsilon, \gamma$  はそれぞれ正の時定数である .

式 (2.24) の右辺から分かるように, モジュールの活性ダイナミクス (ユニットの内部ポテンシャルに対する変更則) は大きく 5 種類の信号で構成される . この信号伝搬の様子は Fig.2.9 と Fig.2.10 に示されている .

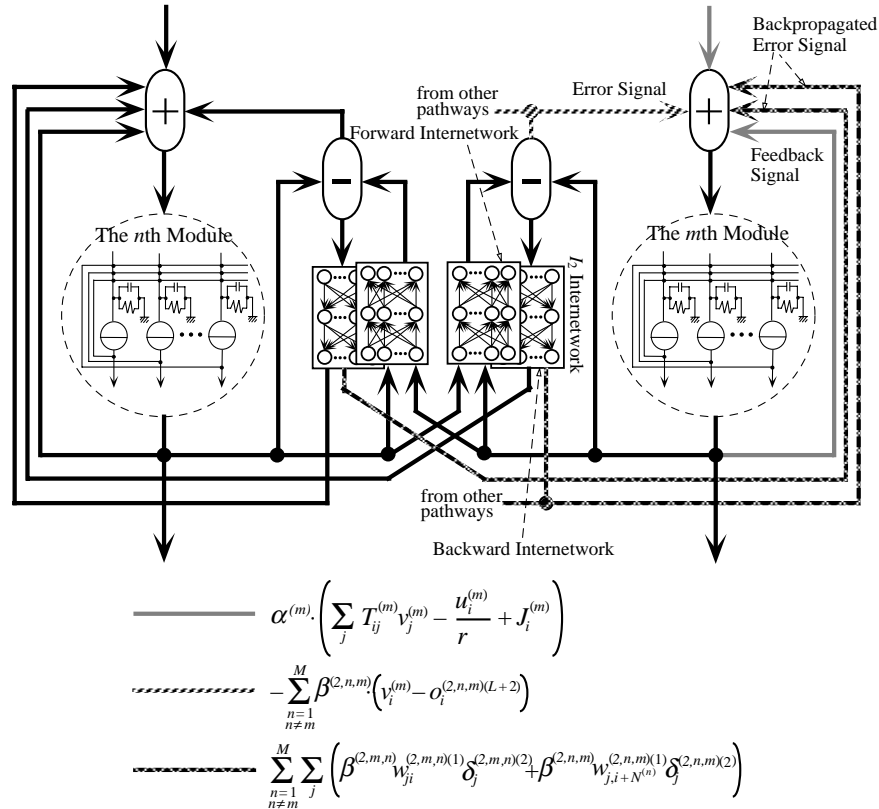


Figure 2.9: A conceptual diagram of signal flows in two-directional  $\mathcal{I}_2$  interactions between the  $n$ th and the  $m$ th modules. The correspondence between the signal flows and the terms of the activation dynamics in Eq.(2.24) is also shown.

Fig.2.9は、第  $m$  モジュールが局所的相互作用 (クラス  $\mathcal{I}_2$ ) を介して第  $n$  モジュールからの影響を受けている様子が示されている。Fig.2.9における第  $m$  モジュール上部の  $\oplus$  には5本の矢線が入っているが、このうち異なる線種は3種類であることがわかる。これらは図下に示されている信号項に相当し、式 (2.24) において最初の3つの項を表している。第1項はモジュールの内部結合  $T^{(m)}$  を介してフィードバックされる信号と外部入力であり、これはモジュールに与えられた情報処理を反映したものである。第2項および第3項はクラス  $\mathcal{I}_2$  の局所的相互作用に関する項である。第2項は第  $m$  モジュールの状態とインターネットワーク出力との誤差信号に対応し、インターネットワークの出力値がモジュールの状態に一致した時点でこの信号は消滅する。つまり、この項が存在することで、第  $m$  モジュールの状態は設計者によって与えられたモジュール間の望ましい状態関係を満足するようになる。第3項のカッコ内の2項は、第  $m$  モジュールの状態  $v^{(m)}$  を入力とする2つのインターネッ

トワークで逆方向に伝搬される誤差信号に相当する．これらは誤差逆伝搬法で結合荷重を変更するのに用いられるデルタ項と等価であり，2モジュール間の状態関係がすべて満たされたときに消滅する．

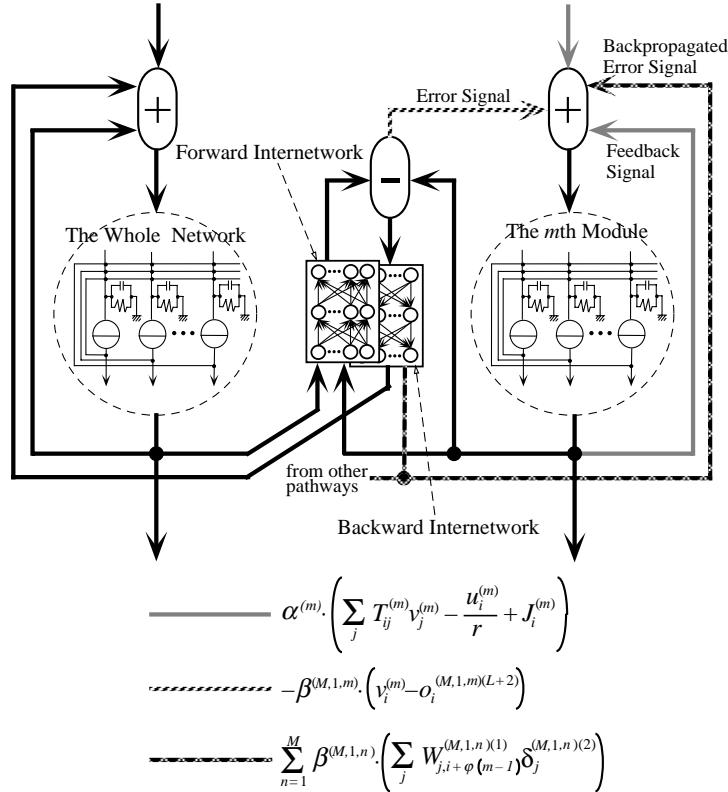


Figure 2.10: A conceptual diagram of signal flows in a  $\mathcal{I}_M$  interaction between the whole network and the  $m$ th module. The correspondence between the signal flows and the terms of the activation dynamics in Eq.(2.24) is also shown.

一方，Fig.2.10は，第  $m$  モジュールが大局的相互作用 (クラス  $\mathcal{I}_M$ ) を介してネットワーク全体 (the whole network) から影響を受けている様子が示されている．Fig.2.10における第  $m$  モジュール上部の  $\oplus$  には4本の矢線が入っているが，このうち異なる線種は3種類である．これらは図下に示されている信号項を表し，式 (2.24) の第1項および第4, 5項に対応する．第1項はモジュールの情報処理を表し，第4,5項はクラス  $\mathcal{I}_M$  の大域的相互作用に関する項である．これも誤差信号と逆誤差信号で構成され，個々のモジュールの状態とネットワーク全体で取るべき状態との間に矛盾が無くなった時点で信号が消滅する．

式 (2.25) はモジュールの結合ダイナミックス (内部結合の変更則) に対応し，状態間

の相関に基づいて結合が増強されるのがわかる。つまり、 $i$  番目のユニットと  $j$  番目のユニットが同時に活性化されるとき結合は増強され、それ以外では結合に変化は与えられない。これは、いわゆる「ヘップ則」に相当する。

式 (2.26),(2.27) は 2 種類のインターネットワークの結合ダイナミクス (外部結合の変更則) に対応し、ここでも式 (2.28),(2.29) で与えられるデルタ項が用いられている。これは微分方程式で与えられた連続時間型の BP 法と考えられる。つまり、モジュール間の写像関係を獲得する学習則として強力なアルゴリズムが得られたことになり、中間層をもつインターネットワークにより任意の非線形写像の学習が可能といえる [59]。

なお、これら活性ダイナミクスや結合ダイナミクスは、情報処理をエネルギー関数で記述したことによって自動的に求められる点に注意する。またエネルギー関数はスカラーであるため、モジュール数や結合形態などに関係なく自由に各エネルギー項を加えることができる。マルチモジュール CCHN-LAGI には 2.2 で述べた構造パラメータを変更することで様々なバリエーションが与えられる。そこで、本論文で取り上げるマルチモジュール CCHN-LAGI の分類とその略記方法について、付録 A.2 にまとめておく。

### 2.3.3 動作方程式の取扱い

2.3.2 で求められたマルチモジュール CCHN-LAGI の動作方程式は、大きく活性ダイナミクスと結合ダイナミクスの 2 つで表される。前者はネットワークの状態を更新するために用いられ、後者はモジュールやインターネットワークの結合荷重を変更するために用いられる。これらは連立微分方程式として与えられ、状態更新と学習とが同時進行するようなダイナミクスを形成する。

しかしながら、本論文では最終状態が明示的に与えられることを前提としており、結合ダイナミクスは最終状態とエネルギー関数の最小点が一致するように働けばよい。つまり状態更新と学習を同時進行させる積極的な必要性はなく、ネットワークが最終状態で安定している間に結合荷重の学習が行われるという状況を想定すればよい。つまり、式 (2.25) ~ (2.27) の解は活性ダイナミクスと切り離してオフラインで求める。このようなオフラインの学習を行うため、厳密には時定数に対して以下の仮定が必要である。

[時定数に関する仮定]

1. 活性ダイナミクスと結合ダイナミクスの時定数には、 $c \ll \varepsilon, \gamma$  の関係が

ある .

2. 時定数  $\varepsilon, \gamma$  は十分大きいものとし , 一回のパターン提示における結合荷重の修正量は十分に小さい .

1. の仮定は「結合荷重の変更スピードは , ユニットの状態が変更されるスピードに比べて十分遅い」ということを意味しており , この仮定のもとでは持続的に安定化された状態でしか学習が進まないことになる . なお , このような仮定は人工ニューラルネットの学習でよく使われ , 生物学的にみても不自然なものではない . また , 2. の仮定は「 $P$  個の最終状態  $\xi^\mu$  ( $\mu = 1, \dots, P$ ) が順に与えられるとき , それらパターンを提示するスピードは結合荷重が変更されるスピードよりも十分に速く , パターン提示に対するダイナミックは無視できる .」ということを表している . このとき , パターンを一括して提示した後に結合荷重の学習を行ういわゆるバッチモードを結合ダイナミクスに導入し , 式 (2.25), (2.26) は以下のように書換えられる .

[モジュールの結合ダイナミクス]

$$\varepsilon \frac{dT_{ij}^{(m)}}{dt} = -T_{ij}^{(m)} + \sum_{\mu=1}^P \xi_i^{\mu(m)} \xi_j^{\mu(m)'} \quad (2.30)$$

[クラス  $\mathcal{I}_2$  インターネットワークの結合ダイナミクス]

$$\gamma \frac{dw_{ij}^{(2,n,m)(l)}}{dt} = \sum_{\mu=1}^P o_j^{\mu(2,n,m)(l)} \delta_i^{\mu(2,n,m)(l+1)} \quad (n \neq m; \quad l = 1, \dots, L+1) \quad (2.31)$$

但し ,

(インターネットワーク入力)

$$o_i^{\mu(2,n,m)(1)} = \begin{cases} \xi_i^{\mu(n)} & (i \leq N^{(n)}) \\ \xi_{i-N^{(n)}}^{\mu(m)} & (N^{(n)} < i \leq N^{(n)} + N^{(m)}) \end{cases} \quad (2.32)$$

(インターネットワーク出力)

$$o_i^{\mu(2,n,m)(l+1)} = h(s_i^{\mu(2,n,m)(l+1)}) \quad (2.33)$$

$$s_i^{\mu(2,n,m)(l+1)} = \sum_{j=1}^{H^{(l)}} w_{ij}^{(2,n,m)(l)} o_j^{\mu(2,n,m)(l)} \quad (2.34)$$

(逆誤差伝搬項)

$$\delta_i^{\mu(2,n,m)(L+2)} = h'(s_i^{\mu(2,n,m)(L+2)}) \cdot (\xi_i^{\mu(m)} - o_i^{\mu(2,n,m)(L+2)}) \quad (2.35)$$

$$\delta_i^{\mu(2,n,m)(l+1)} = h'(s_i^{\mu(2,n,m)(l+1)}) \cdot \left( \sum_j w_{ji}^{(2,n,m)(l+1)} \delta_j^{\mu(2,n,m)(l+2)} \right) \quad (2.36)$$

[クラス  $\mathcal{I}_M$  インターネットワークの結合ダイナミクス]

$$\gamma \frac{dw_{ij}^{(M,1,m)(l)}}{dt} = \sum_{\mu=1}^P o_j^{\mu(M,1,m)(l)} \delta_i^{\mu(M,1,m)(l+1)} \quad (l = 1, \dots, L+1) \quad (2.37)$$

但し,

(インターネットワーク入力)

$$o_i^{\mu(M,1,m)(1)} = \begin{cases} \xi_i^{\mu(1)} & (i \leq \varphi(1)) \\ \xi_{i-\varphi(1)}^{\mu(2)} & (\varphi(1) < i \leq \varphi(2)) \\ \vdots & \\ \xi_{i-\varphi(M-1)}^{\mu(M)} & (\varphi(M-1) < i \leq \varphi(M)). \end{cases} \quad (2.38)$$

(インターネットワーク出力)

$$o_i^{\mu(M,1,m)(l+1)} = h(s_i^{\mu(M,1,m)(l+1)}) \quad (2.39)$$

$$s_i^{\mu(M,1,m)(l+1)} = \sum_{j=1}^{H^{(l)}} w_{ij}^{(M,1,m)(l)} o_j^{\mu(M,1,m)(l)} \quad (2.40)$$

(逆誤差伝搬項)

$$\delta_i^{\mu(M,1,m)(L+2)} = h'(s_i^{\mu(M,1,m)(L+2)}) \cdot (\xi_i^{\mu(m)} - o_i^{\mu(M,1,m)(L+2)}) \quad (2.41)$$

$$\delta_i^{\mu(M,1,m)(l+1)} = h'(s_i^{\mu(M,1,m)(l+1)}) \cdot \left( \sum_j w_{ji}^{(M,1,m)(l+1)} \delta_j^{\mu(M,1,m)(l+2)} \right) \quad (2.42)$$

$$\varphi(k) = \sum_{m=1}^k N^{(a_m)}$$

ここで,  $o_i^{(k,n,m)(l)}$ ,  $s_i^{(k,n,m)(l)}$ ,  $\delta_i^{(k,n,m)(l)}$  の右肩にある  $\mu$  は,  $\mu$  番目の入力パターンを提示したときのインターネットワーク出力, 内部ポテンシャル, 誤差逆伝搬項を表すためのインデックスである.

さて, このような結合ダイナミクスのもとで, 実際にモジュールとインターネットワークの結合荷重を学習する方法について簡単に述べる. 式 (2.30) の解は自明であり, 実際には以下の相関行列で与える.

$$T_{ij}^{(m)} = \sum_{\mu=1}^P \xi_i^{\mu(m)} \xi_j^{\mu(m)} \quad (2.43)$$

また, インターネットワークの結合ダイナミクスは, 差分化すれば誤差逆伝搬法を用いたバッチ学習と等価になる.

よって次章以降のシミュレーション実験では, 結合荷重の学習に結合ダイナミクスをそのまま使うのではなく, 以下のような学習則で代用する.

モジュール： 相関学習

インターネットワーク： 逆誤差伝搬法 (バッチモード)

なお，求められた結合荷重を式 (2.24) の活性ダイナミクスに代入し，4 次のルンゲクッタ法を用いてユニットの状態更新が行われる．このときの時定数  $c$  はすべて 1.0 とする．

注意 2.9 3.3で取り扱う CCHN-OPT には，上記の学習則は適用されない．

## 2.4 まとめ

複数個の動的な情報処理系で構成されたニューラルネットモデルとして，本論文ではマルチモジュール CCHN (multi-module Cross-Coupled Hopfield Nets) を提案した．このモデルにおいては，個々のモジュールだけでなくネットワーク全体に対しても漸近安定性が仮定される．このような仮定は表現できる動特性をかなり制限してしまうことになるが，従来の動的マルチモジュールニューラルネットにはみられなかった以下のような特徴をもっている．

1. ネットワーク構造や各部位の情報処理様式などに依存しないモデル化が可能となる．これによって，任意個のモジュールで構成されたネットワークや様々な相互作用が導入されたものが簡単に構築できる．
2. ネットワーク構造に関係なく，動作方程式を試行錯誤なしで導出できる．
3. ネットワークの動特性がエネルギー関数の幾何学的特性として表されることになり，動特性の考察が容易になる．

2.2では，本モデルにおけるモジュールの情報処理と相互作用の機能について定義し，その機能を実現するためのエネルギー関数について述べた．また，相互作用を介してモジュールに与えられる影響が，ネットワーク全体のどの程度グローバルな情報に基づいたものかを表すために相互作用のクラスを定義した．様々なクラスの相互作用を導入することで，マルチモジュール CCHN が表現できるバリエーションは格段に増える．さらに，マルチモジュール CCHN の動作方程式を導出する方法についても述べた．この動作方程式は，如何なる構造をもったマルチモジュール CCHN に対しても漸近安定性を保証する．



マルチモジュール CCHN が表現できるネットワーク構造のバリエーションが膨大であることから、動作方程式の一般形を具体的な式で記述することは難しい。そこで 2.3 では、導入する相互作用の種類を 2 種類に限定したマルチモジュール CCHN-LAGI をモデル化した。ここで導入された 2 種類の相互作用はクラス  $\mathcal{I}_2, \mathcal{I}_M$  のインターネットワークで生成され、それぞれ局所的相互作用と大局的相互作用と呼ばれる。局所的相互作用を生成するインターネットワークでは、相互作用の影響を受けるモジュールの状態と他に 1 つのモジュールの状態とが入力として参照される。また、大局的相互作用ではネットワーク全体の状態が入力として参照される。このように相互作用の影響を受けるモジュールの状態が自己参照されることで、インターネットワークが獲得しなければならない状態関係 (ローカル・コンテキスト) に一対多のものが含まれなくなる。さらに、マルチモジュール CCHN-LAGI に対して動作方程式の具体的な形が求められた。ここで求められた活性ダイナミクスから、モジュール間相互作用がどのような信号で構成されるかが明確になった。



# 第3章 誤差エネルギーをもつホップフィールド ドモデル — 1 モジュールモデル

## 3.1 まえがき

第2章では、クロス結合ホップフィールドネットをマルチモジュールモデルとして拡張し、エネルギー関数から動作方程式を導く方法について述べた。このマルチモジュール CCHN では、モジュール数やモジュール間結合の形態など様々な構造をもったネットワークを表現でき、その結果として、多様な情報処理を実現できる可能性をもっている。

本章以降では、連想記憶への適用を通して、マルチモジュール CCHN における情報処理の特徴および有効性を議論していく。連想記憶で記憶されるパターンの性質を考慮し、ネットワークの最終状態に以下の仮定を行う。

1. モジュールの最終状態は設計者によってあらかじめ明示的に与えられる。
2. ネットワーク全体の最終状態には、望ましい対応関係 (グローバル・コンテキスト) が存在する。

1. の仮定から、マルチモジュール CCHN を構成しているモジュールが、例えば巡回セールスマン問題のように最終状態が明示的に与えられない問題を解くケースは想定されない。また 2. の仮定から、モジュールの最終状態は単にローカル・コンテキストを満足するだけでなく、グローバル・コンテキストをも満足する必要がある。

注意 3.1 グローバル・コンテキストはいつでも明示的に与えられるのではない。つまり、各モジュールがローカル・コンテキストを満足した結果として、グローバル・コンテキストが生じると考えるのが一般的である。しかしながら、本章以降で取り扱う連想記憶の場合、記憶パターンを各モジュールに分割して記憶するため、ネットワーク全体で想起されるべき最終状態 (グローバル・コンテキスト) はあらかじめわかっている。そこで、ここでは上記 2. の仮定を行った。

前述したように，マルチモジュール CCHN では最終状態にどのような意味付けを行うかで，その情報処理機能が決定まる．このことは，本論文における議論を連想記憶に絞ったとしても，最終状態の意味付けを変更することで一般的な議論が行えることを意味している．

本章以降では，マルチモジュール CCHN モデルの動特性や有効性について議論していくが，その第一ステップとしてホップフィールドネットのエネルギー関数に誤差エネルギー（相互作用に対するエネルギー関数）を加えたニューラルネットを取り上げる．これは，1つのモジュールが1つのインターネットワークにより自己結合されたものに相当し，ここでは「1モジュールモデル」と呼ぶことにする．モジュールは従来型のホップフィールドネットであるので，上記のようなニューラルネットの動特性を調べることで，相互作用の役割について議論することができる．モジュール間相互作用に関する考察を容易にするため，インターネットワークには中間層がない2層の階層型ニューラルネットを採用する．このとき，上記の1モジュールモデルは自己想起型連想記憶モデルと構造的に等価なネットワークに置き換えられることを示す．つまり，モジュールのユニット数が  $N$  である1モジュールモデルの情報処理は， $N \times N$  の結合行列をもつ自己想起型連想記憶（ホップフィールドネット）に置き換えて議論することができる．これによって，これまで自己想起型連想記憶モデルで適用されてきた解析手法を1モジュールモデルに適用することが可能になる．

ニューラルネットにおける連想記憶モデルの動特性や記憶容量に関する解析は，スピングラスの理論 [35] や  $S/N$  解析 [36][37] などを用いた統計的手法と結合行列の固有値を調べる幾何的方法 [38][40] に大別される．前者では，結合荷重を介してフィードバックされた信号のうち，想起すべき記憶パターンと同成分をもつ信号とそれ以外のノイズ信号（クロストーク）を分離し，それらの統計的性質を調べることで記憶容量が計算される．後者では，記憶パターンが張る状態空間（パターン空間）と，その直交補空間であるノイズ空間に分け，それぞれにおけるネットワークの動特性を幾何学的に考察する．前者の手法では，記憶容量や動特性に関して厳密な考察が行えるが，記憶パターンの確率分布やユニット数，記憶パターン数に対してあまり現実的とはいえない条件を仮定する必要がある．また計算の取り扱いが容易でなく，相互作用を生成するためのインターネットワークをもつ1モジュールモデルでは解析がかなり複雑になると考えられる．そこで本章では，後者の幾何的手法を用いて考察を行うことにする．

従来の自己想起型連想記憶モデルは，結合行列に記憶パターンの相関行列を用いるもの（相関モデル）と一般化逆行列を導入したもの（最適連想写像モデル）に大別され

る．本章では，モジュールの結合行列として上記 2 つの場合を取り上げ，1 モジュールモデルの動特性を考察する．また，従来型の相関モデルや最適連想写像モデルとの比較を通して，1 モジュールモデルにおける相互作用の役割およびその効果を明らかにする．

## 3.2 1 モジュールモデルから導出される自己想起型連想記憶 (1) — 相関モデル

### 3.2.1 1 モジュールモデル

本章で取り上げる 1 モジュールモデルは，2.2 のマルチモジュール CCHN においてモジュール数を  $1 (M = 1)$  とし，クラス  $\mathcal{I}_M$  の大局的相互作用を 2 層インターネットワーク (つまり  $L = 0$ ) で実現したものに相当する．このモデルの概念図を Fig.3.1 に示す．ここで，インターネットワーク内の  $\square$  はユニットを表し，その間の矢印は結合

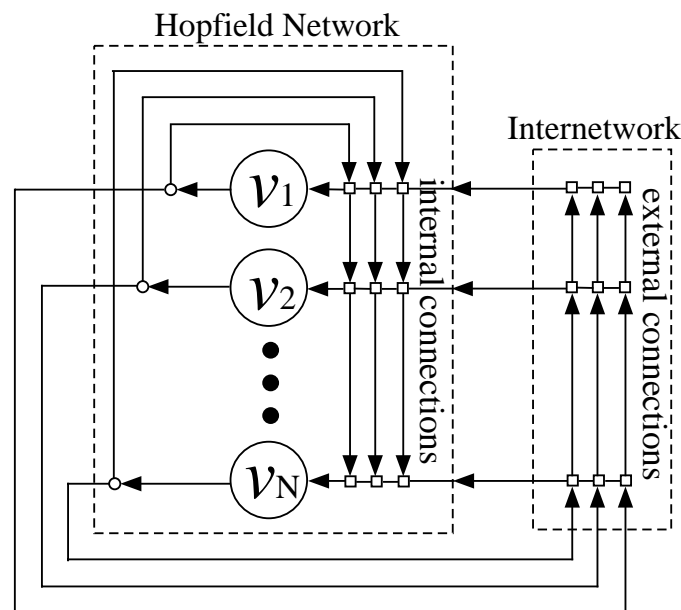


Figure 3.1: A conceptual diagram of 1-module model.

を表す．また，ホップフィールドネット内にある  $\square$  はユニット間結合を表している．なお以下では，連想記憶モデルでは一般に使われないモジュールの外部入力 は考慮しないことにする (つまり  $J = 0$ )．また連想記憶の場合，ユニット出力が連続量で表

されていたとしても，それに最も近い2値情報 ( $\{0,1\}$  または  $\{-1,1\}$ ) に置き換えて考えることが多い．よって，ユニット出力の特性を決定する定数 (後で示す式 (3.6) における  $k$ ) の値を大きくとることが通常である．このとき，モジュールのエネルギー関数におけるエネルギー損失の項はほぼ定数となり無視できる．

そこで，式 (2.23) に対して

$$M = 1, \quad N^{(m)} = N, \quad J_i^{(m)} = 0, \quad \beta^{(2,n,m)} = 0$$

とおけば，1 モジュールモデルのエネルギー関数は式 (2.23) から以下のような行列表現で表される．

$$E = -\frac{\alpha}{2} \mathbf{v}' \mathbf{T} \mathbf{v} + \frac{\beta}{2} (\mathbf{v} - \mathbf{w} \mathbf{v})' (\mathbf{v} - \mathbf{w} \mathbf{v}) + \frac{1}{4} \text{tr}(\mathbf{T} \mathbf{T}') \quad (3.1)$$

但し， $\mathbf{v} = (v_1 \cdots v_N)'$  はモジュールの状態ベクトルであり， $N$  はユニット数である． $'$  は行列またはベクトルの転置を表す．モジュール番号や相互作用の種類を表す上付き添字は，表記の簡単のため省略している．また， $\mathbf{T} = \{T_{ij}\}$ ， $\mathbf{w} = \{w_{ij}\}$  は，それぞれモジュールとインターネットワークの結合行列であり，どちらも  $N \times N$  の正方行列になる．

注意 3.2 2.3の定義に基づくと， $\mathbf{v} = \mathbf{v}^{(1)}$ ， $\mathbf{T} = \mathbf{T}^{(1)}$ ， $\mathbf{w} = \mathbf{w}^{(1,1,1)}$ ， $\alpha = \alpha^{(1)}$ ， $\beta = \beta^{(1,1,1)}$  と対応付けられる．

1 モジュールモデルの動作方程式を求めるため，まず式 (3.1) の時間微分を求める．

$$\begin{aligned} \frac{dE}{dt} &= -\alpha \left( \frac{d\mathbf{v}'}{dt} \mathbf{T} \mathbf{v} + \frac{1}{2} \mathbf{v}' \frac{d\mathbf{T}}{dt} \mathbf{v} \right) + \beta \left[ \frac{d\mathbf{v}'}{dt} (\mathbf{I} - \mathbf{w})' (\mathbf{I} - \mathbf{w}) \mathbf{v} - \mathbf{v}' \frac{d\mathbf{w}'}{dt} (\mathbf{I} - \mathbf{w}) \mathbf{v} \right] \\ &\quad + \frac{1}{4} \text{tr} \left( \frac{d\mathbf{T}}{dt} \mathbf{T}' + \mathbf{T} \frac{d\mathbf{T}'}{dt} \right) \\ &= \frac{d\mathbf{v}'}{dt} \left[ -\alpha \mathbf{T} + \beta (\mathbf{I} - \mathbf{w})' (\mathbf{I} - \mathbf{w}) \right] \mathbf{v} - \frac{\alpha}{2} \mathbf{v}' \frac{d\mathbf{T}}{dt} \mathbf{v} + \frac{1}{2} \text{tr} \left( \frac{d\mathbf{T}}{dt} \mathbf{T}' \right) \\ &\quad - \beta \mathbf{v}' \frac{d\mathbf{w}'}{dt} (\mathbf{I} - \mathbf{w}) \mathbf{v} \end{aligned} \quad (3.2)$$

ここで， $\mathbf{I}$  は  $N \times N$  の単位行列を表す．なお， $\mathbf{T}$  は対称行列となることに注意する．これから動作方程式は次のように決めればよい．

[モジュールの活性ダイナミクス]

$$c \frac{d\mathbf{u}}{dt} = \{ \alpha \mathbf{T} - \beta (\mathbf{I} - \mathbf{w})' (\mathbf{I} - \mathbf{w}) \} \mathbf{v} \quad (3.3)$$

[モジュールの結合ダイナミクス]

$$\varepsilon \frac{d\mathbf{T}}{dt} = -\mathbf{T} + \mathbf{v}\mathbf{v}' \quad (3.4)$$

[インターネットワークの結合ダイナミクス]

$$\gamma \frac{d\mathbf{w}}{dt} = (\mathbf{I} - \mathbf{w})\mathbf{v}\mathbf{v}' \quad (3.5)$$

但し,

$$\mathbf{v} = \mathbf{g}(k\mathbf{u}) \quad (3.6)$$

ここで,  $c, \varepsilon, \gamma$  は正の時定数,  $k$  は  $\mathbf{g}(\cdot)$  の特性を決定する正の定数である.

### 3.2.2 自己想起型連想記憶モデルの導出

式 (3.3) ~ (3.6) で与えられる 1 モジュールモデルは, 自己想起型連想記憶と構造的に等価なネットワークで表される. これを以下で導く.

ネットワークの情報処理様式は結合荷重によって決まる. そこで, ネットワークの状態が記憶パターンに等しいとき, 式 (3.1) のエネルギー関数が最小値をとるよう結合荷重  $\mathbf{T}$  および  $\mathbf{w}$  を構成する必要がある. これには式 (3.4), (3.5) を使えばよいが, 式 (3.3) の活性ダイナミクスとの連立微分方程式で表されており, その解を求めることは容易ではない. そこで, p.36 で述べた時定数に対する仮定を採用する. つまり, 結合が変化するスピードがユニットの状態変化のスピードに比べて十分に遅いことを仮定する. これによって, 式 (3.3) ~ (3.5) の連立微分方程式の解は, 式 (3.4), (3.5) の解を独立に求め, それらを式 (3.3) に代入することで近似的に与えられる.

$\mu$  番目の最終状態 (記憶パターン) を  $\xi^\mu$  とすれば, 式 (2.30), (2.37) ~ (2.42) に基づいて, 式 (3.4), (3.5) の結合ダイナミクスは以下のように表される.

[モジュールの結合ダイナミクス]

$$\varepsilon \frac{d\mathbf{T}}{dt} = -\mathbf{T} + \sum_{\mu=1}^P \xi^\mu \xi^{\mu'} \quad (3.7)$$

[インターネットワークの結合ダイナミクス]

$$\gamma \frac{d\mathbf{w}}{dt} = \sum_{\mu=1}^P (\mathbf{I} - \mathbf{w}) \xi^\mu \xi^{\mu'} \quad (3.8)$$

ここで,  $P$  は記憶パターンの個数である. これらは式 (3.4),(3.5) の右辺に  $v = \xi^\mu$  を代入し,  $\mu$  に関して加算することで得られる. つまり, 記憶パターンを提示するダイナミクスを無視し, バッチモードでモジュールおよびインターネットワークの結合荷重を求めることになる. ここで式 (3.8) の右辺が  $\mu$  に関する単純な加算で表されるのは, 線形なインターネットワークを用いているためである.

式 (3.7),(3.8) は線形の微分方程式であり, 解は容易に求められる (詳細は付録 A.3を参照). 以下にその解を示す.

$$\mathbf{T}(t) = \mathbf{T}(0) \exp\left(-\frac{t}{\varepsilon}\right) + \left(1 - \exp\left(-\frac{t}{\varepsilon}\right)\right) \mathbf{S} \quad (3.9)$$

$$\mathbf{w}(t) = \mathbf{w}(0) \exp\left(-\frac{\mathbf{S}}{\gamma} t\right) + \mathbf{Q} \mathbf{B}(t) \mathbf{Q}^{-1} \quad (3.10)$$

但し,

$$\mathbf{S} = \sum_{\mu=1}^P \xi^\mu \xi^{\mu T} = \mathbf{Q} \mathbf{A} \mathbf{Q}^{-1} \quad (3.11)$$

$$\mathbf{A} = \begin{pmatrix} \lambda_1 & & & & & \\ & \ddots & & & & \\ & & \lambda_\kappa & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix} \quad (3.12)$$

$$\mathbf{B}(t) = \begin{pmatrix} 1 - \exp\left(-\frac{\lambda_1}{\gamma} t\right) & & & & & \\ & \ddots & & & & \\ & & 1 - \exp\left(-\frac{\lambda_\kappa}{\gamma} t\right) & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix} \quad (3.13)$$

$\mathbf{S}$  は記憶パターンの相関行列に相当し, 実対称行列となることから式 (3.11) のように対角化できる. ここで  $\mathbf{Q}$  は固有ベクトルを並べた行列であり, 直交行列となる. 相関行列の性質から,  $\mathbf{S}$  のランクを  $\kappa (\leq P)$  とすれば, 固有値  $\lambda_i$  は  $\kappa$  個の正の固有値と  $(N - \kappa)$  個の零固有値で与えられる. また, 固有ベクトルは互いに直交し,  $\mathbf{Q}$  は直交行列で与えられる.

ここで,  $\mathbf{T}, \mathbf{w}$  を式 (3.3) に代入することで, 活性ダイナミクスに対する次の近似を得る.

$$c \frac{d\mathbf{u}}{dt} = \mathbf{Q} \mathbf{C} \mathbf{Q}^{-1} \mathbf{v} \quad (3.14)$$

$$\mathbf{v} = g(k\mathbf{u}) \quad (3.15)$$





1. 式 (3.4),(3.5) の結合ダイナミクスに従って  $T$  と  $w$  を求め, これを式 (3.3) 右辺の  $\{\alpha T - \beta(I - w)'(I - w)\}$  に代入することで結合行列を構成する.
2. まず相関行列の固有値と固有ベクトルを求める. これから式 (3.16) によって  $C$  を計算し,  $QCQ^{-1}$  によって結合行列を構成する.

上記に手順によって構成された結合行列は式 (3.3) または式 (3.14) に代入され, 想起シミュレーションが実行される. このときのネットワークの初期状態ベクトルも, 要素が  $\{-1, 1\}$  で表されたバイポーラ表現をとる.

パターンの想起過程は記憶パターン  $\xi^\mu$  と時刻  $t$  における状態ベクトル  $v(t)$  との方向余弦  $d^\mu(t)$  をモニタすることで調べられ,  $d^\mu(t)$  は以下の式で与えられる.

$$d^\mu(t) = \frac{1}{N} \xi^{\mu'} v(t) \quad (3.17)$$

この定義から明らかなように,  $d^\mu(t) \rightarrow 1$  となるにつれてネットワークの状態は  $\mu$  番目の記憶パターンに近づくことになる. そこで, 記憶パターンの想起は次のように定義される.

**定義 3.1 (記憶パターンの想起)** 十分長い時間  $\tilde{t}$  に対し, ネットワークの状態ベクトル  $v(\tilde{t})$  と記憶パターン  $\xi^\mu$  との方向余弦が  $d^\mu(\tilde{t}) = 1$  となったとき, ネットワークは「 $\mu$  番目の記憶パターンを想起した」という.

また式 (3.17) の方向余弦を用いて, パターン想起の難しさを次のように定義する.

**定義 3.2 (パターン想起の難しさ)**  $\mu$  番目のパターン想起に対する難しさは, 次式の記憶パターン  $\xi^\mu$  と初期状態ベクトル  $v(0)$  との方向余弦で表される.

$$d^\mu(0) = \frac{1}{N} \xi^{\mu'} v(0) \quad (3.18)$$

このとき,  $d^\mu(0)$  が小さいほど初期状態が記憶パターンから離れていることを表し, 「パターン想起は難しい」という.

パターン想起の難しさはネットワークの初期状態だけでなく, 記憶パターン数にも依存する. 本論文では, 記憶パターン数は記憶パターンベクトルの次元数との比で表され, これを以下に示す.

**定義 3.3 (記憶率)** 記憶パターンベクトルの次元  $N$  に対する記憶パターン数  $P$  の比を「記憶率  $r$ 」と呼び, 次式で表す.

$$r = \frac{P}{N} \quad (3.19)$$

一般に， $r$  が大きいほどパターンの想起が難しくなる．

連想記憶モデルの性能は記憶容量や引き込み領域の大きさを評価される．しかしながら，厳密な意味での引き込み領域を評価することは難しく，以下では，有限個の初期状態から想起をスタートさせ，その成功回数の割合に基づいた引き込み領域の評価を行う．そこで，引き込み領域およびその評価基準を以下のように定義する．

定義 3.4 (引き込み領域と限界方向余弦) ある連想記憶モデルに対し，一定確率  $\rho$  で想起が成功する初期状態ベクトルの集合を「引き込み領域」と定義する．また，引き込み領域内にある初期状態ベクトルに対し，それら方向余弦の最小値を「限界方向余弦 (critical direction cosine)」と呼ぶ．

### 3.2.4 シミュレーション実験および考察

前述したように，1 モジュールモデルの動特性は 3.2.2 で導出した自己想起型連想記憶モデル (CCHN-COR) の結合行列の特性として表される．そこで CCHN-COR を連想記憶に適用し，その連想性能や連想特性を考察することで，1 モジュールモデルの動特性について調べる．これによって，相互作用がモジュールにどのような影響を与えるかについて議論できる．

まず，ランダムパターンに対する CCHN-COR の連想性能をシミュレーション実験により調べ，従来モデルとの比較を行う．さらに CCHN-COR と従来モデルとの動特性の違いを，固有値解析を通して調べる．

#### (a) ランダムパターンに対する連想性能

CCHN-COR は，結合荷重を相関行列  $S$  とした自己想起型連想記憶 (ホップフィールドネット) に 2 層のインターネットワークを導入したモデルである．よって，相互作用が導入されることの効果を調べるには，相関モデルとの比較を通して行うのが適当である．しかしながら，相関モデルでは記憶パターンが互いに直交していないと，パターン間の干渉 (クロストーク) によって著しく連想特性が劣化することが知られている [60]．ランダムな記憶パターンの場合，次元数  $N$  が十分大きければほぼ直交させることが可能であるが，以下のシミュレーション実験では  $N$  を数百オーダとしており，完全に相関モデルにおいてクロストークの影響を排除することは難しい．そこで，記憶パターンが互いに一次独立であればクロストークの影響を完全に排除できる最適連想写像モデル [12][61] についても性能を評価し，本モデルとの比較を行うことにする．

最適連想写像モデルの結合行列  $D$  は次のように与えられる .

$$D = X X^+ \quad (3.20)$$

但し ,

$$X = (\xi^1, \xi^2, \dots, \xi^P) \quad (3.21)$$

$$X^+ = (X' X)^{-1} X' \quad (3.22)$$

ここで ,  $X$  は一次独立な記憶パターンベクトルを横に並べた  $(N \times P)$  行列である .  $X^+$  は  $X$  の Moore-Penrose 一般化逆行列であり , 最適写像モデルは一般化逆行列モデルともいわれる . 以下では , 結合行列が式 (3.11) の  $S$  で表される相関モデルを HN-COR とし , 式 (3.20) の  $D$  を結合行列とする最適連想写像モデルを HN-OPT とする (モデルの表記方法については付録 A.2を参照) .

記憶パターンには , 各ビットが  $1/2$  の確率で  $\pm 1$  の値をとる 200 次元のランダムベクトルを用いた . この場合 , 記憶パターンはほぼ一次独立であると考えてよく , 相関行列  $S$  のランクはパターン数  $P$  に等しくなる . なお , ネットワークのユニット数は記憶パターンの次元数  $N$  と同じ 200 である . 想起シミュレーションで用いられる主なパラメータを Table 3.1にまとめておく .

Table 3.1: Parameter values

number of modules ( $M$ )	1
dimension of memory pattern vectors ( $N$ )	200
number of memory patterns ( $P$ )	10, 20, 30, 40, 50 60, 70, 80, 90, 100
number of hidden layers in internetworks ( $L$ )	0
class of interactions	$\mathcal{I}_M$
contribution of module ( $\alpha$ )	1.0
contribution of interaction ( $\beta$ )	0.3

シミュレーション結果を Fig.3.2に示す . ここで , 横軸の記憶率  $r$  は p.48の定義から  $0.1 \sim 1.0$  となる . また , 限界方向余弦  $d_c$  は定義 3.4において  $\rho = 0.98$  としたものをを用いている . なお , CCHN-COR, HN-COR, HN-OPT とともに自己想起型連想記憶であるので , 方向余弦  $d^\mu(t)$  は次のように求められる .

$$d^\mu(t) = \frac{1}{N} \xi^{\mu'} v(t) \quad (3.23)$$

一般に、 $d_c$  が小さいほど平均的な引き込み領域が大きいといえる。

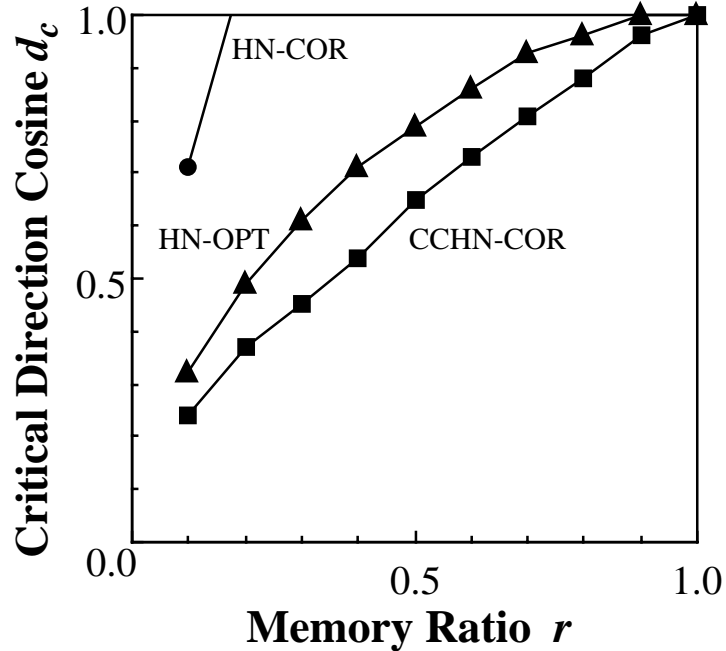


Figure 3.2: Comparison of association performance

Fig.3.2から分かるように、HN-CORは $r > 0.1$ で引き込み領域を失っている。これは、相関モデルの記憶容量や想起能力に関するこれまでの結果と矛盾しない。これに対し、HN-OPTとCCHN-CORでは、 $r$ が大きくなると引き込み領域は小さくなるものの、 $r < 1.0$ であれば引き込み領域が存在し、連想記憶装置として十分機能している。また、CCHN-CORの性能は全般的にHN-OPTより優れていることがわかる。CCHN-CORはモジュールに相関モデルを用いたものであるため、モジュールの情報処理だけでは殆んど引き込み領域をもたない。つまり、この結果は、モジュールに相互作用を導入することが飛躍的な性能向上につながることを示唆している。

式(3.12)と式(3.16)を比較すると明らかなように、対角行列 $A$ における最後の $N - \kappa$ 個の零固有値が $C$ では負の定数になっている。つまり、HN-CORとCCHN-CORの動的性質の違いは、これら $N - \kappa$ 個の固有値の違いに起因している。このような相違点が連想性能を大きく変えるのはなぜだろうか。これについて、次節で議論を試みる。

## (b) 連想特性の考察

式 (3.11) を結合行列とする相関モデル (HN-COR) の場合、結合行列の固有値は正とゼロのものに大きく分けられる。このことは、HN-COR の動特性が大きく2つの部分状態空間において特徴付けられることを表している。正の固有値で表される空間は記憶パターンによって張られる空間であり、これを以下では「パターン空間」と呼ぶことにする。それに対し、零固有値で表される空間はパターン空間と直交した空間であり「ノイズ空間」と呼ばれる。CCHN-COR の結合行列は式 (3.14) からわかるように、相関モデルと同じ固有ベクトルをもっている。つまり、CCHN-COR の動特性は HN-COR と同じ固有空間において議論できることを意味する。式 (3.16) から、CCHN-COR のパターン空間に対する固有値は HN-COR と同じであるが、ノイズ空間に対する固有値が負の定数となっている。つまり HN-COR と CCHN-COR における動特性の差異はノイズ空間におけるダイナミクスに現れるはずである。そこで、これら2つのモデルにおける連想特性をパターン空間とノイズ空間に分けて考察する。

ホップフィールドネットを用いた自己想起型連想記憶モデルの場合、情報処理は大きく以下の2つに分けられる。

1. 結合行列を介した線形変換
2. ユニットの出力関数を介した非線形変換

金道らは、HN-COR の状態変化は主に前者の線形変換に依存するが、これだけでは状態が記憶パターン上で安定化されないことを示した [40][62]。これは、HN-COR において信号が繰返しフィードバックされると、線形変換は最大固有値をもつ固有ベクトルの方向に状態成分を増加させる性質をもっているからである。但し、これはユニット出力での非線形変換を無視した場合であって、実際には、非線形変換による状態の引き込み (量子化) が伴う。この引き込まれた状態が記憶パターンに等しければ、正しく想起されたことになる。しかし、記憶パターン数が多くなると、一般に結合行列の最大固有値が他の固有値に比べ相対的に大きくなり、記憶パターン付近での線形変換による不安定化作用が非線形変換による安定化作用に比べて強まる傾向をもつ。このため、HN-COR の連想性能はパターン数の増加とともに悪化する [40]。

また HN-COR では、ノイズ空間に属する状態成分が線形変換にまったく影響を受けない。つまり、想起の開始時点においてネットワークの状態がノイズ空間に成分をもつとき、HN-COR の線形変換によるダイナミクスだけではこの成分を取り除く

ことはできない．このことは，たとえ記憶パターン数が少なくても，HN-COR が記憶パターン付近であまり大きな引き込み領域をもてない一因となっている．このため，相関行列の対角要素を零にしたものが結合行列としてよく使われる．この場合，固有値の和が零となることから，ノイズ空間の固有値は負の定数となる．つまり，ノイズ空間成分を減少させるようなダイナミクスが発生する．しかし，その固有値の絶対値はパターン空間の最大固有値より大きくならないため，パターン空間におけるダイナミクスが主流であることには変わりはない．ここでは，CCHN におけるモジュール内の学習則は通常 of 相関行列を導くものであるため，対角要素を零とする相関行列は扱わないことにする．

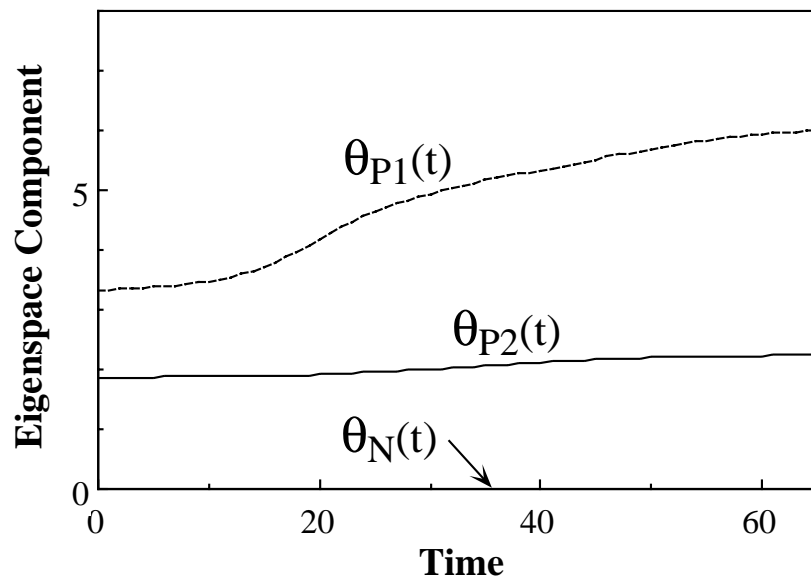
一方，式 (3.16) からわかるように，CCHN-COR の結合行列ではノイズ空間の固有値を定数  $\beta$  によって任意に設定できる．いま，この  $\beta$  の絶対値がパターン空間における最大固有値より十分大きいと仮定する．状態のノイズ空間成分が大きい場合，式 (3.14) のダイナミクスで主流となるのは，明らかに状態のノイズ空間成分を減少させるダイナミクスである．但し，この空間成分の減少に伴ってノイズ空間における状態変化量も減少するため，やがてダイナミクスの主流はパターン空間へ移っていく．この結果として，ネットワークの状態はある記憶パターンで安定化されることになる．また，ノイズ空間成分が大きい間は，パターン空間における状態変化量が相対的に小さくなるため，相関行列の場合に問題となった，過度に最大固有値の空間成分が増大される現象は起こりにくくなる．つまり，ノイズ空間における固有値の絶対値がパターン空間の最大固有値より十分大きいことが偽記憶の想起を防止し，CCHN-COR における優れた連想性能へつながったと考えられる．

以上のことを確認するため，適当な初期状態から想起シミュレーションを行い，そのときのパターン空間成分とノイズ空間成分の時間変化を調べることにする．ここでパターン空間を上位  $P/2$  個の大きい固有値で表される部分空間と下位  $P/2$  個の小さい固有値で表される部分空間に分けることとし，前者を「パターン空間 (大)」，後者を「パターン空間 (小)」と呼ぶ．内部ポテンシャルの時間微分  $du/dt$  が含んでいる時刻  $t$  での固有空間成分は，以下のように計算される．

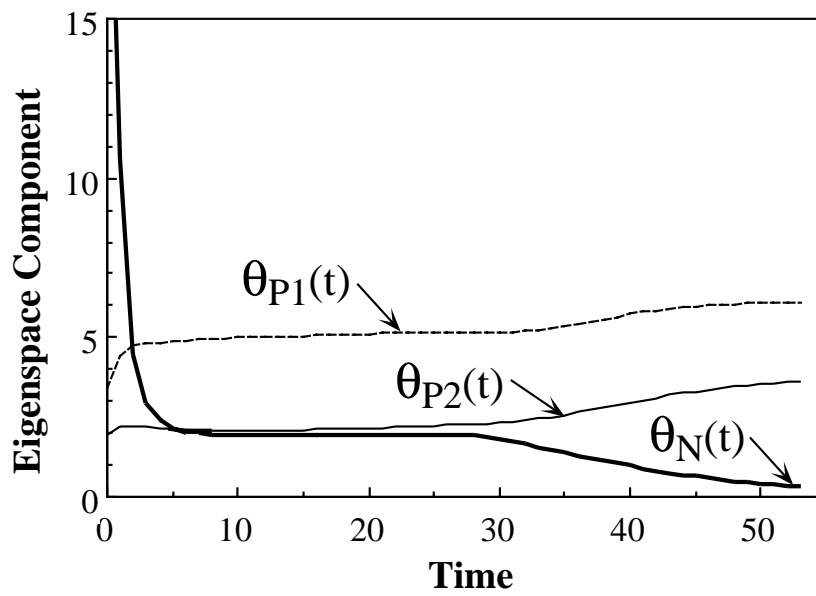
$$\text{(パターン空間 (大))} \quad \theta_{P_1}(t) = \frac{2}{P} \sum_{i=1}^{P/2} \left[ \left( \frac{du(t)}{dt} \right)' \mathbf{q}_i \right]^2 \quad (3.24)$$

$$\text{(パターン空間 (小))} \quad \theta_{P_2}(t) = \frac{2}{P} \sum_{i=P/2+1}^P \left[ \left( \frac{du(t)}{dt} \right)' \mathbf{q}_i \right]^2 \quad (3.25)$$

$$\text{(ノイズ空間)} \quad \theta_N(t) = \frac{1}{N-P} \sum_{i=P+1}^N \left[ \left( \frac{du(t)}{dt} \right)' \mathbf{q}_i \right]^2 \quad (3.26)$$



(a) HN-COR



(b) CCHN-COR

Figure 3.3: Time courses of eigenspace component ( $r = 0.2$ ,  $d^\mu(0) = 0.3$ )



ここで、 $q_i$  は  $i$  番目の固有ベクトルである。但し、HN-COR および CCHN-COR におけるユニット数、 $\alpha, \beta$  の値は前の実験で用いたものと同じとする。また相互作用の効果を調べるため、HN-COR が想起に失敗し、CCHN-COR が想起に成功するケースを選んだ。

注意 3.4 式 (3.24) ~ (3.26) からわかるように、固有空間成分  $\theta_{P_1}(t)$ ,  $\theta_{P_2}(t)$ ,  $\theta_N(t)$  は内部ポテンシャルベクトルの時間微分を固有空間に射影し、その絶対値を求めたものに相当する。よって、これらはそれぞれの固有空間における状態変化量の総和を表している。

Fig.3.3は、記憶率を 0.2、初期状態の方向余弦を 0.3 としたときの計算結果である。Fig.3.3(a) で分かるように HN-COR では、固有値の大きいパターン空間のダイナミクスが時間とともに優勢になっていく様子がわかる。つまり、パターン空間 (大) における状態変化量の増加に比べ、パターン空間 (小) における状態変化量はあまり変化しない。記憶パターン固有空間成分は固有値の大小に関わらず広く分布していることから、このことは明らかに偽記憶を想起する原因となっている。

一方、Fig.3.3(b) でわかるように、CCHN-COR では、まずノイズ空間成分の減少が起こり、30 ステップ程度でパターン空間成分の変化量が増加している。このとき、2つのパターン空間における固有空間成分は同程度の比率で増加しており、パターン空間 (大) だけでなく、パターン空間 (小) のダイナミクスもうまく機能していることがわかる。このことは、想起の初期段階で優勢となるノイズ空間ダイナミクスが最大固有値の空間成分の過度な増大を防止する役割をもっていることを示している。

### 3.3 1 モジュールモデルから導出される自己想起型連想記憶 (2) — 最適連想写像モデル

#### 3.3.1 自己想起型連想記憶モデルの導出

3.2では、モジュールの結合行列  $T$  を可変としたため結果的に相関行列が導かれた。しかし、モジュールのエネルギー関数における最小値が記憶パターンに一致するよう  $T$  をあらかじめ決めておいても、1 モジュールモデルを連想記憶装置として機能させることは可能である。モジュールのエネルギー関数は二次形式で与えられるので、ネットワークの状態  $v$  が任意の記憶パターン  $\xi^\mu$  上にあるとき、次式が成

り立つよう  $T$  を決めればよい。

$$T\xi^\mu = \xi^\mu \quad (\mu = 1, \dots, P) \quad (3.27)$$

これを満たす  $T$  の構成方法として，ここで最適連想写像モデルの結合行列を用いる。つまり，モジュールの結合行列は次のように構成される。

$$T = \mathbf{X}\mathbf{X}^+ \quad (3.28)$$

但し，

$$\mathbf{X} = (\xi^1, \xi^2, \dots, \xi^P) \quad (3.29)$$

$$\mathbf{X}^+ = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \quad (3.30)$$

前述したように， $\mathbf{X}^+$  は  $\mathbf{X}$  に対する Moore-Penrose の一般化逆行列である。

一方，インターネットワークの結合ダイナミクスに相当する式 (3.8) の微分方程式の解は，実は最適連想写像モデルの結合行列に収束する [61]。

$$\mathbf{w} = \mathbf{X}\mathbf{X}^+ \quad (3.31)$$

そこで，簡単のために  $\beta = 1$  として，式 (3.28), (3.31) を式 (3.3) に代入すると，

$$\begin{aligned} c \frac{d\mathbf{u}}{dt} &= [\alpha \mathbf{X}\mathbf{X}^+ - (\mathbf{I} - \mathbf{X}\mathbf{X}^+)'(\mathbf{I} - \mathbf{X}\mathbf{X}^+)] \mathbf{v} \\ &= [\alpha \mathbf{X}\mathbf{X}^+ - \mathbf{I} + \mathbf{X}\mathbf{X}^+] \mathbf{v} \\ &= \mathbf{G}_1 \mathbf{v} \end{aligned} \quad (3.32)$$

但し，

$$\mathbf{G}_1 = (1 + \alpha)\mathbf{X}\mathbf{X}^+ - \mathbf{I} \quad (3.33)$$

$$\mathbf{v} = \tanh(k\mathbf{u}). \quad (3.34)$$

ここで  $c$  は正の時定数， $k$  は出力関数の特性を決める正の定数である。  $\alpha$  も正の定数であり，モジュール情報処理の相互作用に対する相対的な貢献度を表す。シミュレーション実験において，要素が  $\{-1, 1\}$  である記憶パターンベクトルを用いるため，ユニットの出力関数は  $g(x) = \tanh(kx)$  としている。

最適連想写像モデル (HN-OPT) の動作方程式は，

$$c \frac{d\mathbf{u}}{dt} = \mathbf{G}_2 \mathbf{v} \quad (3.35)$$

但し，

$$\mathbf{G}_2 = \mathbf{X}\mathbf{X}^+ \quad (3.36)$$



いる点が大きく異なる。このことは、CCHN-OPT のパターン空間における動特性が最適連想写像モデルと本質的に同じであることを示している。つまり記憶パターンが互いに一次独立であれば、任意の記憶パターンセットに対して式 (3.27) が成り立ち、記憶パターン上での CCHN-OPT の安定性は常に保証される。これに対して HN-COR や CCHN-COR などの相関モデルでは、記憶パターンが互いに直交している場合にのみ上記の性質が保証される。よって HN-OPT や CCHN-OPT などの方が HN-COR や CCHN-COR に比べて、連想記憶装置として機能するために課せられる制約が少ないといえる。このことから、相関モデルは記憶パターンがランダムな場合によく用いられるのに対して、最適連想写像モデルは記憶パターン間に強い相関がある場合によく適用される。

パターン空間における動特性については、HN-OPT と CCHN-OPT は本質的に同じ性質をもつといえるが、両者の違いはノイズ空間において顕著となる。つまりノイズ空間を特徴付ける固有値は、HN-OPT の場合には 0 となるのに対して、CCHN-OPT の場合には  $-1$  となる。よって HN-OPT では、結合行列を介した線形変換によってノイズ空間成分を除去することができず、ユニットの出力関数を介した非線形変換に頼らざるを得ない。これに対し CCHN-OPT では、線形変換によってノイズ空間成分が単調に減少する。このノイズ空間成分の除去能力は、貢献度パラメータ  $\alpha$  に依存することが容易に想像される。つまり  $\alpha < 1$  とすれば、ノイズ空間ダイナミクスがネットワーク全体のダイナミクスに対して支配的となり、一般にノイズ空間成分の除去能力は高まると考えられる。3.2の結果において、ノイズ空間成分の除去能力が高まるほど引き込み領域が広がったことから、CCHN-OPT においても  $\alpha$  はなるべく 1 よりも小さな方が望ましいと予想される。これについては、次のシミュレーション実験で確認する。

### 3.3.3 シミュレーション実験および連想特性に対する考察

前述したように、HN-OPT や CCHN-OPT は記憶パターン間が互いに直交していない場合において特に有効である。そこで以下のシミュレーション実験では、Fig.3.4 のようにクラスタ状に分布した記憶パターンを適用する。記憶パターンにこのような構造を想定する方が、より現実的であることはいうまでもない。

それぞれの記憶パターン  $\xi^\mu$  ( $\mu = 1, \dots, P$ ) は、要素が  $\pm 1$  である  $N$  次元ベクトルで表されているとする (つまり、 $\xi^\mu \in \{-1, 1\}^N$ )。また、 $P$  個の記憶パターンは幾つかのクラスタに分類され、クラスタ内にある記憶パターン同士は一定の強い相関

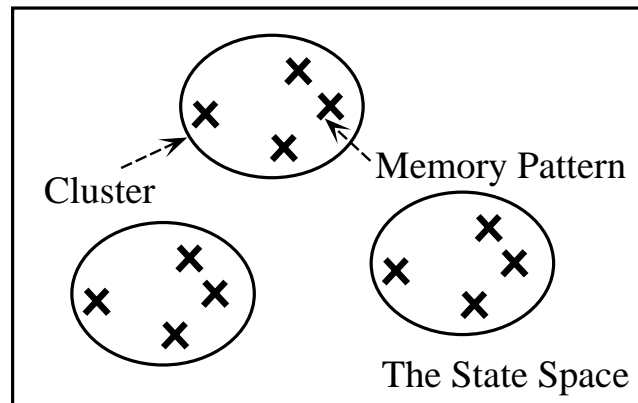


Figure 3.4: Schematic diagram of clustered memory patterns

をもっているものとする．このようなクラスタ状記憶パターンセットの定義方法には様々なものが考えられるが，ここでは以下のパラメータを用いて定義することにした．

- クラスタ数： $C$
- 1クラスタ当たりの記憶パターン数： $S$
- 同一クラスタ内にある記憶パターン間の平均相関： $\bar{\sigma}$
- クラスタ・セントロイド間の平均相関： $\bar{r}$

シミュレーション実験では，同一のクラスタ構造をもつものに対して5つの記憶パターンセットをランダムに生成し，これらに対する連想性能の平均値をシミュレーション結果とした．これは記憶パターンの選び方による性能のばらつきを平均化するためである．

ランダムパターンの場合，記憶パターンが互いに無相関に近いために記憶パターンセットをどのように選んでも問題の難しさは大きく変らない．しかしながら，クラスタ状記憶パターンの場合，上記のような少数のパラメータによっても問題の難しさはかなり変化する．そこで，どのようなクラスタ状記憶パターンに対しても優れた連想性能をもてるかどうか（つまり頑強性）は，連想記憶モデルの有効性を議論する上で重要な指標となる．

一般には，記憶パターン近傍の引き込み領域の大きさを調べ，それが大きいものほど優れた性能をもっているとするのが通常である．しかしながら，引き込み領域

Table 3.2: Parameter values

number of modules ( $M$ )	1
dimension of memory pattern vectors ( $N$ )	100
number of memory patterns ( $P$ )	30, 50, 70
number of clusters ( $C$ )	6, 10, 14
number of memory patterns in each cluster ( $S$ )	5
average correlation between patterns ( $\bar{\sigma}$ )	0.4
average correlation between cluster centroids ( $\bar{\kappa}$ )	0.0
number of hidden layers in internetworks ( $L$ )	0
contribution of module ( $\alpha$ )	0.0625, 0.125, 0.25, 0.5, 1.0, 1.5, 1.75, 1.875
contribution of interaction ( $\beta$ )	1.0

を制御して所望の連想特性が得られることが、連想記憶モデルとして重要であるとする場合もある。前述したように、貢献度パラメータ  $\alpha$  によって CCHN-OPT のダイナミクスが変化するため、この性質を利用すれば引き込み領域を制御できる可能性がある。そこで本節では、以下の2点に関して CCHN-OPT の連想特性をシミュレーション実験を通して考察する。

実験1：貢献度パラメータ  $\alpha$  が引き込み領域の大きさに与える影響

実験2：異なるクラスタ状記憶パターンに対するモデルの頑強性

#### (a) 実験1

記憶パターンのクラスタ構造は以下の通りとした。

$$C = 6, 10, 14, \quad S = 5, \quad \bar{\kappa} \approx 0, \quad \bar{\sigma} = 0.4$$

ここでは、簡単のため各クラスタは同数の記憶パターンで構成されると仮定する。記憶パターン数  $P$  は  $C \times S$  となり、それぞれ 30, 50, 70 となる。記憶パターンは  $\bar{\kappa} \approx 0$  と  $\bar{\sigma} = 0.4$  の条件を満たすようランダムに選ばれる。記憶パターンセットが決まれば、式 (3.33), (3.36) によって CCHN-OPT と HN-OPT の結合行列  $G_1, G_2$  が求められ、記憶パターン近傍の点を初期状態として想起が行われる。なお、これら初期状態は全ての記憶パターンに対して150点づつとっている。前述したように、1

種類のクラスタ構造に対して5つの記憶パターンセットの平均性能を求めるため、調べた初期状態の総数は  $750 \times M$  となる。その他のシミュレーション方法については、3.2.3で述べた通りである。主なパラメータの値を Table 3.2にまとめておく。

シミュレーション実験の結果を Fig.3.5に示す。これから分かるように、 $\alpha$  が大きく

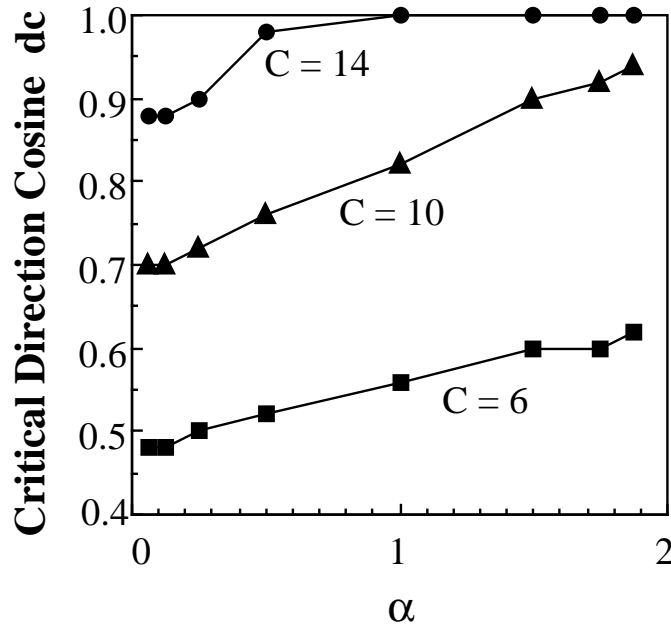


Figure 3.5: The critical direction cosines  $d_c$  for the  $\alpha$  values in the CCHN-OPT ( $S = 5$ ,  $\bar{\kappa} \approx 0$ , and  $\bar{\sigma} = 0.4$ ). The smaller  $d_c$  means the larger size of basins of attraction on average. The three lines represent the results for the different numbers of clusters  $C$ .

なるにつれて限界方向余弦  $d_c$  が大きくなる (つまり引き込み領域が小さくなる)。  $C = 14$  のときを除いて、ほぼ  $\alpha$  に対して直線的に  $d_c$  が変化しており、 $\alpha$  によって引き込み領域の大きさのある程度は調整できることがわかる。但し、 $\alpha$  が 0.125 以下になると、クラスタ数に関係なく連想性能はほとんど変化しない。つまり、CCHN-OPT における最大の引き込み領域は  $\alpha = 0.125$  で与えられることを示している。 $C = 14$  のときに  $\alpha - d_c$  の関係が直線的でないことから、記憶パターンが多くなって問題が難しくなると、パラメータ  $\alpha$  の選択が性能に敏感に反映されるといえる。

3.3.2で考察したように、 $\alpha < 1$  のときはノイズ空間のダイナミクスがパターン空間のダイナミクスに対して優位となり、状態のノイズ空間成分の除去が記憶パターンへの安定化よりも優先される。また、 $\alpha > 1$  のときはその逆のことがいえる。

Fig.3.5より,  $\alpha < 1$  とする方が引き込み領域を大きくできるといえ, これは3.2で考察した CCHN-COR に対する結果とも一致する. つまり CCHN-OPT についても, 状態のノイズ空間成分を単調減少させるダイナミックスが連想性能の向上に大きく寄与するといえる.

### (b) 実験 2

異なるクラスタ構造をもつ記憶パターンに対して連想特性を調べ, CCHN-OPT の連想記憶モデルとしての頑強性を調べる. 比較のため, HN-OPT の他に結合行列の対角要素を 0 とした改良モデル (HN-OPT2 で表す) を採用する. Kanter らは, このような HN-OPT2 が HN-OPT よりも大きな引き込み領域をもつことを示した [65].

CCHN-OPT の貢献度パラメータは, 実験 1 で最も大きな引き込み領域を与えた  $\alpha = 0.125$  と設定した. 記憶パターンのクラスタ構造は以下の条件のもとでランダムに選んだ.

$$C = 2 \sim 16, \quad S = 5, \quad \bar{\kappa} \approx 0, \quad \bar{\sigma} = 0.4, 0.7$$

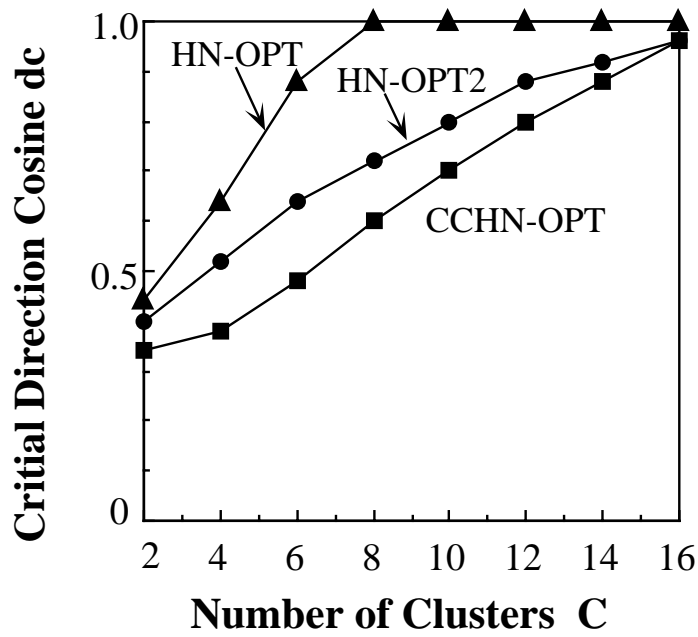
主なパラメータの値を Table 3.3にまとめておく.

Table 3.3: Parameter values

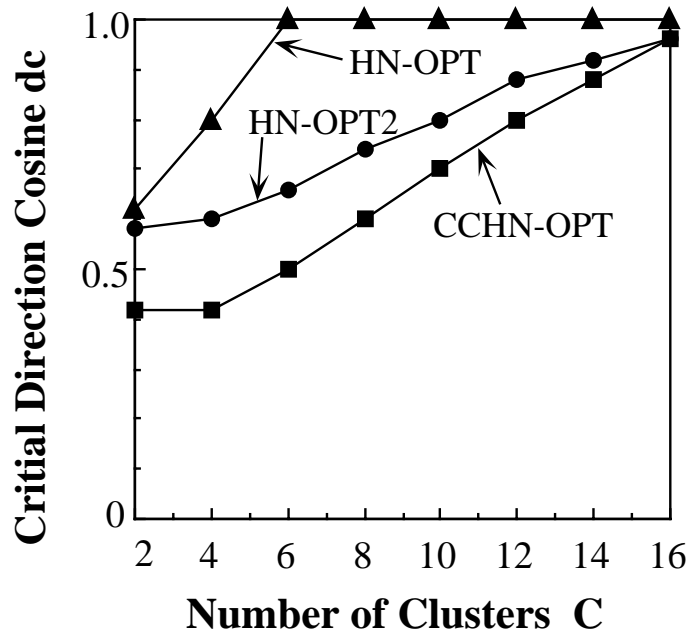
number of modules ( $M$ )	1
dimension of memory pattern vectors ( $N$ )	100
number of memory patterns ( $P$ )	10, 20, 30, 40 50, 60, 70, 80
number of clusters ( $C$ )	2 ~ 16
number of memory patterns in each cluster ( $S$ )	5
average correlation between patterns ( $\bar{\sigma}$ )	0.4, 0.7
average correlation between cluster centroids ( $\bar{\kappa}$ )	0.0
number of hidden layers in internetworks ( $L$ )	0
contribution of module ( $\alpha$ )	0.125
contribution of interaction ( $\beta$ )	1.0

Figs.3.6(a)(b) は, 同一クラスタに属する記憶パターンの平均相関を  $\bar{\sigma} = 0.4$  および 0.7 として性能比較を行った結果である. これより CCHN-OPT は殆どの  $C$  や  $\bar{\sigma}$  に対して HN-OPT や HN-OPT2 より高い連想性能を実現しており, 異なるクラス





(a)  $\bar{\sigma} = 0.4$



(b)  $\bar{\sigma} = 0.7$

Figure 3.6: The critical direction cosines  $d_c$  for the numbers of clusters  $C$  when the average correlation  $\bar{\sigma}$  is set to (a)0.4 or (b)0.7. The three lines represent the results of CCHN-OPT, HN-OPT, and HN-OPT2.



連想特性に近づいていくことになる．HN-OPT2 が最適な CCHN-OPT と等価になるには  $P = 0.89N$  のときである．

但し，本節で行ったシミュレーション実験では  $N$  がそれ程大きくないため，HN-OPT2 と CCHN-OPT における上記の対応関係が厳密に成り立つ訳ではない．

### 3.4 まとめ

本章では，ホップフィールドネットのエネルギー関数に誤差エネルギー（相互作用に対するエネルギー関数）を付加したニューラルネット（1 モジュールモデル）を取り上げた．この1モジュールモデルは1つのホップフィールドネットに相互作用を導入したモデルと考えられ，通常のホップフィールドネットとの比較を通して，マルチモジュール CCHN における相互作用の基本的な役割について議論することができる．

インターネットワークに中間層がない2層の階層型ニューラルネットを採用した場合，1モジュールモデルと等価な情報処理を行う自己想起型連想記憶モデルが導出されることを示した．これにはモジュールが相関モデルによって表されるものと最適連想写像モデルによって表されるものがあり，それぞれ CCHN-COR, CCHN-OPT と呼ばれる．ユニット数を  $N$  としたとき，CCHN-COR と CCHN-OPT の結合行列は  $N \times N$  の正方行列で表され，この性質を調べることで1モジュールモデルの動特性を考察することができる．本節では，結合行列に固有値解析を適用して動特性の考察を行う方法を提案した．

CCHN-COR の結合行列を固有値解析した結果およびランダムパターンに対する連想性能を調べた．結果を以下にまとめる．

1. CCHN-COR の動特性は従来型の相関モデル (HN-COR) と同じ固有空間で議論できる．このとき，記憶パターンが張る空間 (パターン空間) とその直交補空間であるノイズ空間に分けることで，HN-COR と CCHN-COR の動特性の違いが明確になる．
2. パターン空間では，HN-COR と CCHN-COR は同じ動特性をもつ．両者の違いはノイズ空間において顕著となる．
3. CCHN-COR のノイズ空間ダイナミクスは負の固有値で特徴付けられ，記憶パターン成分以外のノイズ成分を単調減少させる．
4. CCHN-COR における連想性能向上は，パターン空間ダイナミクスよりもノ

イズ空間ダイナミックスを優位にしたときに行われる。これは貢献度パラメータを調整することで行える。

5. ノイズ空間ダイナミックスが優位となるようにしたとき，ネットワークは状態のノイズ空間成分をまず減少し，それが十分に小さくなった後でパターンの想起が行われる。
6. 上記の動特性は，最大固有値の空間成分を過度に増大することを防止する。
7. CCHN-COR は，従来から用いられてきた相関モデルや最適写像モデルに比べて優れた想起能力をもつ（つまり引き込み領域が大きい）。

次に，CCHN-OPT の結合行列を固有値解析した結果およびクラスタ状パターンに対する連想性能を調べた結果を以下にまとめる。

- (i) CCHN-OPT においても，動特性はパターン空間とノイズ空間で特徴付けられる。
- (ii) CCHN-OPT のパターン空間における動特性は従来型の最適連想写像モデル (HN-OPT) と同じである。両者の違いは，CCHN-COR の場合と同様に，ノイズ空間において明確になる。
- (iii) CCHN-OPT におけるノイズ空間も負の固有値で特徴付けられ，状態のノイズ空間成分が単調に減少する。
- (iv) ノイズ空間ダイナミックスをパターン空間ダイナミックスよりも優位にしたとき，優れた連想性能が得られる。どちらのダイナミックスが優位になるかは，貢献度パラメータにより調節できる。
- (v) 貢献度パラメータを変更することで，引き込み領域を変化させることができる。特に記憶パターン数があまり多くない範囲では，貢献度パラメータの値と引き込み領域の大きさとの関係がほぼ線形になり，引き込み領域の制御がある程度可能である。
- (vi) 結合行列の対角要素がゼロである HN-OPT2 は，CCHN-OPT と同様にノイズ空間ダイナミックスが負の固有値で表される。しかし HN-OPT2 では，ノイズ空間ダイナミックスのパターン空間ダイナミックスに対する優位性が記憶パターン数によってしか変化せず，特に記憶パターン数が少ないときにおいてノイズ空間ダイナミックスを有効に利用していない。

(vii) HN-OPT, HN-OPT2, CCHN-OPT の 3 つのモデルに対してクラスタ状パターンを記憶させたときの連想性能を調べた。この結果、ノイズ空間ダイナミクスを有効に活用する CCHN-OPT の性能がどのクラスタ構造に対しても優れていることが示された。

CCHN-COR と CCHN-OPT の違いは、モジュールに相関モデルを用いるか最適連想写像モデルを用いるかである。つまり導入される相互作用に違いはない。また、上記の考察結果から両モデルは以下の点で共通した動特性をもっているといえる。

- 従来型モデル (HN-COR と HN-OPT) の動特性との違いは、ノイズ空間において明確になる。
- CCHN-COR と CCHN-OPT がもつノイズ空間ダイナミクスは本質的に同じである。

これらから以下のことが結論できる。

1 モジュールモデルに導入された相互作用によって、記憶パターン成分以外のノイズ成分を単調減少させるダイナミクスが発生する。これは連想記憶モデルとして望ましい性質を与える。

ここで相互作用の機能に関する定義 (p.21の定義 2.3) を思い出してみると、相互作用はモジュールの最終状態間に定義されたローカル・コンテキストを満足するように作用するのであった。

注意 3.6 1 モジュールモデルに導入された相互作用はクラス  $\mathcal{I}_M$  であるので、正確にはグローバル・コンテキストである。

ローカル・コンテキストの定義から明らかなように、これはパターン空間で表現されている。このこととノイズ空間ダイナミクスの特徴とを考え合わせると、モジュール間相互作用の情報処理機能を次のようにいい換えることができる。

モジュール間相互作用は、ネットワークの状態をローカル・コンテキストが表現されているパターン空間に射影する。

さて、相互作用を導入することで実現されたもう一つの特徴は、「貢献度パラメータによる引き込み領域の可変性」である。これは、1 モジュールモデルによって記憶容量や想起特性が改善されるだけでなく、多種多様な連想が行えることを示して

いる。但し、シミュレーション実験で明らかになったように、記憶パターン数やパターン構造によっては貢献度パラメータと引き込み特性との関係が線形でない場合もある。よって引き込み領域の大きさを厳密にコントロールすることは難しい。しかしながら、従来型の連想記憶モデルでは引き込み特性を変更することができないことを考慮すれば、「情報検索の柔軟性」という観点からも、1モジュールモデルの優れた連想特性を評価できるのではないかと思われる。

## 第4章 マルチモジュールニューラルネットの 基本モデル — 2モジュールモデル

### 4.1 まえがき

第3章では、誤差エネルギーをホップフィールドネットのエネルギー関数に加えた1モジュールモデルの動特性を考察し、相互作用を導入することで生じたダイナミクスの変化について議論した。相互作用がもつ基本的な性質を理解することは、より多くのモジュールで構成されたマルチモジュールモデルの動特性を理解する上で重要である。本章以降では、第3章において得られた結果をもとにマルチモジュールニューラルネットの動特性について考察していく。

マルチモジュールニューラルネットの動特性を考察する方法には、第3章と同様にモジュールとインターネットワークの結合荷重を1つの結合行列で表し、固有値解析を通してネットワークの動特性を議論する方法がまず挙げられる(但し、線形のインターネットワークであることが条件)。但し、複数のモジュールおよびインターネットワークの結合荷重を一つの結合行列で表すため、この手法はネットワーク全体の動特性を議論するには適しているが、モジュールの情報処理や相互作用の動特性を個々に議論することはできない。これらを個別に動特性を議論する方法としては、モジュールの状態や相互作用の信号を直接的にモニタリングする方法と、エネルギー関数の値としてモニタリングする方法が考えられる。しかしながら、この方法でネットワークの動特性を調べるには、あるモジュールが他のモジュールから受ける全ての相互作用について一つ一つ調べていく必要がある。このためモジュール数が多くなると、このアプローチによって考察することは難しくなる。

どちらの方法も一長一短であるが、本論文ではモジュール数が多いケースには前者の方法を用い、モジュール数が少ないケースには後者の方法で考察を行っていく。モジュール数が多いケースは次章で取り扱うこととし、本章ではマルチモジュールニューラルネットの基本モデルといえる「2モジュールモデル」を取り上げる。そしてモジュールの情報処理が相互作用によって受ける影響をモジュールごと考察する。

2モジュールモデルでは、2つのモジュールが互いに影響を与え合う様子を直接的に観察できるため、モジュール間相互作用の基本的な動特性について議論できる。

本章では、まず合計4ユニットからなる小規模な2モジュールモデルとユニット数が500である大規模なネットワークに対し、エネルギー平面の形状や引き込み領域、さらにエネルギー変動の様子などを観察し、相互作用が与えるモジュールへの影響について考察する。次に、最終状態間に一对多の関係が生じる可能性のあるケースとして文字パターン対の連想を取り上げる。そして、自己参照型相互作用を導入した2モジュールモデルに適用し、文字対間に定義された望ましい対応関係を満たしながら正しい想起が行われることを確認する。また、記憶パターンがランダムであるときとクラスタ状に分布しているときの両方のケースを取り上げ、2モジュールモデルの連想性能を評価する。

## 4.2 基本動特性の考察

本節の目的は、相互作用を介して2つのモジュールがどのように影響を与え合うのかを考察することにある。4.1で述べたように、なるべく個々の情報処理に対して直接的な考察を試みるが、これにはモジュールの状態や相互作用の信号を観測する方法とエネルギー関数の値を観測する方法とがある。

前者では、モジュールの情報処理および相互作用に対し、次の量が観測される。

- モジュールの情報処理： モジュールの状態と最終状態との方向余弦
- 相互作用： インターネットワーク出力とモジュールの状態との誤差

これに対して後者では、上記の観測量をエネルギー関数の値として求める。モジュールの情報処理と相互作用を個別に観測する場合は、どちらの観測量で考察しても大きな違いは生じない。しかし、ネットワークはエネルギー関数を単調減少するよう動作してしていることから、エネルギー関数の値をモニタする方がその絶対値に対する意味付けは行いやすい。そこで本節では、主にエネルギー関数の値を観測する方法で2モジュールモデルの基本動特性を考察していく。

### 4.2.1 エネルギー関数の形状に対する考察

まず、モジュールのエネルギー関数が相互作用によってどのように影響を受けるのかを、エネルギー関数の形状を視覚化することで考察する。視覚化するにはあま



り状態空間の次元を大きくできないため、以下では、各モジュールのユニット数が2(合計4ユニット)であるシンプルな2モジュールモデルを取り上げる。

モジュールの状態空間は2次元となり、それぞれの状態を  $v^{(1)} = (v_1^{(1)} v_2^{(1)})'$ ,  $v^{(2)} = (v_1^{(2)} v_2^{(2)})'$  で表す。モジュールにおける安定平衡点およびそれらの望ましい対応関係を Table 4.1のように仮定し、次式で表されるクラス  $\mathcal{I}_1$  の線形インターネットワークでモジュール間相互作用を実現する。

$$o_i^{(1, \bar{m}, m)} = \sum_{j=1}^2 w_{ij}^{(1, \bar{m}, m)} v_j^{(\bar{m})} \quad (m = 1, 2; \bar{m} = 2, 1) \quad (4.1)$$

Table 4.1において、モジュール1および2はそれぞれHN1, HN2と表されている。また各モジュールに埋め込まれる安定平衡点は2つ存在し、それぞれ  $A_1^{(*)}$ ,  $A_2^{(*)}$

Table 4.1: The mapping relations between two point attractors

Mapping No.	Mapping Relation Between Attractors	HN1		HN2	
		$v_1^{(1)}$	$v_2^{(1)}$	$v_1^{(2)}$	$v_2^{(2)}$
1	$A_1^{(1)} \iff A_1^{(2)}$	0	0	1	0
2	$A_2^{(1)} \iff A_2^{(2)}$	1	1	0	1

(\* = 1, 2) で表される。表中には、これら安定平衡点がどのような状態であるのかが記されており、例えばHN1の1番目の安定平衡点は  $A_1^{(1)} = \{0, 0\}$  であることがわかる。また、 $A_1^{(1)} \iff A_1^{(2)}$  と  $A_2^{(1)} \iff A_2^{(2)}$  は2つのモジュールの安定平衡点に与えられている望ましい対応関係(ローカル・コンテキスト)を表している。

このような安定平衡点を最終状態とするエネルギー関数を求める。但し、Table 4.1のケースで相関行列を用いて結合荷重を決めてもネットワークは正しく動作しない。この理由は、ユニット数が2のモジュールに対して2つのパターンを記憶しなければならない、記憶率が1となってしまうことから容易に想像できる。また最適写像モデルを用いることも考えられるが、記憶率1の場合には結合行列が単位行列となってしまう、引き込み領域が失われてしまう。そこで、各安定平衡点が満たすべき条件からエネルギー関数を定義する方法を採用する。このアイデアは、Hopfieldが巡回セールスマン問題 [9] や A/D 変換問題 [66] で提案した方法に基づいている。以下に、本問題に対するエネルギー関数を示す。

$$E = \alpha^{(1)} E^{(1)} + \alpha^{(2)} E^{(2)} + \beta^{(1,1,2)} E^{(1,1,2)} + \beta^{(1,2,1)} E^{(1,2,1)}$$

$$\begin{aligned}
&= \frac{1}{2}\alpha^{(1)}(v_1^{(1)} - v_2^{(1)})^2 + \frac{1}{2}\alpha^{(2)}[(v_1^{(2)} - 1)v_2^{(2)} + v_1^{(2)}(v_2^{(2)} - 1) + 1]^2 \\
&\quad + \frac{1}{2}\sum_{m=1}^2\sum_{i=1}^2\alpha^{(k)}v_i^{(k)2}(v_i^{(k)} - 1)^2 \\
&\quad + \frac{1}{2}\beta^{(1,2,1)}\sum_{i=1}^2(v_i^{(1)} - o_i^{(1,2,1)})^2 + \frac{1}{2}\beta^{(1,1,2)}\sum_{i=1}^2(v_i^{(2)} - o_i^{(1,1,2)})^2 \quad (4.2)
\end{aligned}$$

右辺2段目の第1, 2項は, HN1 および HN2 において表中の安定平衡点がエネルギー関数の最小点になるようにする項である. また第3項はモジュールの状態が  $\{0, 1\}$  のどちらかを取るように制約する項である. 第4, 5項は相互作用を定義する誤差関数である.

式(4.2)を時間微分することでネットワークの動作方程式が求められるが, これまでのモデルと異なりモジュールの結合ダイナミクスは導出されない. モジュールの結合荷重は, 式(4.2)を時間微分したときに現われる状態  $v_i^{(m)}$  の係数として与えられる. 2.2で示した方法に基づいてモジュールの活性ダイナミクスを求めた結果を以下に示しておく.

[第1モジュール]

$$\begin{aligned}
c\frac{du_i^{(1)}}{dt} &= -\alpha^{(1)}(P_i^{(1)} + v_i^{(1)} - v_{\bar{i}}^{(1)}) \\
&\quad -\beta^{(1,2,1)}(v_i^{(1)} - o_i^{(1,2,1)}) + \beta^{(1,1,2)}\sum_j(v_j^{(2)} - o_j^{(1,1,2)})w_{ji}^{(1,1,2)} \quad (4.3) \\
&\quad (i = 1, 2; \bar{i} = 2, 1)
\end{aligned}$$

[第2モジュール]

$$\begin{aligned}
c\frac{du_i^{(2)}}{dt} &= -\alpha^{(2)}(P_i^{(2)} + v_i^{(2)} + 4v_i^{(2)}v_{\bar{i}}^{(2)} + 4v_{\bar{i}}^{(2)}v_i^{(2)2} - v_{\bar{i}}^{(2)} + 2v_{\bar{i}}^{(2)2} - 1) \\
&\quad -\beta^{(1,1,2)}(v_i^{(2)} - o_i^{(1,1,2)}) + \beta^{(1,2,1)}\sum_j(v_j^{(1)} - o_j^{(1,2,1)})w_{ji}^{(1,2,1)} \quad (4.4) \\
&\quad (i = 1, 2; \bar{i} = 2, 1)
\end{aligned}$$

但し,

$$P_i^{(m)} = \alpha^{(k)}(4v_i^{(m)3} - 6v_i^{(m)2} + 2v_i^{(m)}) \quad (4.5)$$

式(4.3),(4.4)から分かるように, 相互作用に関連した項はこれまでのモデルと同じであるが, モジュールの情報処理に関連した項がかなり異なっている. 上述したように, 式(4.3),(4.4)の右辺第一項中にある  $v_i^{(m)}$  にかかる係数が結合荷重に相当する. 但し, この活性ダイナミクスには  $v_i^{(m)}$  の高次項が含まれており, 通常のホッ

プフィールドネットを使って実現することができない．これは非線形項を計算する特別なユニットを追加したネットワークによってインプリメント可能である．このようなネットワークが従来のホップフィールドネットの性能を向上させることは既によく知られている [49] ．

注意 4.1 ここで用いるモジュールは非線形計算ユニットを追加した特別なホップフィールドネットで実現される．しかしながら，モジュール情報処理を実現する手法がこれまで取り扱ってきたものと異なるだけであって，p.16で定義したモジュールの情報処理機能は変わっていない．よって，このような2モジュールモデルに対して，モジュールが受ける相互作用の影響を考察しても議論の一般性が失われることはない．

インターネットワークの結合ダイナミクスは，これまでのモデルと同じ手順に基づいて導出される．よって，2.3.3で示した時定数に対する仮定を採用すれば，インターネットワークの結合は Table 4.1に示されている最終状態間の関係を逆誤差伝搬法によって求めることになる．

まず，この2モジュールモデルにおけるモジュールの振舞いについて調べた．Fig.4.1は，HN2のいくつかの初期状態に対してHN1の引き込み領域がどのように変化するかを表したものである．Fig.4.1における横軸と縦軸はHN1の状態  $v_1^{(1)}$ ， $v_2^{(1)}$  を表しており，図中の左下角およびは右上角2つの安定平衡点  $A_1^{(1)}$ ， $A_2^{(1)}$  を表している．また，図中の実線は引き込み領域の境界を表しており，HN1の状態が実線よりも左下にあれば安定平衡点  $A_1^{(1)}$  に引き込まれ，右上にあれば安定平衡点  $A_2^{(1)}$  に引き込まれることを示している．

この結果より，HN1の引き込み領域はHN2の初期状態により変動することがわかる．例えば  $v^{(2)} = (0.7, 0.3)$  のとき，引き込み領域の境界は  $A_1^{(1)}$  よりも  $A_2^{(1)}$  に近く， $A_1^{(1)}$  の引き込み領域の方が広いことを示している．いうまでもなく，HN2との相互作用がなければ  $A_1^{(1)}$  と  $A_2^{(1)}$  の引き込み領域は同じ大きさになるため，HN2との相互作用が  $A_1^{(1)}$  の引き込み領域を拡大したと考えてよい．この理由は，HN2の初期状態が安定平衡点  $A_1^{(2)}$  に近いため， $A_1^{(1)} \Leftrightarrow A_1^{(2)}$  という対応関係を満足するように相互作用が働いたからである．当然ながら，HN2の引き込み領域もHN1の状態に依存して拡大縮小する．このように，あるモジュールの引き込み領域は他のモジュールの状態に依存してダイナミックに変動しながら，最終的にある安定平衡点で安定する．

次に，状態空間におけるエネルギー平面の勾配を調べることで，上記の動特性を

Figure 4.1: Basin boundary between two attractors in HN1.

別の視点から考察を行ってみる．ネットワーク全体のエネルギー関数から勾配ベクトルを求めるが，状態空間が4次元となるため，各モジュールの状態が次の関係を満たす点でのみ勾配を調べることにした．

$$v_1^{(1)} = v_2^{(1)}, \quad v_1^{(2)} = 1 - v_2^{(2)}$$

Fig.4.2は，求めた勾配ベクトルを  $v_1^{(1)} - v_1^{(2)}$  平面（または  $v_2^{(1)} - v_2^{(2)}$  平面）に射影して得られたベクトル場を表している．矢印の向きは勾配方向を，長さは勾配ベクトルの大きさを表している．ここで，前の考察で取り上げた挙げた例について考えてみる．つまり  $v^{(2)}$  を  $(0.7, 0.3)$  に固定した場合において， $v^{(1)}$  を Fig.4.2の点線上で変化させたときの勾配ベクトルについて調べる．但し， $v^{(1)}$  は  $v_1^{(1)} = v_2^{(1)}$  の関係を保ったまま変化させるので，Fig.4.1の斜めの点線上の状態を調べていることに相当する．このとき，Fig.4.1の結果から，引き込み領域の境界は  $v^{(1)} = (0.78, 0.78)$  付近にあることがわかる．Fig.4.2において，確かに  $(0.8, 0.8)$  付近より右側では  $A_2^{(2)}$  (図

Figure 4.2: A calculated vector field of energy gradient by adding the adequate constraints,  $v_1^{(1)} = v_2^{(1)}$  and  $v_1^{(2)} = 1 - v_2^{(2)}$ . These constraints give the calculation on the dashed line in Fig.4.1.

中の右上角) への勾配が大きくなっていくことがわかる。ここで興味深いのは、HN1の状態が  $(0.7, 0.7)$  のときに、勾配ベクトルの方向が点線よりも上部にあることである。これは HN2 の状態空間において、一旦は安定平衡点  $A_1^{(2)}$  (図中の左下角) から遠ざかる方向に引き込まれながらも、結局は  $A_1^{(2)}$  に引き込まれてしまうことを意味している。つまり HN1 と HN2 との間に相互作用を介した競合が起こっていると解釈できる。

この様子は、Fig.4.3をみればさらに明確になる。Fig.4.3は、いくつかの初期状態からネットワークを動作させたときの状態軌跡を示したものである。これからわかるように、引き込み領域の境界付近に相当する図中左上角および右下角の地点から出発した場合、一旦、状態空間の真ん中方向に状態が移っていく。これは、2つの安

Figure 4.3: State trajectories of two modules from some initial states. This results are obtained from the calculated vector field in Fig.4.2.

定平衡点への引き込みが拮抗している様子を表しており、2つのモジュールにおいて競合が起こっていることを意味する。ここで用いている動作方程式は全体のエネルギーを単調に減少させるだけにも関わらず、個々のモジュールに注目すればこのような競合ダイナミクスが生じている点は興味深い。

上記の考察に対して、HN1のエネルギー関数の形状をさらに視覚的に分かりやすくしたのが Figs.4.4(a)-(d) である。Fig.4.4(a) は、HN2との相互作用をもたないHN1のエネルギー平面である。それに対して、Figs.4.4(b)-(d) は  $v^{(2)}$  をそれぞれ  $(0.9, 0.1)$ ,  $(0.5, 0.5)$ ,  $(0.1, 0.9)$  としたときのHN1のエネルギー平面を表している。 $v^{(2)} = (0.9, 0.1)$ ,  $(0.1, 0.9)$  としたときには、それぞれ  $A_1^{(1)}$ ,  $A_2^{(1)}$  の引き込み領域が拡大している様子がよくわかる。つまり、HN2の状態との対応関係を満足しやすいように、HN1のエネルギー関数が適応的に変化していることを表している。また

(a) without cross-coupling

(b)  $(v_i^{(2)}v_2^{(2)}) = (0.9, 0.1)$

(c)  $(v_i^{(2)}v_2^{(2)}) = (0.5, 0.5)$

(d)  $(v_i^{(2)}v_2^{(2)}) = (0.1, 0.9)$

Figure 4.4: Energy planes in the state space,  $(v_1^{(1)} \times v_2^{(1)})$ . (a) Energy plane of HN1 without cross-coupling. (b)-(d) Energy planes of HN1 with cross-coupling for 3 different initial states,  $(v_1^{(2)}(0), v_2^{(2)}(0)) = (0.9, 0.1), (0.5, 0.5), (0.1, 0.9)$ .

$v^{(2)} = (0.5, 0.5)$  のとき, HN1 の状態が引き込み領域の境界付近にあれば, 一旦, 状態空間の真ん中に引き込まれることが容易に確認できる.

#### 4.2.2 想起過程におけるエネルギー変動の考察

4.2.1では, エネルギー関数の形状を観察することでネットワークの動特性を考察した. ここでは, モジュールと相互作用のエネルギー変動の様子を観察することで, 2モジュールモデルにおける動特性の考察を試みる. 2モジュールモデルを連想記憶に適用し, 相関モデル (HN-COR) の想起過程と比較を行いながら考察する.

ここで用いられる2モジュールモデルには, クラス  $\mathcal{I}_M$  の大局的相互作用を導入した. これは, 2.3で取り上げたマルチモジュール CCHN-LAGI において局所的相互作用の貢献度を  $\beta^{(2,n,m)} = 0$  としたものと同じである. また, インターネットワークには中間層をもたない2層の階層型ニューラルネットを用いており, このような2モジュールモデルは CCHN-GI/L(2) で表される (付録 A.2を参照).

CCHN-GI/L(2) をうまく動作させるためには, 相互作用の貢献度をある程度大きくする必要があり, ここでは経験的に  $\alpha^{(m)} = 1$ ,  $\beta^{(M,1,m)} = 0.4$  と決めた. また, 2モジュールモデルの総ユニット数は500であり, これを2つのモジュールに等分して割り当てた. また, ユニット数が500である連続時間型 HN-COR を比較のため用いた. 記憶させるパターン数は150とし, 記憶パターン  $\xi^\mu$  ( $\mu = 1, \dots, 150$ ) の各要素は  $\pm 1$  をランダムに選ぶことで生成した. CCHN-GI/L(2) への記憶パターンの埋め込みは, まず  $\xi^\mu$  を2つの250次元部分ベクトル  $\xi^{\mu(1)}$ ,  $\xi^{\mu(2)}$  に分割し, これらから式 (2.30), (2.37) ~ (2.42) に基づいてオフラインで結合荷重の学習を行う. 但し, 時定数に  $c \ll \gamma$  を仮定している. シミュレーション実験で用いられた主なパラメータ値を Table 4.2にまとめておく.

想起に用いられるネットワークの初期状態は, 1番目の記憶パターン  $\xi^1$  との方向余弦が  $0.0 \sim 1.0$  で均等に選ばれるようランダムに与えた.

注意 4.2 記憶パターンはランダムに選ばれているため,  $\xi^1$  に特定した議論を行っても一般性が失われることはない.

なお, 時刻  $t$  における方向余弦  $d(t)$  は, HN-COR に対して式 (4.6), CCHN-GI/L(2) に対して式 (4.7) で定義される.

$$d(t) = \frac{1}{500} \sum_{i=1}^{500} \xi_i^1 v_i(t) \quad (4.6)$$



Table 4.2: Parameter values

number of modules ( $M$ )	2
dimension of memory pattern vectors ( $N$ )	500
number of memory patterns ( $P$ )	150
number of hidden layers in internetworks ( $L$ )	0
class of interactions	$\mathcal{I}_M$
contribution of module ( $\alpha^{(m)}$ )	1.0
contribution of interaction ( $\beta^{(M,1,m)}$ )	0.4

$$d(t) = \frac{1}{500} \sum_{m=1}^2 \sum_{i=1}^{250} \xi_i^{1(m)} v_i^{(m)}(t) \quad (4.7)$$

ここで、 $v_i$  は HN-COR におけるユニットの状態を表す。

HN-COR および CCHN-GI/L(2) の想起過程の様子を、それぞれ Figs.4.5(a)(b) に示す。ここで、横軸  $t$  は時定数  $c^{(m)} (= 1.0)$  を単位とした時間である。Fig.4.5(a) から明らかなように、HN-COR ではどのような初期状態から想起を開始しても正しいパターンは想起されない。これに対して、CCHN-GI/L(2) では  $d(0) = 0.4 \sim 0.5$  付近で想起特性に変化が見られ、 $d(0)$  がそれよりも大きいと  $\xi^1$  が正しく想起される。3.2で述べた1モジュールモデルの場合と同様に、インターネットワークに中間層がない CCHN-GI/L(2) は HN-COR と構造的に同じネットワークで表すことができる。このことから、明示的にモジュール間の相互作用を導入したことによる効果が、このような想起ダイナミクスの特性に反映したといえる。

以上のような CCHN-GI/L(2) の想起ダイナミクスには、モジュール間の相互作用を定義する誤差エネルギーが効果的に関与すると考えられる。そこで、ネットワークの各部分エネルギーがどのように緩和されるかについて調べ、CCHN-GI/L(2) の想起ダイナミクスの特徴について考察する。

Figs.4.6(a)-(d) に、CCHN-GI/L(2) のエネルギー項  $E^{(1)}$ ,  $E^{(2)}$ ,  $E^{(M,1,2)}$ ,  $E^{(M,1,1)}$  の時間変化を、Fig.4.5のケースを例にとって示す。 $E^{(1)}$ ,  $E^{(2)}$  が大きいとき、モジュールの状態が基本記憶にたどり着いていないことを、 $E^{(M,1,2)}$ ,  $E^{(M,1,1)}$  が大きいとき、モジュール間の写像関係が十分満たされていないことを意味する。

Figs.4.6(a)-(d) でみられる特徴を以下にまとめる。

1. モジュールのエネルギー項  $E^{(1)}$ ,  $E^{(2)}$  は増加減少を繰り返し、正しい想起が行

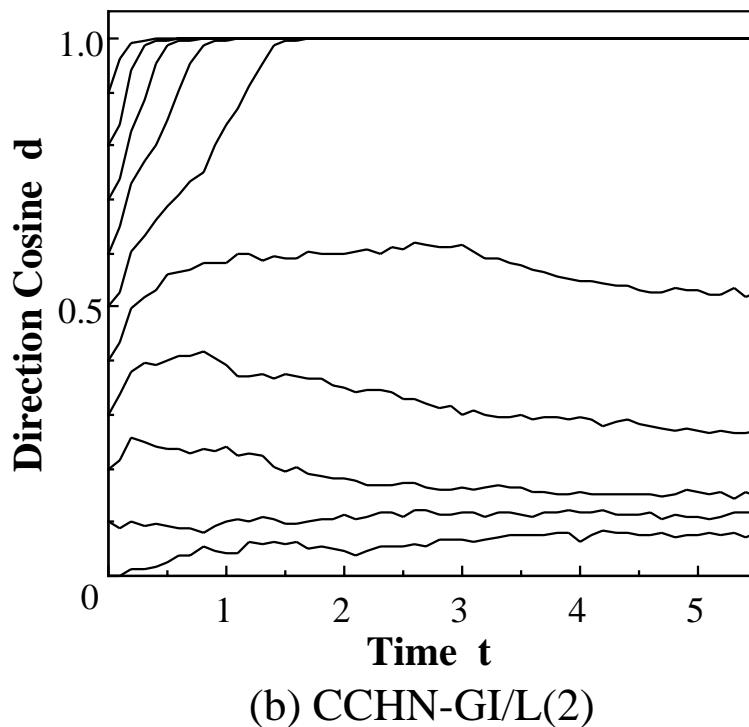
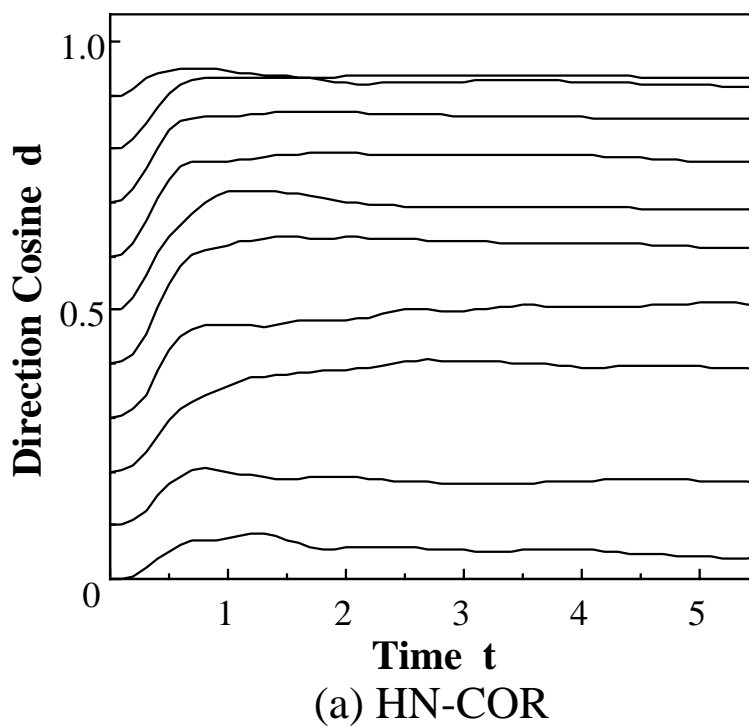


Figure 4.5: Recall processes in (a)HN-COR, (b)CCHN-GI/L(2).

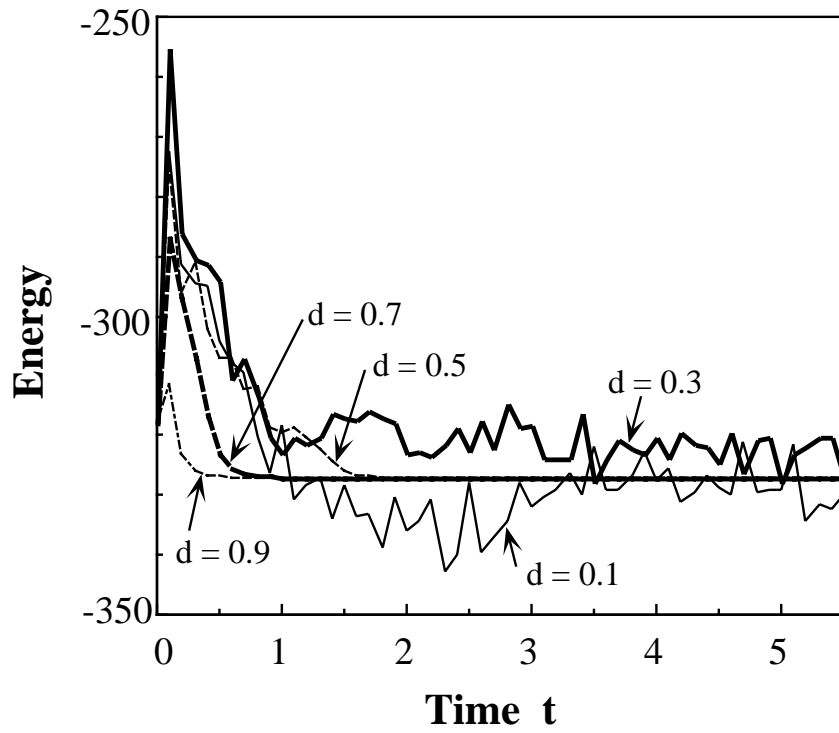
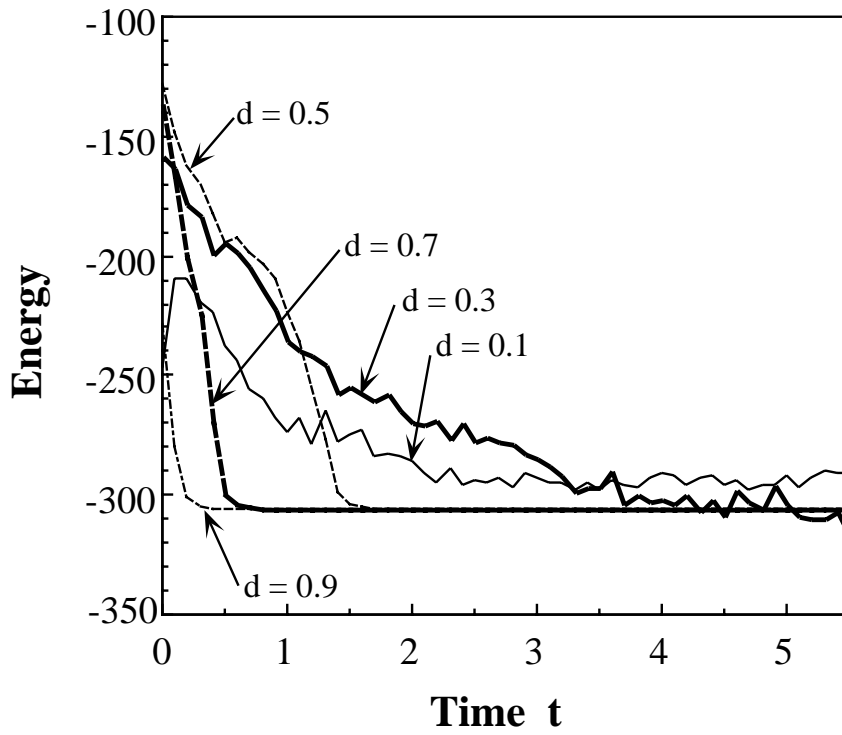
(a)  $E^{(1)}$ (b)  $E^{(2)}$ 

Figure 4.6: Temporal change of energy terms  $E$ , (a)  $E^{(1)}$ , (b)  $E^{(2)}$ , (c)  $E^{(M,1,2)}$ , (d)  $E^{(M,1,1)}$  in CCHN-GI/L(2).

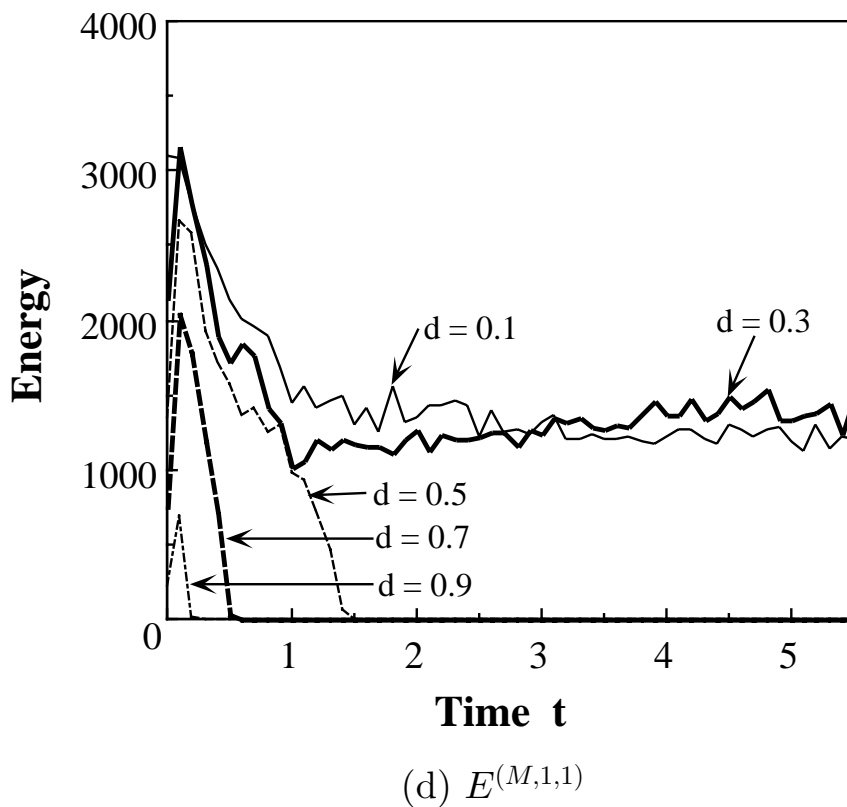
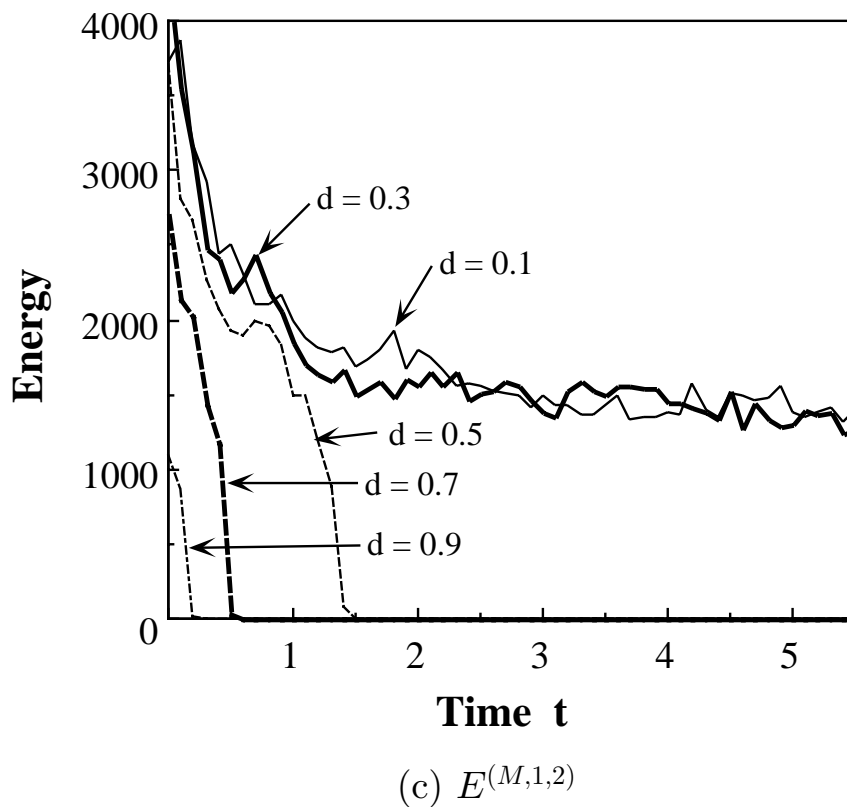


Figure 4.6: Temporal change of energy terms  $E$ , (a)  $E^{(1)}$ , (b)  $E^{(2)}$ , (c)  $E^{(M,1,2)}$ , (d)  $E^{(M,1,1)}$  in CCHN-GI/L(2).

われる場合には最終的に極小点で安定する。

2.  $E^{(M,1,2)}$ ,  $E^{(M,1,1)}$  はいくらか振動しながらも大局的には減少していく。しかし,  $E^{(1)}$ ,  $E^{(2)}$  が安定化されるためには, これらがある程度小さくなる必要がある。
3. 正しい想起が行われない場合, それぞれの部分エネルギーは振動を繰り返し, 安定化されない。

1. の結果は, 各モジュールが自己のエネルギーを常に減少させるわけではなく, ときにはエネルギーを上昇させる方向に動作することを示している。しかし, ネットワークダイナミクスの導出方法からも明らかなように, ネットワーク全体のエネルギーは常に減少する。このことと 2. の結果を考慮すれば, CCHN-GI/L(2) では, まずモジュール間の対応関係に矛盾がなくなるよう状態の探索が行われ, ある程度矛盾が解消された時点で個々のモジュールの安定化が行われると考えられる。この傾向は初期値の方向余弦が小さいときに特に顕著であることから, 偽記憶を想起しやすい場合に,  $E^{(M,1,2)}$ ,  $E^{(M,1,1)}$  の緩和が有効に働くと考えられる。この結果は, 3.2 の 1 モジュールモデルで考察したノイズ空間ダイナミクスの特徴と一致する。また 3. の結果より, 正しく想起できない場合には, モジュール間の写像関係が矛盾した状態のまま  $E^{(1)}$ ,  $E^{(2)}$  の緩和と  $E^{(M,1,2)}$ ,  $E^{(M,1,1)}$  の緩和が拮抗することがわかる。

以上の考察結果は, 4.2.1 で扱った小規模な 2 モジュールモデルにおけるエネルギー平面の考察結果にほぼ一致する。このことから, ここで扱った大規模な 2 モジュールモデルにおいても, 4.2.1 で示したようなエネルギー平面の変動が生じていることが予想される。

## 4.3 連想記憶への適用と性能評価

前節において, 2 モジュールモデルの基本的な動特性について考察し, モジュール構造をもたない従来のホップフィールドネットとの違いを示した。ここでは, 2 モジュールモデルを連想記憶に適用したときに, 上記の動特性がどのように連想特性に反映されるかについて, いくつかのシミュレーション実験を通して調べていく。

### 4.3.1 文字パターン対の連想

2.3 で示したように, マルチモジュールモデルにおいてはモジュールの最終状態間に一対多の関係があるとき, 導入する相互作用の種類に注意する必要があった。こ

のような一対多の関係を生じないようにする最も簡単な方法は、自己参照型相互作用を導入することであった。ここでは、モジュールの最終状態間に一対多の関係が存在する例を取り上げ、クラス  $\mathcal{I}_M$  の相互作用を導入した2モジュールモデルに適用する。

インターネットワークには、3層の階層型ニューラルネットを用いる。これは、2.3で取り上げたマルチモジュール CCHN-LAGI においてモジュール数を  $M = 2$ 、中間層数を  $L = 1$ 、局所的相互作用の貢献度を  $\beta^{(2,n,m)} = 0$  としたものに等しい。そこで、この2モジュールモデルを CCHN-GI/NL(2) と略すことにする(モデルの表記方法については付録 A.2を参照)。

記憶パターンは、画素数  $64(8 \times 8)$ 、2階調 ( $\pm 1$ ) の文字データが対となった4組の文字パターン対 ( $C \leftrightarrow B$ ), ( $A \leftrightarrow B$ ), ( $A \leftrightarrow C$ ), ( $C \leftrightarrow C$ ) である。明らかに、このパターン対には一対多の対応関係が含まれていることがわかる。モジュールのユニット数は64、インターネットワークの中間ユニット数は5とした。文字パターン対を構成する2つの文字はそれぞれ2つのモジュールに割り当てられ、式(2.30)に基づいてモジュールの結合荷重が求められる。その後、式(2.37)~(2.42)に基づいてインターネットワークの学習が行なわれる。なお、 $\alpha^{(m)}$ 、 $\beta^{(M,n,m)}$  はそれぞれ1.0, 0.5とした。ここで用いる主なパラメータの値を Table 4.3にまとめておく。

Table 4.3: Parameter values

number of modules ( $M$ )	2
dimension of memory pattern vectors ( $N$ )	128
number of memory patterns ( $P$ )	4
number of hidden layers in internetworks ( $L$ )	1
number of hidden units in internetworks ( $H^{(2)}$ )	5
class of interactions	$\mathcal{I}_M$
contribution of module ( $\alpha^{(m)}$ )	1.0
contribution of interaction ( $\beta^{(M,1,m)}$ )	0.5

Figs.4.7(a)(b) には、パターン対 ( $C \leftrightarrow B$ ), ( $A \leftrightarrow B$ ) に30%のランダムノイズを加えて初期状態としたときの想起過程を示す。第2モジュールで最終状態となったのは“B”であり、これに対応する第1モジュールの最終状態には“A”と“C”の2通りが考えられる(つまり一対多の関係にある)。よって、第1モジュールの状態次第で

Figure 4.7: Recall processes of CCHN-GI/NL(2) from noisy character patterns.

$(C \leftrightarrow B)$  が想起されるか、 $(A \leftrightarrow B)$  が想起されるかが決まるような特性をもっていることが望まれる。Figs.4.7(a)(b) からわかるように、第2モジュールの初期状態はともに同じであるにも関わらず、第1モジュールの初期状態に依存して2つの文字パターン対が想起されている。

また、モジュールに与えられる初期状態が欠損をもった場合についても想起実験をおこなった。Figs.4.7(c)(d) に想起過程の例を示す。これらの場合、第2モジュールで最終状態となったのは“C”であり、これに対応する第1モジュールの最終状態にはやはり“A”と“C”の2通りが考えられる。このときも第2モジュールには同じ初期状態が与えられており、第1モジュールの初期状態に依存して、 $(A \leftrightarrow C)$  と  $(C \leftrightarrow C)$  の2つの文字パターン対が想起されている。

このことから、パターン対に一对多の関係がある場合にも相互作用が正しく機能

し, CCHN-GI/NL(2) は連想記憶装置として望ましい性質をもっているといえる. なお, モジュール間に相互作用をもたない CCHN-GI/NL(2) (すなわち, 2つの独立した HN) では, いずれのケースも正しい想起が行えない.

### 4.3.2 ランダムパターンに対する連想性能

2モジュールモデルのランダムパターンに対する連想特性を定量的に評価する. ここではクラス  $\mathcal{I}_M$  の相互作用を導入した2モジュールモデルを取り上げる. インターネットワークには, 2層および3層の階層型ニューラルネットを用いているおり, 2.3で取り上げたマルチモジュール CCHN-LAGI においてモジュール数を  $M = 2$ , 中間層数を  $L = 0, 1$ , 局所的相互作用の貢献度を  $\beta^{(2,n,m)} = 0$  としたものに等しい. なお,  $L = 0$  であるものを CCHN-GI/L(2),  $L = 1$  であるものを CCHN-GI/NL(2) と略す (付録 A.2を参照). また比較のため, HN-COR にも適用して連想性能を調べる. 記憶パターン  $\xi^\mu$  ( $\mu = 1, 2, \dots, P$ ) はランダムに選ばれた 500次元のベクトルとし, モジュールの総ユニット数も 500 である. その他の実験条件, 学習方法, 想起方法は 4.2.2と同じである. ここで用いる各種パラメータの値を Table 4.4にまとめておく.

Table 4.4: Parameter values

number of modules ( $M$ )	2
dimension of memory pattern vectors ( $N$ )	500
number of memory patterns ( $P$ )	10 ~ 500
number of hidden layers in internetworks ( $L$ )	0, 1
number of hidden units in internetworks ( $H^{(2)}$ )	250
class of interactions	$\mathcal{I}_M$
contribution of module ( $\alpha^{(m)}$ )	1.0
contribution of interaction ( $\beta^{(M,1,m)}$ )	0.4

Fig.4.8は, 記憶率  $r$  に対する限界方向余弦  $d_c$  を調べた結果である. 記憶率  $r$  は p.48 の定義 3.3に基づき, 記憶パターンベクトルの次元に対する記憶パターン数の比で定義する. なお, 限界方向余弦は定義 3.4において  $\rho = 0.5$  としたものをを用いる.

注意 4.3 連想記憶において, 記憶率はネットワークのユニット数に対する記憶パターン数で定義することが多い. この定義に基づくと, CCHN-GI/NL(2) は中間ユ



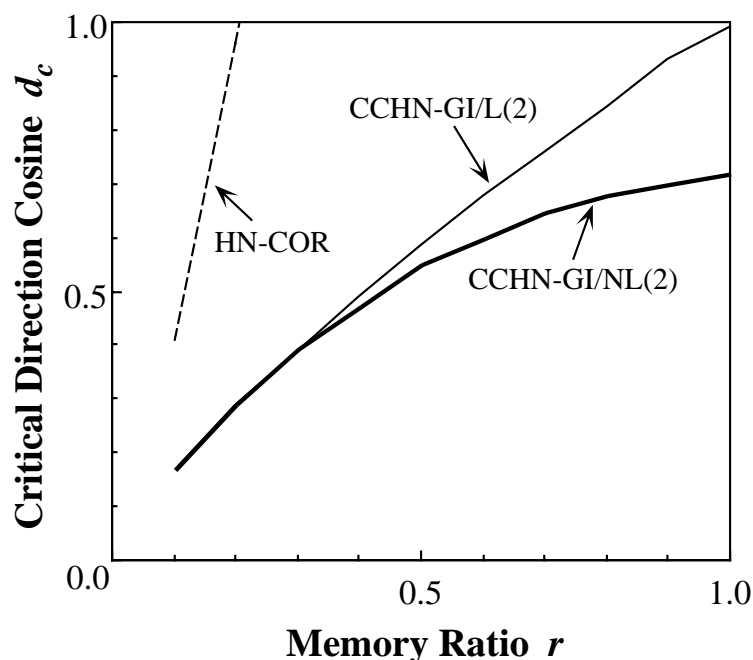


Figure 4.8: Critical direction cosines  $d_c$  for memory ratio.

ユニットだけ総ユニット数が多くなってしまふ。しかしながら，CCHN-GI/NL(2)におけるインターネットワークはモジュール間相互作用を生成するのに用いられ，パターンを直接的に記憶するために用いられるわけではない。よってマルチモジュールニューラルネットに対して，従来の記憶率の定義を適用することは適当でないと考えた。

得られたシミュレーション結果を以下にまとめる。

1. すべての記憶率  $r$  に対して，CCHN-GI/L(2) と CCHN-GI/NL(2) の引き込み領域は HN-COR よりも広い。
2. HN-COR では  $r \simeq 0.2$  以降で引き込み領域をもたないのに対し，CCHN-GI/L(2) および CCHN-GI/NL(2) では  $r = 1.0$  付近でも引き込み領域をもつ。
3. 特に CCHN-GI/NL(2) に関しては， $r$  が大きくなるに従って  $d_c$  の増加は緩やかになる。

CCHN-GI/L(2) と CCHN-GI/NL(2) の比較であるが， $r$  が小さいときにはそれほど顕著な差は見られない。しかし， $r$  が大きい場合，特に  $r > 0.3$  となるときには後者の性能が前者を上回る。これは，記憶パターン数の増加に伴ってインターネッ

トワークで獲得すべき写像が複雑となり，中間層をもたないインターネットワークでは十分な学習が行なえないためと考えられる．このことは，モジュールの最終状態間に定義された対応関係 (ローカル・コンテキスト) を非線形関数で表現することの有効性を示している．つまり，モジュール間に非線形な相互作用を導入することによって，2モジュールモデルの連想性能が向上することを示唆している．

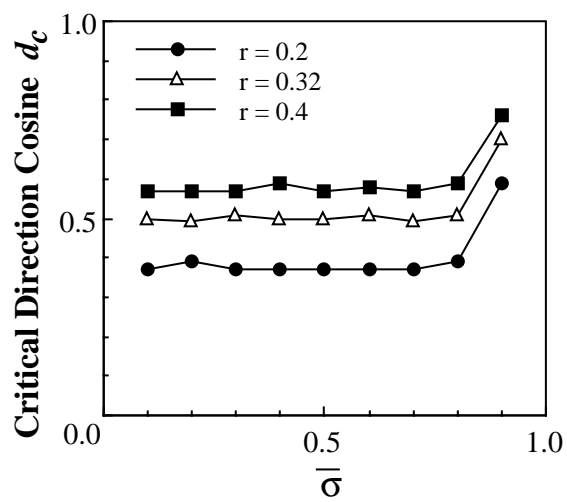
### 4.3.3 クラスタ状パターンに対する連想性能

前節では，2モジュールモデルである CCHN-GI/L(2) と CCHN-GI/NL(2) がランダムパターンに対して優れた連想性能をもつことを示した．ここでは，クラスタ状に分布した記憶パターンに対して，上記2種類のモデルの連想性能を調べてみる．

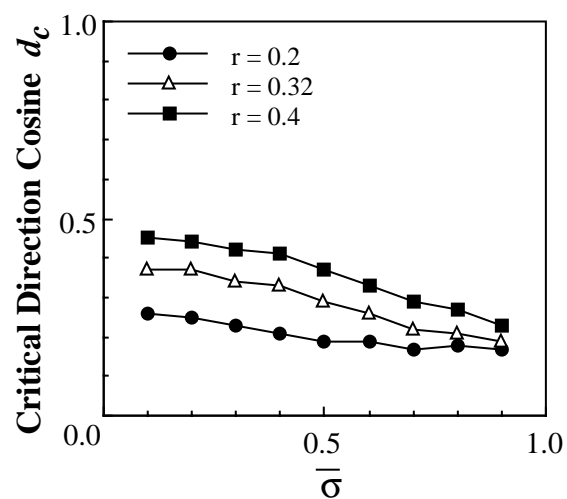
記憶パターンベクトルの次元は100で，これを2つの50次元ベクトルに分割して CCHN-GI/L(2) と CCHN-GI/NL(2) に記憶する．なお，CCHN-GI/NL(2) におけるインターネットワークの中間ユニットは50とした．比較のため，100ユニットの最適連想写像モデル (HN-OPT) の連想性能も調べた．記憶パターンの構造には，3.3.3で取り上げたのと同じクラスタ状のものを用いた．本シミュレーションで用いた主なパラメータの値を Table 4.5 にまとめておく．

Table 4.5: Parameter values

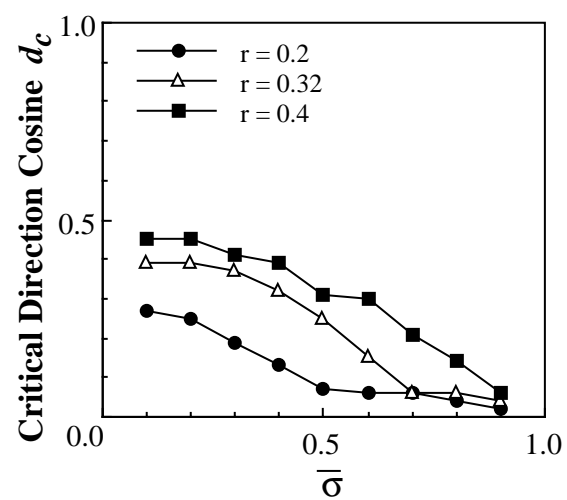
number of modules ( $M$ )	2
dimension of memory pattern vectors ( $N$ )	100
number of memory patterns ( $P$ )	20, 32, 40
number of clusters ( $C$ )	4
number of memory patterns in each cluster ( $S$ )	5, 8, 10
average correlation between patterns ( $\bar{\sigma}$ )	0.1, 0.2, 0.3, 0.4, 0.5 0.6, 0.7, 0.8, 0.9
average correlation between cluster centroids ( $\bar{\kappa}$ )	0.0
number of hidden layers in internetworks ( $L$ )	0, 1
number of hidden units in internetworks ( $H^{(2)}$ )	50
class of interactions	$\mathcal{I}_M$
contribution of module ( $\alpha^{(m)}$ )	1.0
contribution of interaction ( $\beta^{(M,1,m)}$ )	0.25



(a) HN-OPT



(b) CCHN-GI/L(2)



(c) CCHN-GI/NL(2)

Figure 4.9: Critical direction cosine  $d_c$  for memory ratio  $r$  and correlation  $\bar{\sigma}$ .

Figs.4.9(a)-(c) は、記憶率  $r (=P/N)$  とパターンの平均相関  $\bar{\sigma}$  に対する HN-OPT, CCHN-GI/L(2), CCHN-GI/NL(2) の限界方向余弦  $d_c$  を調べた結果である。なお、限界方向余弦は p.49の定義 3.4 において  $\rho = 0.5$  としたものをを用いる。

これからわかるように、すべての  $r$  と  $\bar{\sigma}$  に対して CCHN-GI/L(2) と CCHN-GI/NL(2) の引き込み領域は HN-OPT のそれより大きい。ここで注意すべきは、CCHN-GI/L(2) と CCHN-GI/NL(2) におけるモジュールの結合荷重が相関行列で与えられている点である。HN-OPT の代わりに相関モデル (HN-COR) を用いた場合、ここで用いた記憶パターンの想起はほとんど行えない。このことから考えて、モジュール間の相互作用が如何に連想記憶能力の向上に寄与しているかがわかる。また CCHN-GI/L(2) と CCHN-GI/NL(2) の結果を比べると、記憶パターン数が多かったり、パターン間相関が高くなる程、相互作用を非線形とすることの効果が大きく現われるといえる。

## 4.4 まとめ

本章ではマルチモジュールニューラルネットの基本モデルである 2モジュールモデルを取り上げ、モジュールの情報処理が相互作用によって受ける影響について考察した。この方法として、引き込み領域やエネルギー平面の形状、さらにエネルギーの時間変動の様子をモジュールごとに調べることにし、モジュールが受ける相互作用の影響について議論した。以下に結果をまとめる。

1. 小規模な 2モジュールモデル (1モジュール当たり 2ユニット) に対して 2種類の最終状態を与えたとき、以下のような結果を得た。
  - (a) モジュールの引き込み領域は、他方のモジュールの状態に依存してダイナミックに拡大縮小する。
  - (b) モジュールのエネルギー関数は他方のモジュールの状態に依存して、その形状をダイナミックに変化させる。但し、2モジュールモデルはエネルギー関数を最小化しよう動作するため、この結果は (a) の結果と本質的に同じといってよい。
  - (c) ネットワーク全体の初期状態が 2つの最終状態のどちらを想起するかあいまいなケースでは、2モジュール間に相互作用を介した競合が生じる。これは状態空間での状態軌跡を調べた結果からも、エネルギー平面の形状を調べた結果からも確認された。

2. 大規模な 2 モジュールモデル (1 モジュール当たり 250 ユニット) に対して 150 個の最終状態を与えたとき、以下のような結果を得た。

- (a) モジュール情報処理のエネルギー関数を個々に調べると、エネルギーは時間的に増減を繰り返し、正しい想起が行われるケースでは最終的に安定する。正しくない想起が行われるケースでは、全体のエネルギーが安定したとしてもモジュールのエネルギーは安定化されない。
- (b) 個々の相互作用のエネルギーに注目すると、いくらか上下に振動しながらも大局的には時間とともに減少する。このとき、モジュールのエネルギー関数が安定化されるためには、相互作用のエネルギーが一定レベルまで下がる必要がある。
- (c) 正しい想起が行われない場合、モジュール情報処理と相互作用のエネルギーはすべて安定化されない。しかしながら、全体のエネルギーは安定化されることから、個々のエネルギー項が振動しているのは、モジュール間の競合が解消されないためと考えられる。

上記の 2 種類のシミュレーション結果から、ネットワークの規模に関わらず、2 モジュールモデルがもつ基本動特性として以下のものが考えられる。

- (i) モジュールは、他方のモジュールの状態に依存して自己の情報処理様式をダイナミックに変化させている。これは明らかにモジュール間相互作用が存在するためである。
- (ii) モジュールの状態がローカル・コンテキスト (2 モジュールモデルなのでグローバル・コンテキストといってもよい) を簡単に満たすような状況にないとき、モジュール間に相互作用を介した競合が生じる。競合状態はダイナミックに相互作用し合うことでやがて解消され、ある最終状態に落ち着く。これは、あるローカル・コンテキストを満足させるために 2 モジュール間で起こった「協調的行動」ともいえる。

また上記の基本動特性が連想性能としてどのように反映されるのかについて調べられた。その結果、記憶パターンがランダムな構造をもっている場合、クラスタ状に分布した構造をもっている場合、2 モジュールモデルは従来手法である相関モデルや最適写像モデルより優れた連想性能をもつことが示された。また、インターネットワークに 3 層の階層型ニューラルネットを用いた 2 モジュールモデルについても連

想性能を調べた。その結果、記憶パターン数が多いときやパターン間相関が強いときに優れた想起特性(引き込み特性)をもつことがわかった。このことは、モジュールの最終状態間に定義された対応関係(ローカル・コンテキスト)を非線形関数で表現することの有効性を示唆している。つまり、モジュール間に非線形な相互作用を導入することによって、2モジュールモデルの連想性能が向上することを示している。

また、モジュールの最終状態間に一對多の対応関係が存在する例として文字対パターンの連想を取り上げ、自己参照型相互作用を導入した2モジュールモデルに適用した。その結果、一方のモジュールに同じ初期状態を与えた2種類のケースにおいて、他方の初期状態に依存して最終的に想起される文字対パターンが選択されることを確認した。このことから、パターン対に一對多の関係がある場合にも相互作用が正しく機能し、2モジュールモデルは連想記憶として望ましい性質をもっているといえる。

以下に、2モジュールモデルを連想記憶モデルとして評価した結果を示す。

- I. 2モジュール間に相互作用を導入することで、パターンの構造に関係なく優れた連想性能が実現される。
- II. 記憶パターン数が多い場合やパターン間相関が強い場合、モジュール間相互作用に非線形性をもたせることが有効である。
- III. モジュールの最終状態間に一對多の関係がある場合でも、自己参照型相互作用を導入することで正しい想起が行われる。

# 第5章 マルチモジュールニューラルネットの動特性に関する考察および連想記憶への適用

## 5.1 まえがき

第2章および第3章において、本論文で提案したマルチモジュールニューラルネットの基本動特性と連想性能について様々な観点から考察を行ってきた。その結果、2モジュール以下のモデルに対しては、相互作用がネットワークのダイナミクスにどのように影響を与えるかについて、かなり理解できるようになったと思われる。

これまでに得られた結果は、一般的なマルチモジュールモデルの動特性を理解する上でも役立つことは間違いないであろう。しかしながら、これらはマルチモジュールモデルの基本的な動特性にすぎず、多数のモジュールで構成されたネットワークでなければ生じない動特性も多くあると考えられる。このことは、p.24の Table 2.1 に示した本モデルの構造パラメータにおいて、2モジュール以下のモデルがこれらのごく限られたものしか考慮していないことから容易に予想される。

本モデルが含んでいるバリエーションは構造パラメータを変更することで与えられるが、その数は膨大である。例えば、相互作用のクラスが与えるバリエーションは、単にクラスの違いだけでなく、同一クラスであってもインターネットワークの入力情報をどのように選ぶかで多くのケースが考えられる。またモジュール数が増えれば、上記のバリエーションはさらに増えていく。このようなバリエーションはネットワークに多種多様な動特性を与えると考えられるが、残念ながら、この多様性について効率よく考察する手段は今のところない。

そこで本章では、考慮する構造パラメータを以下のものに限定し、マルチモジュールモデルの動特性を考察していくことにする。

- モジュール数
- 相互作用のクラス
- モジュールおよび相互作用の貢献度

但し，導入する相互作用のクラスは  $\mathcal{I}_2$  と  $\mathcal{I}_M$  の2つとし，さらに自己参照型相互作用に限ることとする (2.3で導出したマルチモジュール CCHN-LAGI に相当)．これは，モデルの本質を損なわず，なるべく単純なかたちで以下の点を議論するためである．

- 異種の相互作用が同時に導入されることの効果

本節の前半では，上記の点についてマルチモジュールモデルの動特性を考察する．ここでは，3.2で行った1モジュールモデルに対する動特性の解析手法を採用する．つまり，モジュールの情報処理と相互作用を一つの相互想起型連想記憶モデル (HN) の情報処理として表現し，その結合行列に対して固有値解析を行うことで動特性の考察を行う．そこで，まずモジュール数が  $M$  であるときのマルチモジュールモデルと構造的に等価な HN の結合行列を求める．次に，この結合行列の固有値を数値的に求め，上述の点について議論を試みる．また後半では，マルチモジュールモデルがもっている構造の多様性が実際に動特性にどのように反映されるのかについて，連想記憶の問題に適用することで議論する．

## 5.2 異種の相互作用を導入する効果

### 5.2.1 マルチモジュール CCHN-LAGI に導入される相互作用

本節では，2.3で導出したクラス  $\mathcal{I}_2$  と  $\mathcal{I}_M$  の2種類の相互作用をもつマルチモジュール CCHN-LAGI を取り上げ，異種の相互作用が導入されることの効果について考察する．インターネットワークには2層の階層型ニューラルネットを用い，ユニットの出力関数には線形のものを用いる．モジュール数が  $M$  とすると，このようなマルチモジュール CCHN-LAGI は CCHN-LAGI/L( $M$ ) で表される (付録 A.2を参照)．

マルチモジュール CCHN-LAGI に導入される2種類の相互作用について，以下のように仮定する．

クラス  $\mathcal{I}_2$  : 考えられる全ての2モジュール間に自己参照型相互作用を導入する．すなわち，局所的相互作用 (クラス  $\mathcal{I}_2$ ) は  $M(M-1)/2$  種類ある．

クラス  $\mathcal{I}_M$  :  $M/4$  個のモジュールにだけ大局的相互作用 (クラス  $\mathcal{I}_M$ ) を導入する．

便宜上，大局的相互作用はインデックスの小さなモジュールから順に導入されるものとする．



例 5.1 8 モジュールで構成される場合，局所的相互作用は合計  $(8 \times 7)/2 = 28$ ，大局的相互作用は合計  $8/4 = 2$  だけ導入される．また，大局的相互作用は第 1 モジュールと第 2 モジュールにだけ導入する．

なお，この仮定の下では， $M < 4$  のネットワークには大局的相互作用が導入されないことになる（但し， $M = 2$  の場合，局所的相互作用と大局的相互作用が同じになることに注意）．

局所的相互作用はモジュール間の局所的な状態関係（ローカル・コンテキスト）を満足させるように働くのに対し，大局的相互作用はネットワークの大局的な状態関係（グローバル・コンテキスト）を満足させるように働く．本モデルを連想記憶に適用する場合，記憶パターンをモジュールの個数分だけ分割し，それらをモジュールに割り当てて記憶する．よって最終的に正しくパターン想起が行われるためには，すべてのモジュールの状態がグローバル・コンテキストを満足しなければならない．しかしながら，上述の仮定のもとでは，すべてのモジュールが直接的に大局的相互作用に影響を受けるわけではない．局所的相互作用によってしか影響を受けないモジュールでは，他のモジュールとの局所的相互作用を介してグローバル・コンテキストの情報を獲得しなければならない．

CCHN-LAGI/L( $M$ ) に，大局的相互作用をすべてのモジュールに導入しないのは，ある一部のモジュールにだけ与えられたグローバル・コンテキストの情報が局所的相互作用を介して他のモジュールに伝搬するかどうかを確認するためである．つまり，2 種類の相互作用の協調によって生み出されるネットワークの動特性を考察し，これらを同時に導入することの効果について議論する．また，他のモジュールとの局所的相互作用のみを介してグローバル・コンテキストの情報を獲得できるかも調べておく必要がある．そこで，局所的相互作用だけが導入された CCHN-LI/L( $M$ ) の連想特性も調べる．

### 5.2.2 連想性能の評価

マルチモジュールモデルに記憶させるパターンは，各要素の値をランダムに  $\pm 1$  とした 512 次元のランダムパターンとする．各モジュールには，記憶パターンを  $M$  等分した  $512/M$  次元の部分記憶ベクトルを割り当てる．これらからモジュールとインターネットワークの結合荷重が式 (2.30) ~ 式 (2.42) に基づいて求められる．

想起方法および性能評価の方法は 3.2.3 で述べた通りである．限界方向余弦は p.49 の定義に基づき， $\rho = 0.98$  として求めた（つまり，98%以上想起に成功する最小の方向

余弦を限界方向余弦とする)。なお, 1 実験当たりの想起回数は記憶パターン数の約 10 倍とした。記憶パターン数  $P$  やモジュール数  $M$  など, その他の実験条件については Table 5.1 にまとめている。

Table 5.1: Parameter values

number of modules ( $M$ )	2,4,8,16
dimension of memory pattern vectors ( $N$ )	512
number of memory patterns ( $P$ )	52, 103, 154, 205, 256, 308, 359, 410, 461, 512
number of hidden layers in internetworks ( $L$ )	0
class of interactions	$\mathcal{I}_2 + \mathcal{I}_M$
contribution of module ( $\alpha^{(m)}$ )	1.0
contribution of local interaction ( $\beta^{(2,n,m)}$ )	0.1
contribution of global interaction ( $\beta^{(M,1,m)}$ )	1.0

CCHN-LAGI/ $L(M)$  と CCHN-LI/ $L(M)$  の連想性能を Figs.5.1(a)(b) に示す。横軸の記憶率  $r$  は p.48 の定義 3.3 に基づいている。また参考のため, Fig.5.2 には 3.2 で取り上げた相関モデル (HN-COR), 最適写像モデル (HN-OPT) および 1 モジュールモデル (CCHN-COR) の連想性能も示しておく。

Fig.5.1(b) からわかるように, 大局的相互作用が導入されていない CCHN-LI/ $L(M)$  の連想性能は, モジュール数が多くなるほど劣化していく。クラス  $\mathcal{I}_2$  インターネットワークは 2 モジュール間のローカル・コンテキストを獲得しており, それに基づいて行われる相互作用は, モジュール数が多くなるにつれてネットワークの局所的な情報しか反映されない。よって, 1 つのモジュールに影響を与えている相互作用間に一種の競合状態が生じる可能性が高くなり, これが解消されない場合, 正しい記憶パターンの想起ができなくなる。これが, モジュール数の増加による CCHN-LI/ $L(M)$  における連想性能の劣化につながっているものと考えられる [67]。

一方, Fig.5.1(a) からわかるように, 大局的相互作用が導入された CCHN-LAGI/ $L(M)$  では連想性能がかなり改善される。Fig.5.2 における HN-OPT の性能と比べても, CCHN-LAGI( $M$ ) の性能は  $M = 16$  のときを除いて同程度かそれ以上の連想性能をもっているのがわかる。ここで, 大局的相互作用は一部のモジュールにしか与えられていない点に注意する必要がある。つまりこの連想性能の改善は, 大局的相互作

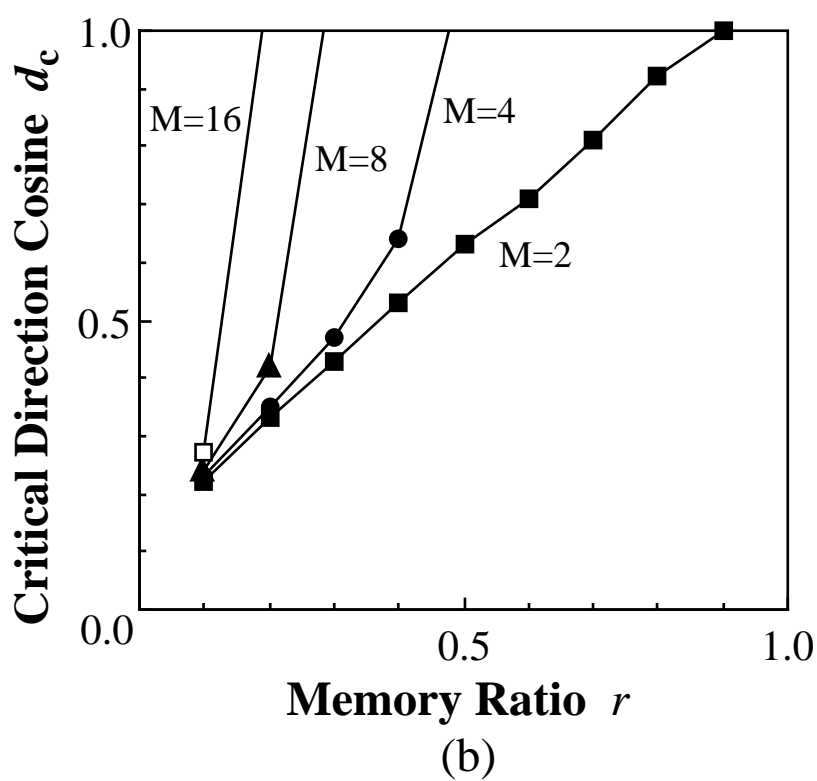
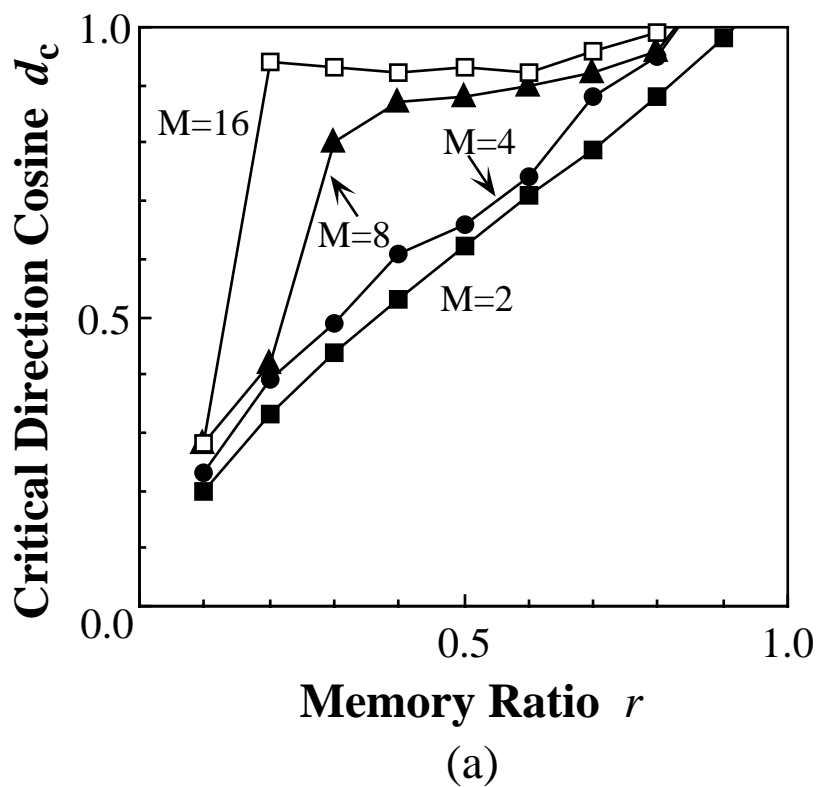


Figure 5.1: The average association performance for different number of modules in (a)CCHN-LAGI/L( $M$ ) and (b)CCHN-LI/L( $M$ ).

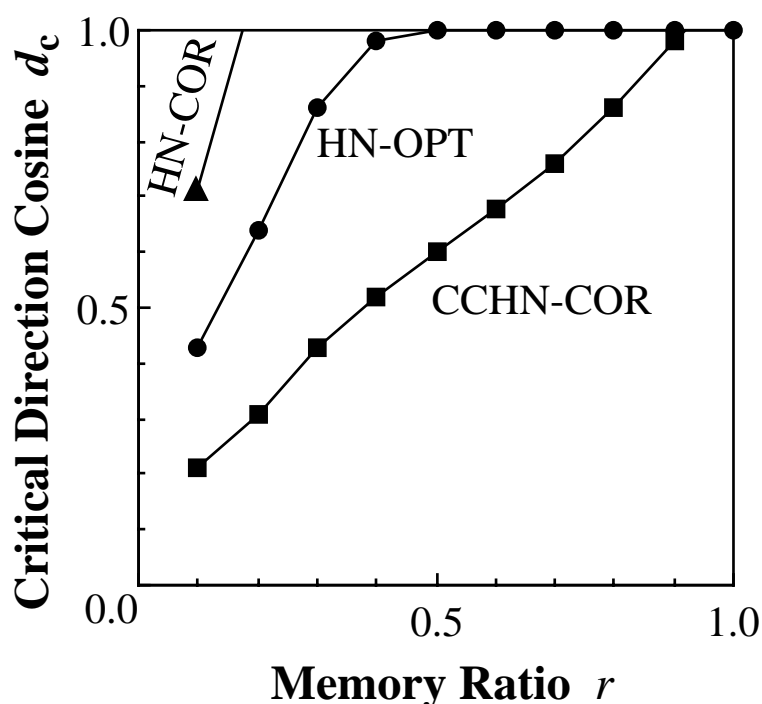


Figure 5.2: The average association performance of 1-module CCHN, HN-COR, and HN-OPT.

用を介して一部のモジュールに与えられたグローバル・コンテキストの情報が局所的相互作用を介してすべてのモジュールに伝搬した結果を反映していると考えられる。これは、2種類の相互作用の協調によって生み出される動特性が連想性能の改善に寄与することを示唆している。

### 5.2.3 結合行列の固有値解析

5.2.2では、2種類の相互作用が導入されることの効果で連想性能を調べることで示した。ここでは、3.2で行った解析手法をCCHN-LAGI/L( $M$ )に適用し、2種類の相互作用を導入することでどのようなダイナミクスの変化が起こったのかについて考察する。

まず、CCHN-LAGI/L( $M$ )と等価な情報処理を行う自己想起型連想記憶モデル(ホップフィールドネット)の結合行列を求める。前述したが、このような導出が可能となる条件は以下の通りである。

1. インターネットワークは中間層をもたない。

2. インターネットワークの出力関数は線形である .

3. 式 (2.24) ~ (2.27) の時定数に  $c \ll \varepsilon, \gamma$  の関係がある .

CCHN-LAGI/L( $M$ ) における各モジュールのユニット総数を  $N$  としたとき , 最終的に求められるネットワークの結合荷重は  $N \times N$  行列である . そこで , まずモジュールおよびインターネットワークの結合荷重を  $N \times N$  行列で表す . 変換前の結合行列をそれぞれを  $\mathbf{T}^{(m)} = \{T_{ij}^{(m)}\} \in \mathcal{R}^{N^{(m)}} \times \mathcal{R}^{N^{(m)}}$  ,  $\mathbf{w}^{(2,n,m)} = \{w_{ij}^{(2,n,m)}\} \in \mathcal{R}^{N^{(m)}} \times \mathcal{R}^{N^{(n)}+N^{(m)}}$  ,  $\mathbf{w}^{(M,1,m)} = \{w_{ij}^{(M,1,m)}\} \in \mathcal{R}^{N^{(m)}} \times \mathcal{R}^N$  とし , 変換後の結合行列はこれらに  $\hat{\phantom{x}}$  を付けて表すことにする . 以下に変換後の結合行列を示す .

$$\hat{\mathbf{T}}_f^{(m)} = \begin{bmatrix} 0_{aa} & 0_{ag} & 0_{ab} \\ 0_{ga} & \mathbf{T}_f^{(m)} & 0_{gb} \\ 0_{ba} & 0_{bg} & 0_{bb} \end{bmatrix} \in \mathcal{R}^N \times \mathcal{R}^N \quad (5.1)$$

$$\hat{\mathbf{w}}_f^{(2,n,m)} = \begin{cases} \begin{bmatrix} 0_{ac} & 0_{ah} & 0_{ae} & 0_{ag} & 0_{ab} \\ 0_{gc} & \mathbf{w}_{f1}^{(2,n,m)} & 0_{ge} & \mathbf{w}_{f2}^{(2,n,m)} & 0_{gb} \\ 0_{bc} & 0_{bh} & 0_{be} & 0_{bg} & 0_{bb} \end{bmatrix} & (n < m) \\ \begin{bmatrix} 0_{aa} & 0_{ag} & 0_{af} & 0_{ah} & 0_{ad} \\ 0_{ga} & \mathbf{w}_{f2}^{(2,n,m)} & 0_{gf} & \mathbf{w}_{f1}^{(2,n,m)} & 0_{gd} \\ 0_{ba} & 0_{bg} & 0_{bf} & 0_{bh} & 0_{bd} \end{bmatrix} & (n > m) \end{cases} \quad (5.2)$$

$$\hat{\mathbf{w}}_f^{(M,1,m)} = \begin{bmatrix} 0_{aN} \\ \mathbf{w}_f^{(M,1,m)} \\ 0_{bN} \end{bmatrix} \in \mathcal{R}^N \times \mathcal{R}^N \quad (5.3)$$

但し ,

$$\begin{aligned} a &= \varphi(m-1) & b &= N - \varphi(m) \\ c &= \varphi(n-1) & d &= N - \varphi(n) \\ e &= \varphi(m-1) - \varphi(n) & f &= \varphi(n-1) - \varphi(m) \\ g &= N^{(m)} & h &= N^{(n)} \end{aligned} \quad (5.4)$$

$$\varphi(k) = \sum_{m=1}^k N^{(m)}$$

ここで ,  $0_{xy}$  は  $x \times y$  零行列を表し , 結合行列の下付添字  $f$  は  $c \ll \varepsilon, \gamma$  の条件のもとで求められた式 (2.25) ~ (2.27) の解であることを表す . また , 式 (5.2) の

$\mathbf{w}_{f_1}^{(2,n,m)} \in \mathcal{R}^{N^{(m)}} \times \mathcal{R}^{N^{(n)}}$ ,  $\mathbf{w}_{f_2}^{(2,n,m)} \in \mathcal{R}^{N^{(m)}} \times \mathcal{R}^{N^{(m)}}$  は, それぞれ第  $n$  モジュールの状態を入力とする結合行列と第  $m$  モジュールの状態を入力とする結合行列に相当する. これらと結合行列  $\mathbf{w}^{(2,n,m)}$  との関係は次のように表される.

$$\mathbf{w}_f^{(2,n,m)} = (\mathbf{w}_{f_1}^{(2,n,m)} \mathbf{w}_{f_2}^{(2,n,m)}) \quad (5.5)$$

例 5.2  $N = 6, M = 3, N^{(m)} = 2, m = 2, n = 3$  のときの結合行列を  $N \times N$  行列で表してみる. まず, 式 (5.4) の各添字は以下ようになる.

$$\begin{aligned} a &= \varphi(1) = 2 & b &= 6 - \varphi(2) = 2 \\ c &= \varphi(2) = 4 & d &= N - \varphi(3) = 0 \\ e &= \varphi(1) - \varphi(3) = -4 & f &= \varphi(2) - \varphi(2) = 0 \\ g &= N^{(2)} = 2 & h &= N^{(3)} = 2 \end{aligned}$$

これから, 各結合行列は以下のように変換される.

$$\begin{aligned} \widehat{\mathbf{T}}_f^{(m)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & T_{11} & T_{12} & 0 & 0 \\ 0 & 0 & T_{21} & T_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ \widehat{\mathbf{w}}_f^{(2,n,m)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & w_{13} & w_{14} & w_{11} & w_{12} \\ 0 & 0 & w_{23} & w_{24} & w_{21} & w_{22} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ \widehat{\mathbf{w}}_f^{(M,1,m)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ w_{11} & w_{12} & w_{13} & w_{14} & w_{15} & w_{16} \\ w_{21} & w_{22} & w_{23} & w_{24} & w_{25} & w_{26} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

ここで、モジュールや相互作用のクラスを表す上付き記号は省略している。

上記のような行列表現を用いて、CCHN-LAGI/L( $M$ ) と等価な情報処理を行う自己想起型連想記憶モデルは、以下の  $N \times N$  行列  $C_f$  で与えられる (導出の詳細は付録 A.4を参照)。なお、モジュールへの外部入力は  $J = 0$  とし、エネルギー損失の項の影響は無視している。

$$C_f = \sum_{m=1}^M \left[ \alpha^{(m)} P^{(m)} + \sum_{\substack{n=1 \\ n \neq m}}^M \beta^{(2,n,m)} Q_1^{(2,n,m)'} Q_1^{(2,n,m)} + \beta^{(M,1,m)} Q_2^{(M,1,m)'} Q_2^{(M,1,m)} \right] \quad (5.6)$$

但し、

$$P^{(m)} = -\mathbf{H}^{(m)} \hat{\mathbf{T}}_f^{(m)} \mathbf{H}^{(m)} \quad (5.7)$$

$$Q_1^{(n,m)} = \mathbf{H}^{(m)} - \hat{\mathbf{w}}_f^{(2,n,m)} (\mathbf{H}^{(n)} + \mathbf{H}^{(m)}) \quad (5.8)$$

$$Q_2^{(M,1,m)} = \mathbf{H}^{(m)} - \hat{\mathbf{w}}_f^{(M,1,m)} \quad (5.9)$$

$$\mathbf{H}^{(m)} = \begin{bmatrix} 0_{aa} & 0_{ag} & 0_{ab} \\ 0_{ga} & \mathbf{I}_g & 0_{gb} \\ 0_{ba} & 0_{bg} & 0_{bb} \end{bmatrix} \in \mathcal{R}^N \times \mathcal{R}^N \quad (5.10)$$

ここで、 $\mathbf{I}_x$  は  $x \times x$  単位行列、 $'$  は行列の転置である。

式 (5.6) によって、如何なる構造パラメータをもつ CCHN-LAGI/L( $M$ ) に対しても、それと等価な情報処理を行う自己想起型連想記憶モデルに変換できる。5.2.2で行ったシミュレーション実験の中から、Table 5.2で表される  $M = 8$  のケースを取り上げて調べる。CCHN-LAGI/L(8) と CCHN-LI/L(8) に対し、式 (5.6) の  $C_f$  を求め

Table 5.2: Parameter values

number of modules ( $M$ )	8
dimension of memory pattern vectors ( $N$ )	512
number of memory patterns ( $P$ )	256 ( $r = 0.5$ )
number of hidden layers in internetworks ( $L$ )	0
class of interactions	$\mathcal{I}_2 + \mathcal{I}_M$
contribution of module ( $\alpha^{(m)}$ )	1.0
contribution of local interaction ( $\beta^{(2,n,m)}$ )	0.1
contribution of global interaction ( $\beta^{(M,1,m)}$ )	1.0

て解析を行った。

Fig.5.3(a) と Fig.5.4(a) は、横軸を固有値の番号、縦軸を固有値の大きさとした固有値の分布を表す。また、Fig.5.3(b) と Fig.5.4(b) は記憶パターンが各固有ベクトル方向にもつ空間成分  $\sigma_i$  の分布を表している。これは次式で求められる。

$$\sigma_i = \frac{1}{P} \sum_{\mu=1}^P (\xi^{\mu} q_i)^2 \quad (5.11)$$

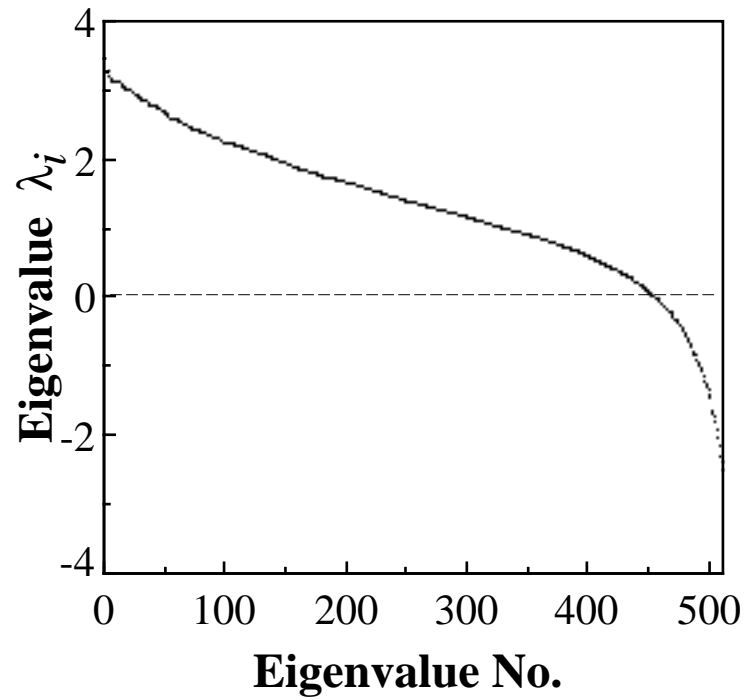
ここで、 $q_i$  は  $i$  番目の固有ベクトルである。

1 モジュールモデル (HN-COR) の場合には、モジュールの結合行列のランクからパターン空間とノイズ空間が明確に定義できた。しかしながら、マルチモジュールモデルの場合にはこれらは明確に定義できず、モジュールのユニット数やモジュール間結合の形態などに依存して変化する。そこで、 $\sigma_i$  の値を数値的に調べることでパターン空間とノイズ空間を区別する。いうまでもなく  $\sigma_i \neq 0$  となる固有ベクトルがパターン空間を構成する。

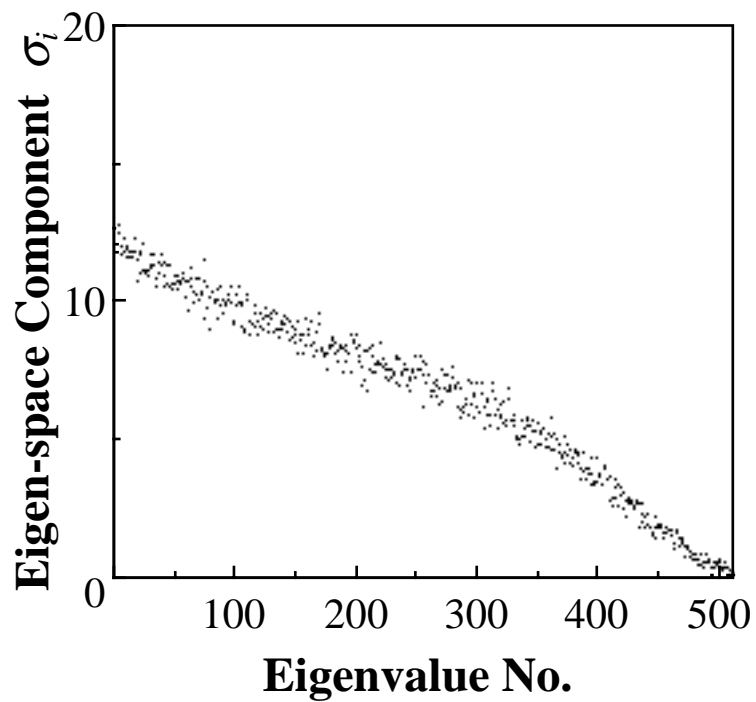
CCHN-LI/L(8) における固有値と空間成分の分布を Fig.5.3 に示す。Fig.5.3(b) より、すべての固有値に対して  $\sigma_i \neq 0$  となっており、CCHN-LI/L(8) のパターン空間がすべての固有ベクトルを用いて表されている。記憶パターン数が 256 であるので、HN-COR ではたかだか 256 個の固有ベクトルで表されるはずである。このことから、CCHN-LI/L(8) におけるパターン空間の表現はかなり冗長といえる。Fig.5.3(a) からわかるように、CCHN-LI(8) では最初から 450 個程度の固有値は正であるが、残りの 50 個程度は負となっている。3.2 における議論を思い出せば、正の固有値で表される空間成分は常に増加し、逆に負の固有値の空間成分は常に減少するのであった。このことから考えて、最後の 50 程度の固有値で表される部分空間においては記憶パターンの成分が含まれているにも関わらず、常にその成分を減少させるようなダイナミクスが働いている。これは正しいパターン想起を行う上で好ましくないダイナミクスである。CCHN-LI/L(8) がこのケースにおいて正しく想起できない理由は、このようなダイナミクスが生成されたためと考えられる。

これに対して CCHN-LAGI/L(8) では、Fig.5.4(b) からわかるように、最初の 380 個程度の固有値に対して空間成分  $\sigma_i$  が非零となり、残りは零になっている。このことは、CCHN-LAGI/L(8) においてノイズ空間が生成されていることを表す。また Figs.5.4(a)(b) から、固有値が正から負へ変る点と空間成分が非零から零へ変る点とがほぼ一致しているのがわかる。つまり、パターン空間においては常に空間成分を増加させるようなダイナミクスが形成され、ノイズ空間においては常に空間成分を減少させるようなダイナミクスが形成されている。この場合においても、HN-COR に比べれば CCHN-LAGI/L(8) のパターン空間は冗長といえるが、パター



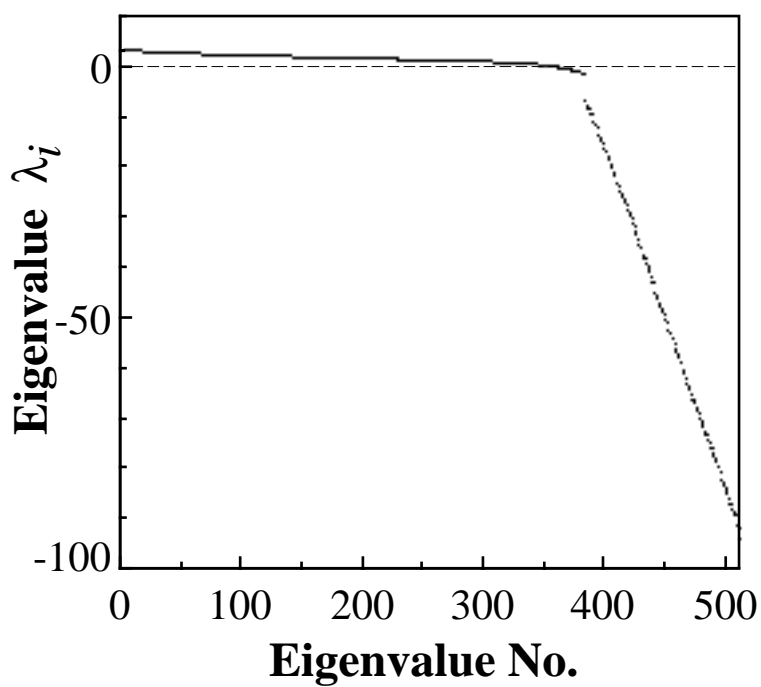


(a)

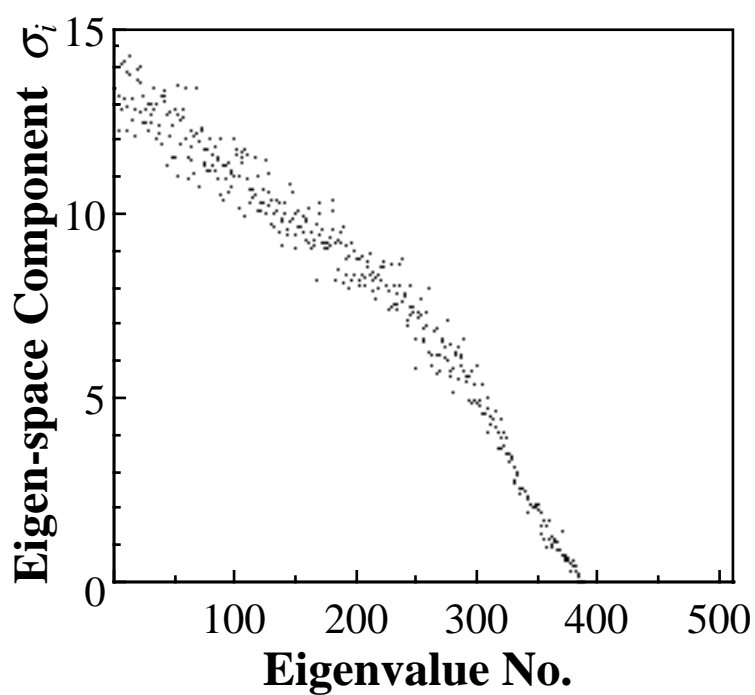


(b)

Figure 5.3: (a)The eigenvalues of the degenerated connection matrix and (b) the eigen-space component  $\sigma_i$  of stored total-patterns in CCHN-LI/(8).



(a)



(b)

Figure 5.4: (a) The eigenvalues of the degenerated connection matrix and (b) the eigen-space component  $\sigma_i$  of stored total-patterns in CCHN-LAGI/L(8).

ン空間とノイズ空間における両者の動特性はよく似ていることがわかる．よって，CCHN-LAGI/L(8) の連想特性が CCHN-LI/L(8) のそれに比べて格段に改善したのは，やはりノイズ空間の成分を単調減少するダイナミックスの存在にあるといえる．このようなダイナミックスの生成は明らかに一部のモジュールに導入された大域的相互作用の貢献によるものであり，この相互作用によって与えられたグローバル・コンテキストの情報が局所的相互作用を介してすべてのモジュールに伝搬されたことに起因している．

### 5.3 連想特性の多様性に対する考察

2.2で述べたように，マルチモジュール CCHN モデルは多数の構造パラメータをもち，それを変化させることで膨大な数の異なったマルチモジュールニューラルネットを実現することができる．前節で取り上げた CCHN-LAGI/L( $M$ ) では，導入する相互作用の種類や性質にいくらかの制限が加えられているものの，モジュール数や貢献度パラメータの値，モジュール間結合の形態など，様々なマルチモジュールニューラルネットを実現するだけの自由度を十分にもっている．よって CCHN-LAGI/L( $M$ ) に限ったとしても，それがもち得る動特性の多様性について十分な議論を行うことは容易でない．

そこで，本節では CCHN-LAGI/L( $M$ ) が表現できる以下の構造に限って，連想特性の多様性をシミュレーション実験により調べることにする．

1. 貢献度パラメータ
2. モジュール数
3. 導入される相互作用の種類

1. では，モジュールの情報処理と相互作用の貢献度が異なることで，どのような連想特性の変化がみられるのかを調べる．2. では，モジュール数が異なることによって生じる連想特性の変化について調べる．3. では，大域的相互作用と局所的相互作用のどちらかだけを導入した CCHN-GI/L( $M$ ) と CCHN-LI/L( $M$ ) の連想特性を調べ，CCHN-LAGI/L( $M$ ) との違いについて調べる．もちろん，これらを調べたとしてもマルチモジュール CCHN のもつ多様性のすべてについて議論できるわけではない．しかし本節の試みは，マルチモジュール CCHN が潜在的にもつ多様性の一端を実証することにつながる．

前節において，CCHN-LAGI/L( $M$ ) と等価な情報処理を行う自己想起型連想記憶モデルを導出した．本節におけるシミュレーション実験では，この等価モデルを用いて連想特性の多様性を検証する．これは本節の議論を行う上で必要不可欠なことではなく，想起シミュレーションにかかる計算コストを減らす目的で行われる．一旦，この等価モデルの結合行列を求めてしまえば，モジュール構造に依存せず一定の計算量で想起シミュレーションが可能となる．

注意 5.1 等価モデルによる想起シミュレーションが可能なのは，CCHN-LAGI/L( $M$ ) における高速な情報検索やハードウェア化を実現する意味で重要である．また，これまでの自己想起型連想記憶の研究では，多様な連想特性を実現するための努力はあまり行われておらず，連想特性は記憶パターン数でほぼ決まってしまう [68]．しかしながら序論で述べたように，ニューラルネットを用いた連想記憶モデルに要求されるのは，本来，柔軟な適応・学習能力や情報検索の柔軟性であるはずである．この意味においても，ここでの等価モデルは非常に興味深い試みであるといえる．

### 5.3.1 記憶パターンとシミュレーション方法

記憶パターンには要素が  $\pm 1$  の 2 値で表される 100 次元ベクトルを用い，これらは 3.3 で扱ったようにクラスター状に分布しているものとする．

同一クラスターに属する記憶パターンが 4 個として，以下の 2 種類の方法で生成した．

- I. Fig.5.5に示すように，記憶パターンを 4 つの 25 次元部分パターンに分け， $p$  番目の記憶パターンにおける  $k$  番目の部分パターンを  $\xi_{pk}$  ( $p, k = 1, \dots, 4$ ) とする．このとき，以下の関係が成り立つようにクラスター内の記憶パターンを決

	k=1	k=2	k=3	k=4
p=1	$\xi_{11}$	$\xi_{12}$	$\xi_{13}$	$\xi_{14} = z$
p=2	$\xi_{21}$	$\xi_{22}$	$\xi_{23} = y$	$\xi_{24} = z$
p=3	$\xi_{31}$	$\xi_{32} = x$	$\xi_{33} = y$	$\xi_{34} = z$
p=4	$\xi_{41}$	$\xi_{42} = x$	$\xi_{43} = y$	$\xi_{44} = z$

Figure 5.5: Construction of memory patterns (I)

定する .

$$\begin{aligned}\xi_{32} &= \xi_{42} \stackrel{\text{def}}{=} x \\ \xi_{23} &= \xi_{33} = \xi_{43} \stackrel{\text{def}}{=} y \\ \xi_{14} &= \xi_{24} = \xi_{34} = \xi_{44} \stackrel{\text{def}}{=} z\end{aligned}$$

ここで,  $x, y, z$  はランダムに選ばれた 25 次元部分パターンであり, 上記以外の部分パターンもランダムに選ばれるものとする . このとき, 記憶パターン間の平均相関は  $\bar{\rho} \simeq 0.42$  となる . このような記憶パターンの特徴は, パターン間の重複が特定部分に集中している点にある .

II. 記憶パターン間の相関が 0.42 程度となるよう, 重複ビットをランダムに選択する . このような記憶パターンでは, パターン間の重複が分散化される .

なお, クラスタ中心の相関は約 0.1 とした .

3.3 で述べたように, クラスタ構造を決定するパラメータは以下の通りである .

- クラスタ数 :  $C$
- 1 クラスタ当たりのパターン数 :  $L$
- 同一クラスタ内にあるパターン間の平均相関 :  $\bar{\sigma}$
- クラスタ・セントロイド間の平均相関 :  $\bar{\kappa}$

上述した記憶パターンの生成方法から, 上記パラメータのうちクラスタ数以外は以下のように固定される .

$$L = 4, \quad \bar{\sigma} \simeq 0.42, \quad \bar{\kappa} \simeq 0.1$$

これら記憶パターンから式 (5.6) に基づいて結合行列が生成され, 想起テストが行われる . ここで, ネットワークの初期状態となるテストパターン  $v(0)$  の総数は 13,200 で, 第 1 クラスタを構成する 4 つの記憶パターンにランダムなノイズを加えて生成している . このときに加えるノイズの大きさは, 初期状態と記憶パターンとの方向余弦  $d^\mu$  が 0.15 ~ 1.0 の範囲で一様分布するようランダムに決めた . なお, 比較実験を行うすべての連想記憶モデルに対して同一のテストパターンセットを与えるものとする . テストパターンセットは記憶パターンに対するおおよその近傍を形成することになるため, これらをネットワークの初期状態とし, それぞれについて

想起が成功するかどうかを調べることで引き込み領域を調べることができる。ただし、同一クラスター内の記憶パターンは相互に強い相関をもっているため、一般に引き込み領域は記憶パターンを中心とした超球状にはなりにくい点に注意する。すなわち、同じ方向余弦をもつテストパターンであっても、想起に成功するかどうかは記憶パターンとの位置関係に依存する。そこで引き込み領域の大きさを表す指標として、テストパターン数に対する想起の成功回数の割合を用いることにする。ここでは、これを想起率 (successful recall rate) と呼び、 $a$  で表すことにする。

注意 5.2 これまでのシミュレーション実験においては、p.49の定義 3.4の限界方向余弦を調べることで連想性能の評価を行ってきた。一般に、引き込み領域が超球状になるほど、この限界方向余弦を用いて、より正確な引き込み領域の大きさを知ることができる。

$a$  はある一定領域に占める引き込み領域のおおよその割合を指標化したものと考えられ、 $a$  が大きいほど引き込み領域が大きいといえる。なお、テストするパターンセットによって  $a$  の値の意味が異なることに注意する。つまり、生成したすべてのテストパターンから  $a$  を求めた場合、この  $a$  は第1クラスターの4つの記憶パターンに対する平均的な引き込み領域の大きさを表すことになる。一方、特定の記憶パターン近傍のテストパターンから  $a$  を求めた場合、その記憶パターン近傍の引き込み領域の大きさを表すことになる。

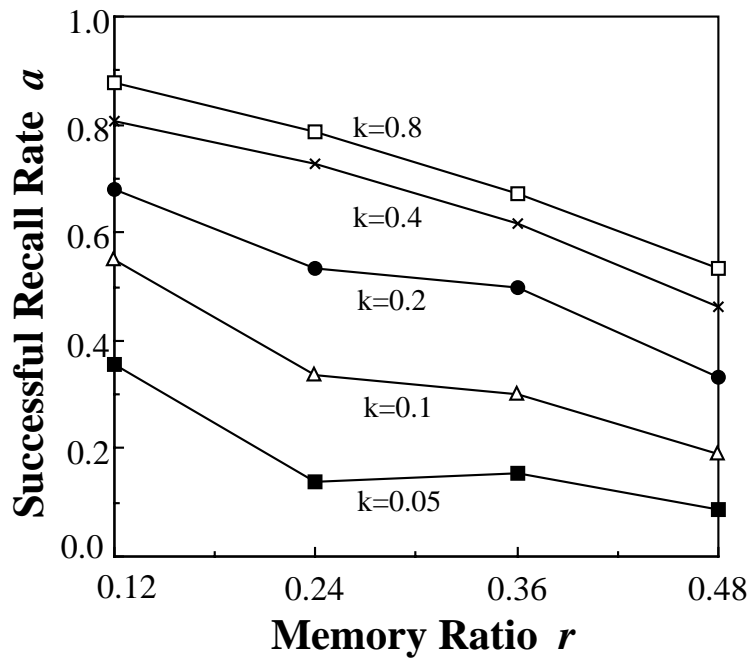
なお、これまでのシミュレーション実験では引き込み領域が大きいものほど優れた連想特性をもっているとしたが、本節ではどの程度多様な連想特性 (具体的には引き込み特性) を実現できるかが評価の対象となる。

### 5.3.2 連想特性の多様性に対する考察

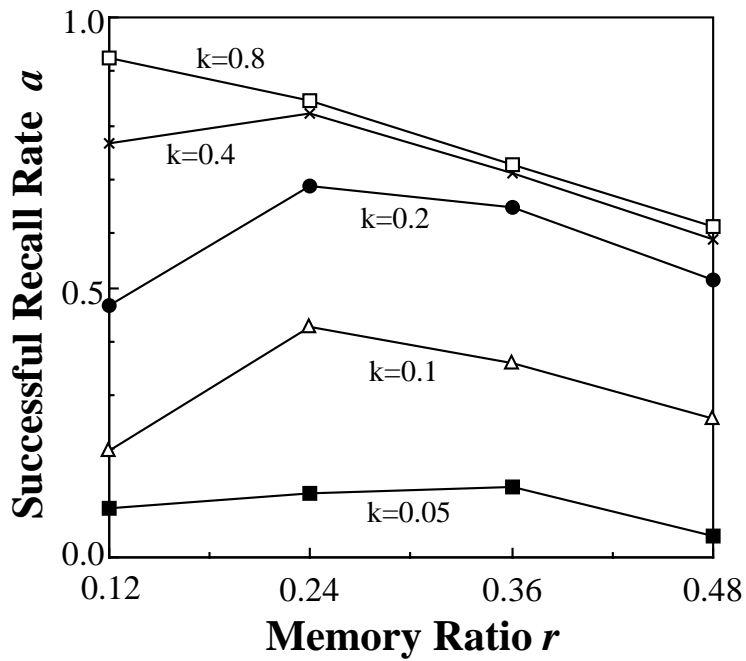
#### (a) 貢献度の違いによる連想特性の変化

CCHN-LAGI/L( $M$ ) における情報処理はモジュールの情報処理とモジュール間相互作用に分けられ、これらのネットワーク全体に対する貢献度は  $\alpha^{(m)}$ ,  $\beta^{(2,n,m)}$ ,  $\beta^{(M,1,m)}$  で決まる。

以下では、2モジュールモデルである CCHN-GI/L(2) の貢献度パラメータを  $\alpha^{(m)} = 1.0$ ,  $\beta^{(M,1,m)} \stackrel{\text{def}}{=} k$  とし、モジュールの情報処理に対する相互作用の相対的な貢献度の影響を調べる。Table 5.3にシミュレーション実験で用いたパラメータの値をまとめておく。



(a) Memory Pattern I



(b) Memory Pattern II

Figure 5.6: Association performance for different  $k$

Table 5.3: Parameter values

number of modules ( $M$ )	2
dimension of memory pattern vectors ( $N$ )	100
number of memory patterns ( $P$ )	12, 24, 36, 48
number of clusters ( $C$ )	3, 6, 9, 12
number of memory patterns in each cluster ( $S$ )	4
average correlation between patterns ( $\bar{\sigma}$ )	0.42
average correlation between cluster centroids ( $\bar{\kappa}$ )	0.1
number of hidden layers in internetworks ( $L$ )	0
class of interactions	$\mathcal{I}_M$
contribution of module ( $\alpha^{(m)}$ )	1.0
contribution of global interaction ( $\beta^{(M,1,m)}$ )	0.05, 0.1, 0.2, 0.4, 0.8

Figs.5.6(a)(b) は、記憶パターンセット (I)(II) に対する想起率  $a$  の変化を調べた結果である。なお、この  $a$  は第1クラスタに属する4つの記憶パターン近傍での平均的な引き込み領域の大きさを表す。横軸の記憶率  $r$  は p.48の定義 3.3に基づき  $P/100$  で与えられる。

記憶パターンセット (I)(II) で連想特性の変化の様子が異なるが、全体的に  $k$  が小さいほど引き込み領域は小さくなっている。これは  $k=0$  のとき、CCHN-LAGI/L(2) が2個の独立した相関モデル (HN-COR) と等しくなることを考えれば容易に理解できるであろう。このように  $k$  の値を変えるだけで様々な連想特性を実現でき、ここには示していないが、この特性はモジュール数の異なる CCHN-LAGI/L や CCHN-LI/L, CCHN-GI/L についても同様のことがいえる。但し、 $k$  と連想特性の劣化の関係が非線形であることや、これが記憶パターンの性質にも依存するため、要求された連想特性から適切な貢献度パラメータを選択するには、ある程度の試行錯誤を必要とする。

#### (b) モジュール数の違いによる連想特性の変化

モジュール数の異なる CCHN-LI/L( $M$ ) に対し、同一クラスタ内にある4つの記憶パターン近傍の平均的な連想特性についてまず調べる。Table 5.4にシミュレーション実験で用いたパラメータの値をまとめておく。記憶パターンセット (I)(II) に対す



Table 5.4: Parameter values

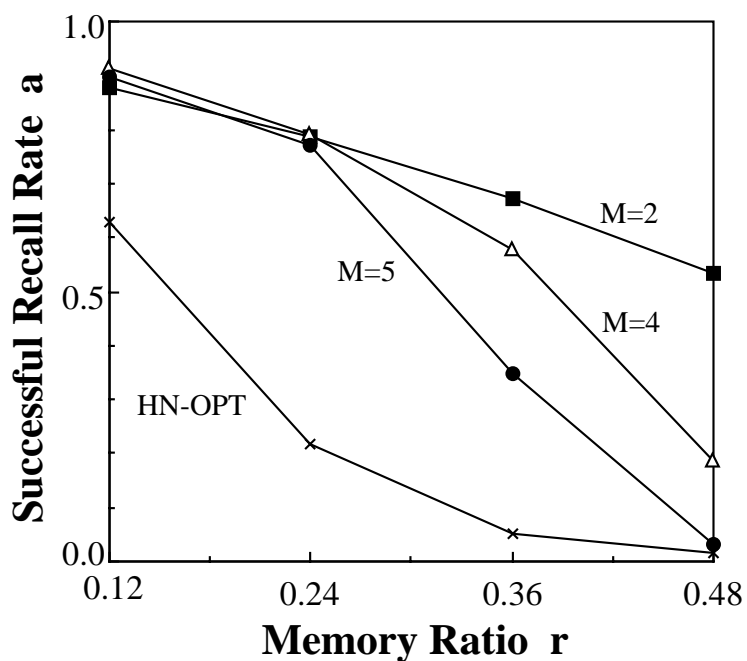
number of modules ( $M$ )	2, 4, 5
dimension of memory pattern vectors ( $N$ )	100
number of memory patterns ( $P$ )	12, 24, 36, 48
number of clusters ( $C$ )	3,6,9,12
number of memory patterns in each cluster ( $S$ )	4
average correlation between patterns ( $\bar{\sigma}$ )	0.42
average correlation between cluster centroids ( $\bar{\kappa}$ )	0.1
number of hidden layers in internetworks ( $L$ )	0
class of interactions	$\mathcal{I}_2$
contribution of module ( $\alpha^{(m)}$ )	1.0
contribution of local interaction ( $\beta^{(2,n,m)}$ )	0.8

る想起率  $a$  の変化を Figs.5.7(a)(b) にそれぞれ示す．なお，この  $a$  は第1クラスタに属する記憶パターン近傍の平均的な引き込み領域の大きさを表す．最適写像モデル (HN-OPT) の結果も，比較のため Fig.5.7 に示す．

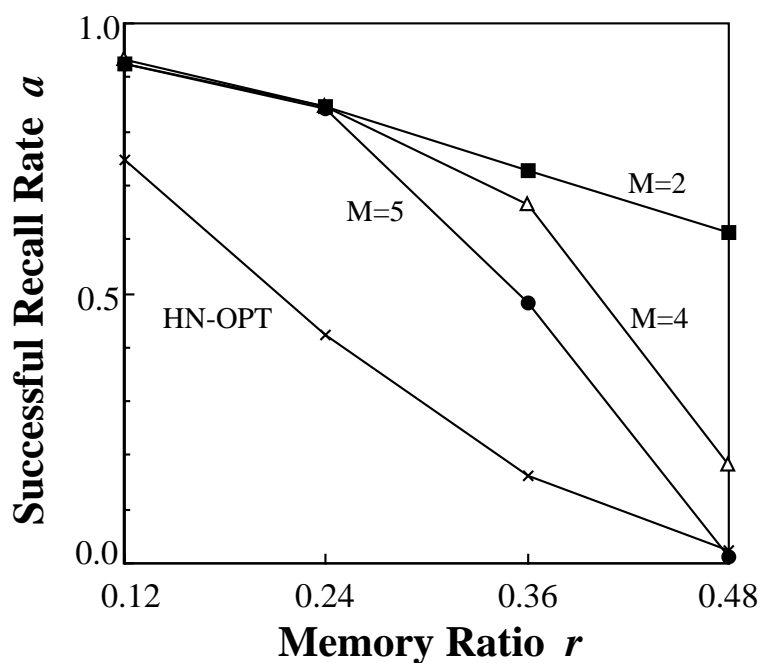
Fig.5.7からわかるように， $r \leq 0.24$  においては2種類の記憶パターンセットに対して，モジュール数の違いによる連想特性の違いはあまり生じていない．それに対し， $r > 0.24$  についてはどちらのパターンセットとも，モジュール数が多くなるに従って引き込み領域が小さくなっていく．これはインターネットワークとしてクラス  $\mathcal{I}_2$  のみが導入されており，前節で考察したように  $M$  が大きくなるとモジュールがグローバル・コンテキストの情報を得にくくなるためである．

以上からわかるように，記憶率  $r$  がある程度大きければ CCHN-LI/L のモジュール数  $M$  を変更することで，特性の異なる連想記憶装置を実現できる．また前項の結果から予想されるように， $M$  だけでなく貢献度パラメータを同時に変更すれば，より柔軟に連想特性を変化させることができる．

次に，同一クラスタ内の記憶パターン近傍における連想特性をそれぞれ個別に調べてみる． $r=0.24$  としたとき，第1クラスタの4つの記憶パターン近傍における想起率  $a$  を個別に調べた．シミュレーション結果を Fig.5.8に示す．Figs.5.8(a)(b) はそれぞれ記憶パターンセット (I)(II) に対する結果を表す．Fig.5.8(b) でわかるように，重複箇所がランダムに分散している記憶パターンセット (II) では，モジュール数の



(a) Memory Pattern I



(b) Memory Pattern II

Figure 5.7: Association performance for different  $M$

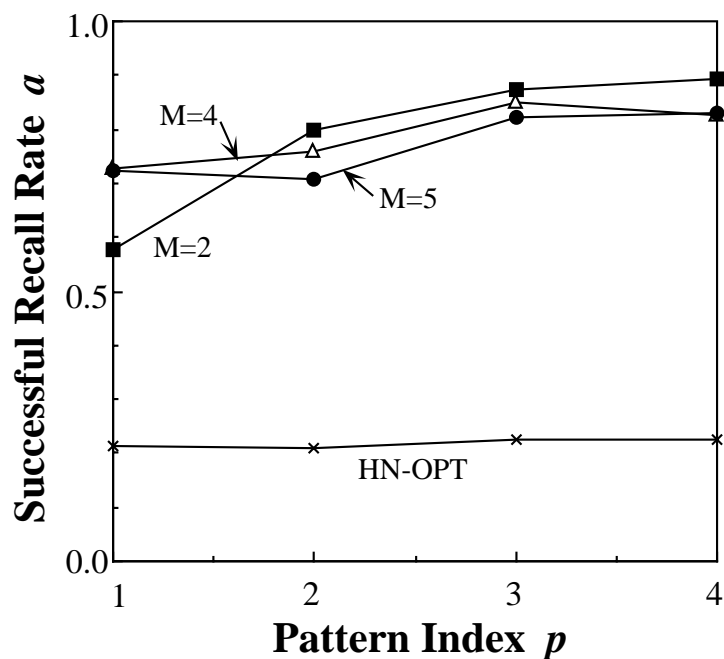
違いによる連想特性の変化はほとんどない．これに対し，重複箇所が局在化した記憶パターンセット (I) では，モジュール数によってパターン近傍の連想特性が異なっている． $M = 2$  の場合，1 番目の記憶パターンの引き込み領域が最も小さく，3,4 番目の記憶パターンの引き込み領域が最も大きくなっている．これに比べて  $M = 4$  のとき，記憶パターンに対する連想特性のばらつきは比較的少ないのがわかる．

これを説明するため，Fig.5.9を使って各モジュールに割り当てられる部分パターンについて考えてみる． $M = 2$  の場合，第1モジュールに割り当てられる4種類の部分パターン  $(\xi'_{p1} \xi'_{p2})'$  ( $p = 1, \dots, 4$ ) のうち， $p = 3, 4$  の部分パターン間の相関は約 0.5 となる．他の部分パターンは互いにほぼ無相関となるため，このような場合，第1モジュールでは上記2つの部分パターンへの引き込みが強くなると考えられる [64][69]．つまり，第1モジュールの情報処理だけを見ると，3番目と4番目の記憶パターンが想起しやすくなると考えられる．第2モジュールには2種類の部分パターン  $(\xi'_{13} z')$  と  $(y' z')$  のみが記憶されており，それらの相関は約 0.5 となる．ただし，後者の部分パターンは3つの記憶パターン ( $\mu = 2, \dots, 4$ ) で重複しているため (Fig.5.9(a) の斜線部分)，結合荷重には前者の部分パターンに比べて3倍の強さで記憶されている．この結果，第2モジュールでは2~4番目の記憶パターンが想起しやすくなる．以上のことを考えると， $M = 2$  の場合，3,4番目の記憶パターンを最も想起しやすく，次に2番目の記憶パターンを想起しやすくなると考えられる．

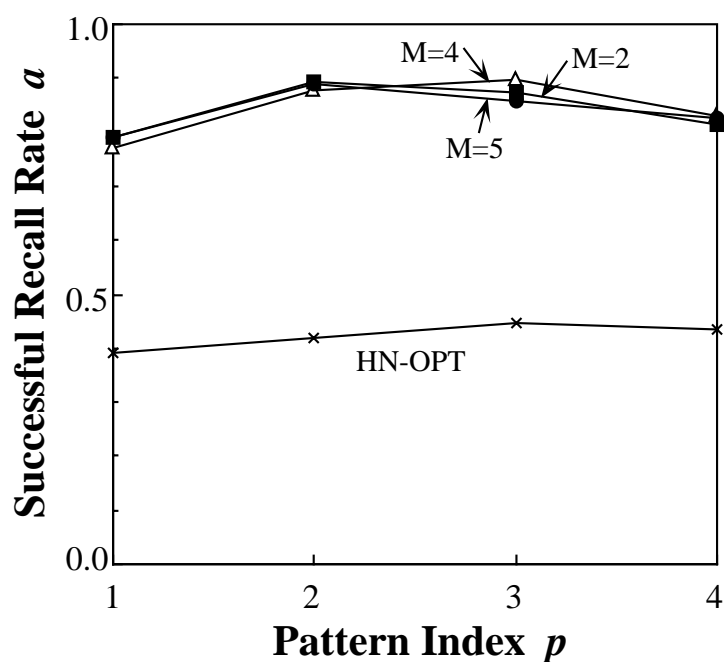
一方， $M = 4$  の場合， $M = 2$  のときと同様に複数の記憶パターンに重複する部分パターンが存在するが (Fig.5.9(b) の網掛け部分)，各モジュールで記憶される部分ベクトルが互いにほぼ無相関であるため，特定の部分パターンへの引き込みが強くなり過ぎることはない．これらの結果より，記憶パターン近傍の引き込み特性は，その記憶パターンがモジュールにどのように割り当てられるかに依存することがわかる．

以上，特定の記憶パターン近傍における引き込み特性がモジュール数  $M$  に影響を受ける理由について考察したが，厳密にはモジュールの情報処理だけでなくモジュール間相互作用も考慮しなければならない．これにはインターネットワークの写像特性を解析する必要があるが，これは一般に難しい．しかしながら，上記の考察結果は Fig.5.8(a) の結果と矛盾しているわけではないので，相互作用の連想特性への影響はモジュールの情報処理のそれと大きく異なるものと考えられる．

なお，参考のため HN-OPT の結果も Fig.5.8に示している．HN-OPT の結合行列は記憶パターンを直交化して記憶するため，各記憶パターン近傍の引き込み領域はほぼ等しくなる．しかし，その引き込み領域はそれほど小さくなく，またその大きさを変更することもできない．



(a) Memory Pattern I



(b) Memory Pattern II

Figure 5.8: Association performance in the neighborhood of different memory patterns ( $r=0.24$ )

	m=1		m=2	
p=1	$\xi_{11}$	$\xi_{12}$	$\xi_{13}$	$\xi_{14} = z$
p=2	$\xi_{21}$	$\xi_{22}$	$\xi_{23} = y$	$\xi_{24} = z$
p=3	$\xi_{31}$	$\xi_{32} = x$	$\xi_{33} = y$	$\xi_{34} = z$
p=4	$\xi_{41}$	$\xi_{42} = x$	$\xi_{43} = y$	$\xi_{44} = z$

(a) M=2

	m=1	m=2	m=3	m=4
p=1	$\xi_{11}$	$\xi_{12}$	$\xi_{13}$	$\xi_{14} = z$
p=2	$\xi_{21}$	$\xi_{22}$	$\xi_{23} = y$	$\xi_{24} = z$
p=3	$\xi_{31}$	$\xi_{32} = x$	$\xi_{33} = y$	$\xi_{34} = z$
p=4	$\xi_{41}$	$\xi_{42} = x$	$\xi_{43} = y$	$\xi_{44} = z$

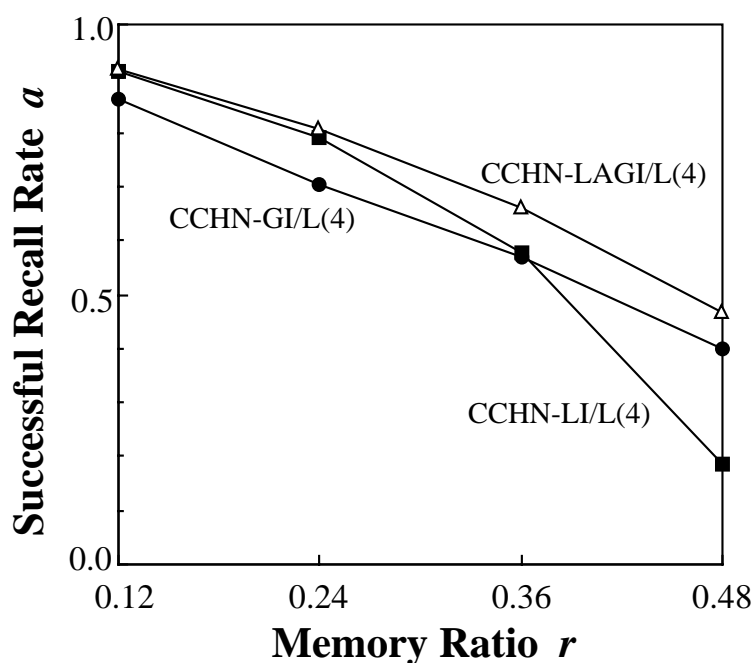
(b) M=4

Figure 5.9: Allocation of memory patterns to module networks ( $p$ : pattern index,  $m$ : module index)

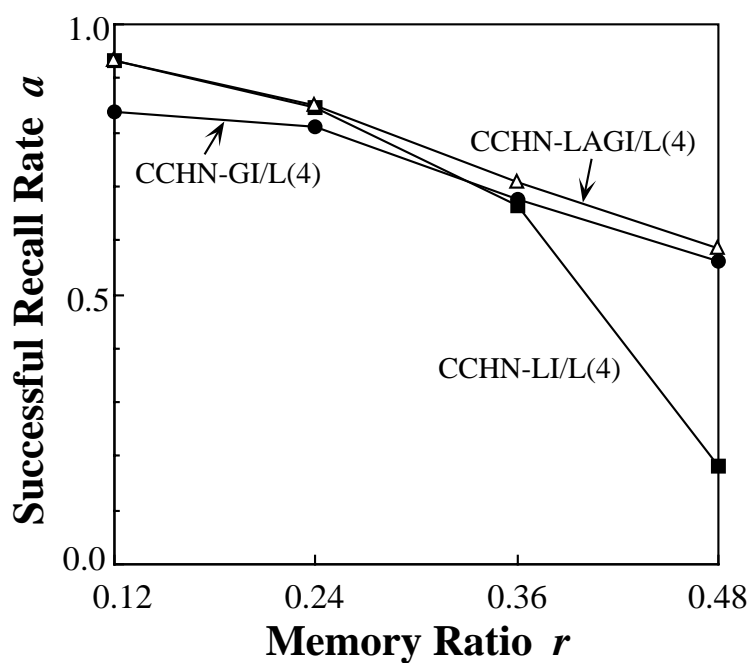
### (c) 相互作用の違いによる連想特性の変化

導入される相互作用の違いによる連想特性の変化を調べるため、 $M = 4$  とした CCHN-LAGI/L(4) と CCHN-LI/L(4), CCHN-GI/L(4) の 3 つのモデルを取り上げる。Table 5.5 にシミュレーション実験で用いたパラメータの値をまとめておく。

Figs.5.10(a)(b) は、記憶パターンセット (I)(II) に対する 3 つのモデルの想起率  $a$  を調べた結果である。なお、この  $a$  は第 1 クラスタに属する記憶パターン近傍の平均的な引き込み領域の大きさを表す。Figs.5.10(a)(b) より、CCHN-LAGI/L(4) および CCHN-GI/L(4) は CCHN-LI/L(4) に比べて記憶率  $r$  が大きくなっても急激な連想特性の劣化はみられない。また、CCHN-LAGI/L(4) と CCHN-GI/L(4) の  $r$  に対する劣化傾向は類似しているが、CCHN-LAGI/L(4) の方が引き込み領域が大きい。このことは、大局的相互作用が記憶率の増加に対する連想特性の急激な劣化を抑える性質をもち、局所的相互作用との協調によってさらに引き込み領域が拡大されることを示している。よって、平均的な連想特性を最大化するには CCHN-LAGI/L が適している。



(a) Memory Pattern I



(b) Memory Pattern II

Figure 5.10: Association performance for different types of introduced interaction

Table 5.5: Parameter values

number of modules ( $M$ )	4
dimension of memory pattern vectors ( $N$ )	100
number of memory patterns ( $P$ )	12, 24, 36, 48
number of clusters ( $C$ )	3,6,9,12
number of memory patterns in each cluster ( $S$ )	4
average correlation between patterns ( $\bar{\sigma}$ )	0.42
average correlation between cluster centroids ( $\bar{\kappa}$ )	0.1
number of hidden layers in internetworks ( $L$ )	0
class of interactions	$\mathcal{I}_2, \mathcal{I}_M, \mathcal{I}_2 + \mathcal{I}_M$
contribution of module ( $\alpha^{(m)}$ )	1.0
contribution of local interaction ( $\beta^{(2,n,m)}$ )	0.8
contribution of local interaction ( $\beta^{(M,1,m)}$ )	0.4

## 5.4 まとめ

本章では，マルチモジュール CCHN において表現可能な膨大なバリエーションの中から，相互作用のクラスを  $\mathcal{I}_2$  と  $\mathcal{I}_M$  に限定したマルチモジュール CCHN-LAGI を取り上げた．クラス  $\mathcal{I}_2$  の局所的相互作用には自己参照型のものが用いられる．相互作用が 2 種類に限定されたのは，以下の観点からマルチモジュール CCHN の動特性を考察するためである．

- 異種の相互作用が同時に導入されることの効果

上記に対する考察は，モジュールの情報処理と相互作用を一つの自己想起型連想記憶モデル (HN) の情報処理として表現し，この結合行列の固有値解析を通して行われた．そこで，マルチモジュール CCHN-LAGI と等価な情報処理を行う HN の結合行列を導いた．但し，インターネットワークは 2 層の階層型ニューラルネットとし，ユニットの出力関数には線形なものを用いた．このようなマルチモジュールモデルは CCHN-LAGI/L( $M$ ) と表される．

導入された 2 種類の相互作用に対して以下の条件を仮定し，CCHN-LAGI/L( $M$ ) の動特性について考察を行った．

- クラス  $\mathcal{I}_2$ ：考えられる全ての 2 モジュール間に自己参照型相互作用を導入する．

クラス  $\mathcal{I}_M$  :  $M/4$  個のモジュールにだけ大局的相互作用 (クラス  $\mathcal{I}_M$ ) を導入する .  
 このような CCHN-LAGI/L( $M$ ) に対して , まずランダムパターンを記憶させて連想性能を調べた . また比較のため , クラス  $\mathcal{I}_2$  の局所的相互作用のみが導入された CCHN-LI/L( $M$ ) および最適連想写像モデル (HN-OPT) についても調べられた . シミュレーション実験の結果を以下にまとめる .

1. 大局的相互作用をもたない CCHN-LI/L( $M$ ) ではモジュール数が増えると急速に連想性能が劣化していく . これに対し , CCHN-LAGI/L( $M$ ) ではこの傾向がかなり改善される .
2. モジュール数があまり多くなければ , CCHN-LAGI/L( $M$ ) は HN-OPT と比べて同等以上の優れた性能をもつ .

また ,  $M = 8$  とした CCHN-LAGI/L(8) および CCHN-LI/L(8) を等価モデルに変換し , これらの結合行列を固有値解析することで動特性の考察を行った . この結果 , 以下の特徴が確認された (但し , CCHN-LI/L(8) が連想記憶として全く機能しないケースを取り上げている) .

- I. CCHN-LI/L( $M$ ) におけるパターン空間はすべての固有ベクトルを用いた冗長な表現になっている . さらに , パターン空間を構成する一部の固有ベクトルの固有値が負となり , パターン空間成分が減少するような望ましくないダイナミックスが生成されてしまう .
- II. CCHN-LAGI/L(8) におけるパターン空間も冗長な表現となっているが , その冗長度は CCHN-LI/L(8) に比べて小さい . この結果として , 負の固有値をもったノイズ空間が生じ , ノイズ空間成分を単調に減少させるダイナミックスが発生する .

連想性能を調べた結果と上記の考察結果から , マルチモジュールモデルの連想性能を高めるにはパターン空間の冗長度をなるべく小さくし , ノイズ空間ダイナミックスを生じさせることが重要であるといえる . また , CCHN-LAGI/L( $M$ ) においてノイズ空間ダイナミックスが生成されたのは , 明らかに一部のモジュールに導入された大域的相互作用による . 本章で取り上げたケースでは ,  $3/4$  のモジュールがグローバル・コンテキストの情報を直接的には与えられない . これにも関わらず連想性能が飛躍的に改善されたのは , 一部のモジュールに与えられたグローバル・コンテキストの情報が局所的相互作用を介してすべてのモジュールに伝搬されたためと



考えられる．このことは，2種類の相互作用が協調することで生み出される動特性が，連想特性の改善に寄与したことを示唆している．

次に，上記の2つのモデルにおいてパターン空間が冗長になる理由について考えてみる．局所的相互作用はモジュールの最終状態に定義されたローカル・コンテキストを満足するように作用する．よって局所的相互作用によって満足されようとしているローカル・コンテキストのすべてが，一致した一つのグローバル・コンテキスト(ネットワーク全体の望ましい最終状態)を満足しなければならない．しかしながら，局所的相互作用はモジュール数を  $M$  とすると  $M(M-1)/2$  だけ導入され，一つのグローバル・コンテキストに一致する確率はモジュール数が増えるとともに低下すると考えられる．また，局所的相互作用はモジュール数が増えるほどインターネットで参照される入力情報の局所性が高まり，これもローカル・コンテキストがコンフリクトを起こす確率を高くする．つまり，モジュール間に生じる競合関係が解消できなくなる確率が高くなり，その結果としてグローバル・コンテキストを満足しない最終状態で落ち着く確率が高くなる．つまり偽の最終状態(連想記憶という偽記憶)が生成されることになり，これが記憶空間を冗長にしているものと考えられる．

これまで(優れた連想性能) = (大きな引き込み領域) という観点からマルチモジュールモデルの評価を行ってきたが，本章の後半では，(優れた連想性能) = (多様な引き込み特性の実現) という観点で連想性能を評価した．CCHN-LAGI/L( $M$ ) がもつ構造パラメータのうち，以下のパラメータを変化させ，クラスタ状パターンに対する連想性能を調べた．

- (1) 貢献度パラメータ    (2) モジュール数    (3) 導入される相互作用の種類

以下に，この結果をまとめる．

- (i) モジュールの情報処理と相互作用の貢献度パラメータを変更することで，平均的な引き込み特性が記憶パターン数にかかわらず変化する．モジュールの貢献度を固定した場合，一般に，相互作用の貢献度が小さくなるほど引き込み領域は小さくなる．これは，3.3で考察した1モジュールモデルにおける結果と一致する．
- (ii) 局所的相互作用しか導入されない CCHN-LI/L( $M$ ) では，記憶パターン数がある程度多いとき，モジュール数によって平均的な引き込み特性が変化する．

- (iii) 記憶パターン間の重複が分散して存在する場合，モジュール数の変化による連想特性への影響はあまりない．しかし記憶パターン間の重複が局在する場合，CCHN-LI/L(M) ではモジュール数の変化によって各記憶パターン近傍の引き込み特性が異なる．この連想特性は，局在する記憶パターン間の重複がどのように各モジュールに割り当てられるかに依存する．
- (iv) 導入される相互作用の種類によっても引き込み特性は変化する．大局的相互作用は記憶率の増加に対する連想特性の急激な劣化を抑える性質をもち，局所的相互作用との協調によってさらに引き込み領域が拡大される．よって，平均的な引き込み領域は CCHN-LAGI/L(M) を用いることにより最大化される．

以上から，貢献度パラメータやモジュール数，導入する相互作用の種類を変更することで，様々な引き込み特性がマルチモジュールモデルにおいて実現できるといえる．平均的な引き込み特性を変更するのであれば，貢献度パラメータを使えばある程度のコントロールは可能といえる．しかしながら，例えば記憶パターンごとに異なった引き込み特性をもたせることは，ここで用いた構造パラメータの範囲ではかなり困難といえる．パターンの重複が局在している記憶パターン (I) を用いた場合，モジュール数を変更することで，ある特定の記憶パターンの引き込み領域が広がった．しかし，これはモジュール構造とパターン構造がうまく合致したために生じた稀なケースといえ，一般的なパターン構造に対して，このような引き込み特性を実現するパラメータは今のところ見つかっていない．記憶パターンごとに引き込み特性を変化させるには，モジュール間結合の形態を変更したり，相互作用に非線形性を導入したりするなどの試みが必要と思われる．

## 第6章 結 論

本論文では，複数個の動的な情報処理系で構成された動的マルチモジュールニューラルネットのモデル化手法を提案し，このモデルで実現される動特性の考察および連想記憶モデルとしての評価を行った．

まず第2章において，モジュールの情報処理と相互作用の機能について定義し，多様な相互作用を導入するためのクラスを定義した．そして，動的マルチモジュールニューラルネットの情報処理をエネルギー関数によって記述する手法を提案し，その動作方程式の導出方法について述べた．エネルギー関数で情報処理を記述する方法は，ネットワークに漸近安定な特性をもたせたいケースにしか適用できないが，以下の点で従来モデルにはない特徴をもっている．

1. ネットワーク構造や各部位の情報処理様式などに依存しないモデル化が可能である．これによって，任意個のモジュールで構成されたネットワークや多種多様な相互作用が導入されたものが簡単に構築できる．
2. ネットワーク構造に依存せず，試行錯誤なしで適切な動作方程式 (ユニットの更新則や結合の学習則) を導出できる．
3. ネットワークの動特性がエネルギー関数の幾何学的特性として表されることになり，動特性の考察が容易になる．

次に，相互作用のクラスが2種類導入されたケースにおいて，エネルギー関数や動作方程式を実際に求めた．また，活性ダイナミクスと結合ダイナミクスの時定数に対してある関係を仮定すると，本モデルにおける学習の取り扱いが容易になることを示した．

第3章から第5章にかけては，上記の動的マルチモジュールニューラルネットの動特性に関する考察および連想記憶モデルとしての評価が行われた．これらに関する議論は，マルチモジュールニューラルネットとしての本モデルの特徴が明らかに

なるよう、構造の単純なもの(1モジュールモデル)から複雑なもの(マルチモジュールモデル)へと順に議論を展開していった。各章で得られた結果を以下にまとめる。

第3章では、ホップフィールドネットのエネルギー関数に誤差エネルギーを付加した1モジュールモデルを取り上げた。この1モジュールモデルでは、モジュールに従来型のホップフィールドネットを用い、これを自己結合するインターネットワーク(相互作用を導入するために用いられる階層型ニューラルネット)が1つ導入されている。インターネットワークに2層の階層型ニューラルネットを用い、ユニットの出力関数を線形にした場合、この1モジュールモデルと等価な情報処理を行う自己想起型連想記憶モデルが導出されることを示した。さらに、この等価モデルの結合行列に対して固有値解析を行うことで、1モジュールの動特性を考察する手法を提案した。この手法を用いることで、相互作用が導入されたことによるダイナミックスの変化について考察を行った。この結果、以下の動特性が明らかになった。

- I. 従来型の自己想起型連想記憶モデルにおける動特性は記憶パターンによって張られる「パターン空間」とその直交補空間である「ノイズ空間」に分類される。この従来型モデルに相互作用を導入した1モジュールモデルの動特性も、上記と同じ2つの部分空間で特徴付けられる。
- II. 相互作用が導入されてもパターン空間のダイナミックスは変化しない。つまり、1モジュールモデルと従来型モデルがもつ動特性の違いは、ノイズ空間において明確になる。
- III. 1モジュールモデルにおけるノイズ空間ダイナミックスは、記憶パターン成分以外のノイズ成分を単調に減少させる。このことは、本論文で定義した相互作用の情報処理機能が正しくネットワークの動特性として反映されていることを示している。
- IV. パターン空間ダイナミックスに対してノイズ空間ダイナミックスを優位にしたとき、従来モデルと比べて、飛躍的に連想性能が向上する。また、貢献度パラメータを調整することで、引き込み特性を変化させることができる。

第4章では、マルチモジュールニューラルネットの基本モデルである2モジュールモデルを取り上げ、モジュール間相互作用の基本的な性質について考察した。モジュールが受ける相互作用の影響について議論するため、モジュールごとに引き込み領域やエネルギー平面の形状、さらにエネルギー関数値の時間変動の様子を調べた。この結果、以下のような動特性が明らかになった。

- (i) モジュールは、他方のモジュールの状態に依存して自己の情報処理様式をダイナミックに変化させている。これは引き込み領域の拡大縮小、エネルギー平面の変化というかたちで視覚的に確認された。
- (ii) モジュールの状態がローカル・コンテキストを簡単に満たすような状況にないとき、モジュール間に相互作用を介した競合が生じる。競合状態はダイナミックに相互作用し合うことで、やがて解消され、ある最終状態に落ち着く。

また、上記の動特性が連想性能としてどのように反映されるのかについて調べた。この結果、以下のような連想記憶モデルとしての特徴が明らかになった。

- A. 2モジュール間に相互作用を導入することで、パターンの構造に関係なく優れた連想性能が実現される。
- B. 記憶パターン数が多い場合やパターン間相関が強い場合、モジュール間相互作用に非線形性をもたせる（インターネットワークに多層ニューラルネットを用いる）ことが有効である。
- C. モジュールの最終状態間に一对多の関係がある場合でも、自己参照型相互作用を導入することで正しい想起が行われる。

第5章では、クラスを  $\mathcal{I}_2$  と  $\mathcal{I}_M$  の自己参照型相互作用を導入したマルチモジュール CCHN-LAGI を取り上げ、2種類の相互作用が同時に導入されることの効果について主に考察を行った。そこで、大局的相互作用（クラス  $\mathcal{I}_M$ ）が一部のモジュールだけに導入され、グローバル・コンテキストの情報が局所的相互作用（クラス  $\mathcal{I}_2$ ）を介してすべてのモジュールに伝搬するかどうか調べられた。動特性の考察には、第3章で提案した等価モデルの結合行列を固有値解析する方法を採用した。そこで、インターネットワークには2層の階層型ニューラルネットを用い、ユニットの出力関数を線形関数と仮定した。ランダムな記憶パターンを与えたとき、上記の CCHN-LAGI/L( $M$ ) がもつ動特性として以下のことが明らかになった。

- (a) モジュール数が多い場合、パターン空間における表現が冗長になり、1モジュールモデルに比べればノイズ空間は小さくなる。しかしながら、CCHN-LAGI/L( $M$ ) に大局的相互作用を導入しなければ、全くノイズ空間が生じないケースもあり、大局的相互作用を導入することの効果は明らかである。
- (b) 1モジュールモデルと同様、ノイズ空間ダイナミクスが有効に利用されることが連想性能の向上につながる。

(c) パターン空間が冗長になるのは、モジュール間に生じた競合関係が完全に解消されないために生じた偽の最終状態によるものと考えられる。

また、(優れた連想性能) = (多様な引き込み特性の実現) という観点から、CCHN-LAGI/L(M) の連想記憶モデルとしての評価が行われた。この結果、貢献度パラメータやモジュール数、さらに導入される相互作用の種類を変更することで、様々な連想特性(引き込み特性)が実現されることをシミュレーション実験により実証した。

以上の結果より、提案したマルチモジュールモデルの特徴として、以下のことが明らかになった。

- I. 相互作用を導入することでノイズ空間のダイナミクスが変化し、これはネットワークの状態をパターン空間に射影する機能をもつ。相互作用にはローカル・コンテキストを満足させるような機能を定義したが、上記のダイナミクスはこの機能を実現するために形成されたと考えられる。
- II. 局所的相互作用は局所的な入力情報に依存しているため、満足しようとしているローカル・コンテキストにコンフリクトが生じることが多い。これはモジュール間に存在する一種の競合と見てよく、これがうまく解消されないときは偽のグローバル・コンテキストを生じてしまう。
- III. II. の偽のグローバル・コンテキストがパターン空間を埋め尽くしてしまうとノイズ空間ダイナミクスは消滅し、連想特性は極度に低下する。
- IV. パターン空間ダイナミクスに対するノイズ空間ダイナミクスの優位性は引き込み特性を変化させる。また、II. の偽のグローバル・コンテキストを意図的に発生させることによっても引き込み特性を変化させることができる。これらは、本モデルのもつ構造パラメータを調整することで実現可能である。

本研究では、提案した動的マルチモジュールニューラルネットがランダムパターンだけでなくクラスター構造をもつパターンに対しても優れた連想性能をもつことを示した。また、連想性能の改善だけでなく、様々な連想特性が実現可能であることを示した。このことから、動的マルチモジュールニューラルネットを構築する上で、本研究が有効な手段を提供するといえる。

最後に、本研究に残された今後の課題について述べる。

まず、本研究で提案したモデル化手法を適用するために必要な仮定を以下に列挙する。

1. ネットワーク全体や構成モジュールの機能はあらかじめ与えられている。
2. モジュール構造や導入する相互作用の種類などはあらかじめ与えられている。
3. 漸近安定な特性をもったネットワークで解くことのできる問題に限定する。
4. 最終状態 (安定平衡点) はあらかじめ明示的に与えられる。

1. および 2. は、マルチモジュールニューラルネットにおいて最も重要でかつ困難な問題を含んでいる。これは、ある問題が与えられたときに

- どのように機能分担させて問題を解くのがよいか
- どのような構造をもったネットワークで解くのがよいか

ということを自律的に分析し、構造や情報処理様式を自己組織する能力がニューラルネットに求められる。しかしながら、このような能力の獲得をニューラルネットにおける適応・学習の枠組だけで議論することは難しい。生物においても、長時間にわたる淘汰と選択が繰り返された進化の産物として、上記の能力が実現されたと考えられる。また現実的には、ネットワークを自己組織する際に与えられる教師信号が非明示的であることも多いことから、上記の2点について本研究を発展させていくには、強化学習や遺伝的アルゴリズム、創発システム、人工生命などで行われている研究を参考にしていくべきであると思われる。

3. および 4. の仮定は、ネットワークが行う情報処理の様式や目的を固定的でかつ明示的なものに制限している。もちろん、この範疇のなかで解ける問題も多く存在するのだが、生物の神経系や個体集団など多要素系において行われる一般的な情報処理では、これらは動的でかつ非明示なもので表される場合が多い。このような問題は「創発システム」において最近活発に議論されている。北村は、創発に関して以下のように定義している [70]。

自律的に振る舞う個体 (要素) 間および環境との間の局所的な相互作用が大域的な秩序を発現し、他方、そのように生じた秩序が個体に振舞いを拘束するという双方向の動的過程により、新しい機能、形質、行動などが獲得されること

本研究で定義したローカル・コンテキストとグローバル・コンテキストという言葉がこの定義に無理矢理あてはめて考えてみると、ローカル・コンテキストは局所的な相互作用によって達成される目的に相当し、グローバル・コンテキストは局所的

な相互作用を通して発現する大局的な秩序に相当する．当然のことながら，本研究で行ったグローバル・コンテキストの想起は創発現象と呼べるほど高度な情報処理ではないし，大局的な秩序からの拘束というのも情報処理に組み込まれていない．しかしながら，このように本研究を創発システムという一般的なシステムへのアプローチと照らし合わせてみることは，本研究における今後の方向性を決定する上で重要であると思われる．今後，このような方向で本研究を発展させていきたいと考えている．



## 付録 A.1 マルチモジュール CCHN-LAGI の動作方程式

まず，式 (2.23) の時間微分を求める．これは，モジュールの結合行列が対称である（つまり  $T_{ij}^{(m)} = T_{ji}^{(m)}$ ）ことに注意すると，以下の式で表される．

$$\begin{aligned}
 \frac{dE}{dt} = & \sum_{m=1}^M \alpha^{(m)} \frac{dv_i^{(m)}}{dt} \left[ -\sum_i \sum_j T_{ij}^{(m)} v_j^{(m)} + \sum_i \frac{u_i^{(m)}}{r} - \sum_i J_i^{(m)} \right] \\
 & - \sum_{m=1}^M \frac{\alpha^{(m)}}{2} \sum_i \sum_j \frac{dT_{ij}^{(m)}}{dt} (-T_{ij}^{(m)} + v_i^{(m)} v_j^{(m)}) \\
 & + \sum_{m=1}^M \sum_{\substack{n=1 \\ n \neq m}}^M \beta^{(2,n,m)} \sum_i (v_i^{(m)} - o_i^{(2,n,m)(L+2)}) \left( \frac{dv_i^{(m)}}{dt} - \frac{do_i^{(2,n,m)(L+2)}}{dt} \right) \\
 & + \sum_{m=1}^M \beta^{(M,1,m)} \sum_i (v_i^{(m)} - o_i^{(M,1,m)(L+2)}) \left( \frac{dv_i^{(m)}}{dt} - \frac{do_i^{(M,1,m)(L+2)}}{dt} \right) \quad (\text{A.1.1})
 \end{aligned}$$

ここで，インターネットワーク出力の時間微分を求める必要があるが，この計算を簡単にするために以下の  $R^{(k,n,m)}$  を定義する．

$$R^{(k,n,m)} = \sum_i \frac{do_i^{(k,n,m)(L+2)}}{dt} (v_i^{(m)} - o_i^{(k,n,m)(L+2)}) \quad (\text{A.1.2})$$

式 (2.7), (2.8) の関係を用いて， $do_i^{(k,n,m)(L+2)}/dt$  は以下のようになる．

$$\frac{do_i^{(k,n,m)(L+2)}}{dt} = h'(s_i^{(k,n,m)(L+2)}) \sum_j \left[ \frac{dw_{ij}^{(k,n,m)(L+1)}}{dt} o_j^{(k,n,m)(L+1)} + w_{ij}^{(k,n,m)(L+1)} \frac{do_j^{(k,n,m)(L+1)}}{dt} \right] \quad (\text{A.1.3})$$

式 (A.1.2) に式 (A.1.3) を代入すると  $R^{(k,n,m)}$  は次のように書換えられる．

$$\begin{aligned}
 R^{(k,n,m)} = & \sum_i \sum_j \frac{dw_{ij}^{(k,n,m)(L+1)}}{dt} (o_j^{(k,n,m)(L+1)} \delta_i^{(k,n,m)(L+2)}) \\
 & + \sum_j \frac{do_j^{(k,n,m)(L+1)}}{dt} \left( \sum_i w_{ij}^{(k,n,m)(L+1)} \delta_i^{(k,n,m)(L+2)} \right)
 \end{aligned}$$

但し，

$$\delta_i^{(k,n,m)(L+2)} = h'(s_i^{(k,n,m)(L+2)}) \cdot (v_i^{(m)} - o_i^{(k,n,m)(L+2)})$$

式 (A.1.3) の関係を用いて， $do_j^{(k,n,m)(l)}/dt$  を層番号  $l$  に関して再帰的に求めていくと，最終的に  $R^{(k,n,m)}$  は次のように与えられる．

$$R^{(k,n,m)} = \sum_{l=1}^{L+1} \sum_i \sum_j \frac{dw_{ij}^{(k,n,m)(l)}}{dt} (o_j^{(k,n,m)(l)} \delta_i^{(k,n,m)(l+1)})$$

$$+ \sum_i \frac{do_i^{(k,n,m)(1)}}{dt} \left( \sum_j w_{ji}^{(k,n,m)(1)} \delta_j^{(k,n,m)(2)} \right) \quad (\text{A.1.4})$$

但し,

$$\delta_i^{(k,n,m)(L+2)} = h'(s_i^{(k,n,m)(L+2)}) \cdot (v_i^{(m)} - o_i^{(k,n,m)(L+2)}) \quad (\text{A.1.5})$$

$$\delta_i^{(k,n,m)(l+1)} = h'(s_i^{(k,n,m)(l+1)}) \cdot \left( \sum_j w_{ji}^{(k,n,m)(l+1)} \delta_j^{(k,n,m)(l+2)} \right) \quad (l = 1, \dots, L) \quad (\text{A.1.6})$$

クラス  $\mathcal{I}_2$  と  $\mathcal{I}_M$  のインターネットワーク入力が式 (2.21),(2.22) で与えられることを考慮すれば,  $dE/dt$  は式 (A.1.4) ~ (A.1.6) を用いて次のように書換えられる.

$$\begin{aligned} \frac{dE}{dt} = & \sum_{m=1}^M \sum_{i=1}^{N^{(m)}} \frac{dv_i^{(m)}}{dt} \left[ \alpha^{(m)} \cdot \left( - \sum_j T_{ij}^{(m)} v_j^{(m)} + \frac{u_i^{(m)}}{r} - J_i^{(m)} \right) \right. \\ & + \sum_{\substack{n=1 \\ n \neq m}}^M \beta^{(2,n,m)} \cdot \left( v_i^{(m)} - o_i^{(2,n,m)(L+2)} \right) \\ & - \sum_{\substack{n=1 \\ n \neq m}}^M \sum_j \left( \beta^{(2,m,n)} w_{ji}^{(2,m,n)(1)} \delta_j^{(2,m,n)(2)} + \beta^{(2,n,m)} w_{j,i+N^{(n)}}^{(2,n,m)(1)} \delta_j^{(2,n,m)(2)} \right) \\ & + \beta^{(M,1,m)} \cdot \left( v_i^{(m)} - o_i^{(M,1,m)(L+2)} \right) - \sum_{n=1}^M \beta^{(M,1,n)} \cdot \left( \sum_j w_{j,i+\varphi(m-1)}^{(M,1,n)(1)} \delta_j^{(M,1,n)(2)} \right) \\ & - \sum_{m=1}^M \sum_i \sum_j \frac{\alpha^{(m)} dT_{ij}^{(m)}}{2 dt} \left( -T_{ij}^{(m)} + v_i^{(m)} v_j^{(m)} \right) \\ & - \sum_{m=1}^M \sum_{\substack{n=1 \\ n \neq m}}^M \beta^{(2,n,m)} \sum_{l=1}^{L+1} \sum_i \sum_j \frac{dw_{ij}^{(2,n,m)(l)}}{dt} \left( o_j^{(2,n,m)(l)} \delta_i^{(2,n,m)(l+1)} \right) \\ & - \sum_{m=1}^M \beta^{(M,1,m)} \sum_{l=1}^{L+1} \sum_i \sum_j \frac{dw_{ij}^{(M,1,m)(l)}}{dt} \left( o_j^{(M,1,m)(l)} \delta_i^{(M,1,m)(l+1)} \right) \quad (\text{A.1.7}) \end{aligned}$$

マルチモジュール CCHN の動作方程式を導いた 2.2 の方法に従えば, 式 (A.1.7) より式 (2.24) ~ (2.29) で表されるマルチモジュール CCHN-LAGI の動作方程式を得る. これら動作方程式によって式 (2.23) のエネルギー関数が単調減少することは, 以下のように式 (2.24) ~ (2.29) を式 (A.1.7) に代入し直せば明らかである.

$$\begin{aligned} \frac{dE}{dt} = & - \sum_{m=1}^M \sum_i ckg'(ku_i^{(m)}) \cdot \left( \frac{du_i^{(m)}}{dt} \right)^2 - \sum_{m=1}^M \sum_i \sum_j \frac{\alpha^{(m)} \varepsilon}{2} \cdot \left( \frac{dT_{ij}^{(m)}}{dt} \right)^2 \\ & - \sum_{m=1}^M \sum_{\substack{n=1 \\ n \neq m}}^M \sum_{l=1}^{L+1} \sum_i \sum_j \beta^{(2,n,m)} \gamma \cdot \left( \frac{dw_{ij}^{(2,n,m)(l)}}{dt} \right)^2 \\ & - \sum_{m=1}^M \sum_{l=1}^{L+1} \sum_i \sum_j \beta^{(M,1,m)} \gamma \cdot \left( \frac{dw_{ij}^{(M,1,m)(l)}}{dt} \right)^2 \leq 0 \end{aligned}$$

## 付録 A.2 モデルの分類と略記方法

本論文のシミュレーション実験においては，様々なニューラルネットが用いられる．シミュレーション実験の目的をより明確にするため，これらを整理し，その略記方法についてまとめておく．

取り上げられるニューラルネットの種類は以下のように分類される．

1. マルチモジュール CCHN およびその等価モデル (CCHN)
  - (a) 相互作用のクラス —  $\mathcal{I}_1$ ,  $\mathcal{I}_2$  (LI),  $\mathcal{I}_M$  (GI),  $\mathcal{I}_2 + \mathcal{I}_M$  (LAGI)
  - (b) インターネットワークの種類 — 線形 (L), 非線形 (NL)
  - (c) モジュール数 ( $M$ )
2. ホップフィールドネット
  - (a) 従来モデル (HN)
    - i. 相関モデル (COR)
    - ii. 最適写像モデル (OPT)
  - (b) 1 モジュール CCHN の等価モデル (CCHN)
    - i. 相関モデル (COR)
    - ii. 最適写像モデル (OPT)

ここで ( ) 内の文字は，モデルの略記号を構成するための記号である．インターネットワークが線形とは出力関数が線形である 2 層の階層型ニューラルネットで構成されたもので，非線形とはそれ以外のものである．また，等価モデルとはマルチモジュール CCHN および 1 モジュール CCHN のなかで自己想起型連想記憶モデルとして置き換えられるものを表している．

1. のマルチモジュール CCHN の場合，次のように略記号を構成する．

CCHN-相互作用のクラス/インターネットワークの種類 (モジュール数)

例えば，相互作用のクラスが  $\mathcal{I}_2 + \mathcal{I}_M$ ，インターネットワークが線形である 2 モジュール CCHN は，“CCHN-LAGI/L(2)” で表される．また，この等価モデルについても “CCHN-LAGI/L(2)” と表される (これは 5.3 で取り扱われる)．2. のホップフィールドモデルの場合，従来モデルなのか 1 モジュール CCHN の等価モデルかで分けられ，さらに結合行列の種類で細分される．略記方法は以下の通りである．

CCHN-結合行列の種類

HN-結合行列の種類

例えば，モジュールの結合行列が最適写像モデルで構成された 1 モジュール CCHN の等価モデルは “CCHN-OPT” で表される．

## 付録 A.3 HN-COR における結合ダイナミックスの解

式 (3.7) の解は自明であるので，式 (3.8) の解の求め方だけを示す．

記憶パターンの相関行列  $S$  を以下のように定義する．

$$S = \sum_{\mu=1}^P \xi^{\mu} \xi^{\mu}$$

これを式 (3.8) に代入し，以下の関係を用いれば，

$$\frac{d}{dt} \left\{ \mathbf{w} \exp\left(\frac{\mathbf{S}}{\gamma} t\right) \right\} = \frac{d\mathbf{w}}{dt} \exp\left(\frac{\mathbf{S}}{\gamma} t\right) + \frac{1}{\gamma} \mathbf{w} \mathbf{S} \exp\left(\frac{\mathbf{S}}{\gamma} t\right) = \frac{1}{\gamma} \mathbf{S} \exp\left(\frac{\mathbf{S}}{\gamma} t\right)$$

$\mathbf{w}(t)$  の解は以下のように与えられる．

$$\mathbf{w}(t) = \mathbf{w}(0) \exp\left(-\frac{\mathbf{S}}{\gamma} t\right) + \frac{1}{\gamma} \mathbf{S} \left[ \int_0^t \exp\left(\frac{\mathbf{S}}{\gamma} \tau\right) d\tau \right] \exp\left(-\frac{\mathbf{S}}{\gamma} t\right) \quad (\text{A.3.1})$$

ここで， $\mathbf{w}(0)$  は初期結合行列である． $\mathbf{w}(t)$  に対する計算を進める前に，式 (A.3.1) 中のいくつかの項について計算を行っておく．

$$\begin{aligned} \exp\left(-\frac{\mathbf{S}}{\gamma} t\right) &= \left[ \mathbf{I} + \sum_{l=1}^{\infty} \frac{(t/\gamma)^l}{l!} \mathbf{S}^l \right]^{-1} \\ &= \left[ \mathbf{I} + \sum_{l=1}^{\infty} \frac{(t/\gamma)^l}{l!} \mathbf{Q} \mathbf{A}^l \mathbf{Q}^{-1} \right]^{-1} \\ &= \mathbf{Q} \begin{pmatrix} 1 + \sum_{l=1}^{\infty} \frac{(t/\gamma)^l}{l!} \lambda_1^l & & & & \\ & \ddots & & & \\ & & 1 + \sum_{l=1}^{\infty} \frac{(t/\gamma)^l}{l!} \lambda_{\kappa}^l & & \\ & & & 1 & \\ & & & & \ddots \\ & & & & & 1 \end{pmatrix}^{-1} \mathbf{Q}^{-1} \\ &= \mathbf{Q} \begin{pmatrix} \exp\left(-\frac{\lambda_1}{\gamma} t\right) & & & & \\ & \ddots & & & \\ & & \exp\left(-\frac{\lambda_{\kappa}}{\gamma} t\right) & & \\ & & & 1 & \\ & & & & \ddots \\ & & & & & 1 \end{pmatrix} \mathbf{Q}^{-1} \quad (\text{A.3.2}) \end{aligned}$$

$$\begin{aligned}
 \int_0^t \exp\left(\frac{\mathbf{S}}{\gamma}\tau\right)d\tau &= \int_0^t \left\{ \mathbf{I} + \sum_{l=1}^{\infty} \frac{(\tau/\gamma)^l}{l!} \mathbf{S}^l \right\} d\tau \\
 &= \int_0^t \left\{ \mathbf{I} + \sum_{l=1}^{\infty} \frac{(\tau/\gamma)^l}{l!} \mathbf{Q} \mathbf{A}^l \mathbf{Q}^{-1} \right\} d\tau \\
 &= \mathbf{Q} \begin{pmatrix} \frac{\gamma}{\lambda_1} \left\{ \exp\left(\frac{\lambda_1}{\gamma}t\right) - 1 \right\} & & & & \\ & \ddots & & & \\ & & \frac{\gamma}{\lambda_\kappa} \left\{ \exp\left(\frac{\lambda_\kappa}{\gamma}t\right) - 1 \right\} & & \\ & & & t & \\ & & & & \ddots \\ & & & & & t \end{pmatrix} \mathbf{Q}^{-1} \quad (\text{A.3.3})
 \end{aligned}$$

但し,

$$\mathbf{S} = \mathbf{Q} \mathbf{A} \mathbf{Q}^{-1}$$

$$\mathbf{A} = \text{diag}\{\lambda_1, \dots, \lambda_\kappa, 0, \dots, 0\}$$

ここで,  $\kappa$  は  $\mathbf{S}$  のランクである. よく知られているように,  $\mathbf{S}$  は相関行列であるので非ゼロ固有値  $\lambda_i$  ( $i = 1, \dots, \kappa$ ) ( $\mathbf{A}$  の非ゼロ対角要素) は正となる. 式 (A.3.2), (A.3.3) を式 (A.3.1) に代入すれば, 以下のような  $w(t)$  の解が得られる.

$$\begin{aligned}
 w(t) &= w(0) \exp\left(-\frac{\mathbf{S}}{\gamma}t\right) + \frac{1}{\gamma} \mathbf{Q} \mathbf{A} \mathbf{Q}^{-1} \mathbf{Q} \begin{pmatrix} \frac{\gamma}{\lambda_1} \left\{ \exp\left(\frac{\lambda_1}{\gamma}t\right) - 1 \right\} & & & & \\ & \ddots & & & \\ & & \frac{\gamma}{\lambda_\kappa} \left\{ \exp\left(\frac{\lambda_\kappa}{\gamma}t\right) - 1 \right\} & & \\ & & & t & \\ & & & & \ddots \\ & & & & & t \end{pmatrix} \mathbf{Q}^{-1} \\
 &\quad \times \mathbf{Q} \begin{pmatrix} \exp\left(-\frac{\lambda_1}{\gamma}t\right) & & & & \\ & \ddots & & & \\ & & \exp\left(-\frac{\lambda_\kappa}{\gamma}t\right) & & \\ & & & 1 & \\ & & & & \ddots \\ & & & & & 1 \end{pmatrix} \mathbf{Q}^{-1} \\
 &= w(0) \exp\left(-\frac{\mathbf{S}}{\gamma}t\right) + \mathbf{Q} \begin{pmatrix} 1 - \exp\left(-\frac{\lambda_1}{\gamma}t\right) & & & & \\ & \ddots & & & \\ & & 1 - \exp\left(-\frac{\lambda_\kappa}{\gamma}t\right) & & \\ & & & 0 & \\ & & & & \ddots \\ & & & & & 0 \end{pmatrix} \mathbf{Q}^{-1}.
 \end{aligned}$$

## 付録 A.4 CCHN-LAGI/L( $M$ ) と等価な自己想起型連想記憶の結合行列

以下では、式 (2.23) において  $J_i^{(m)} = 0, r_i^{(m)} \gg 0$  として結合行列を導出していく。まず、式 (2.23) を以下のような行列形式で書き直す。

$$E = -\frac{1}{2} \sum_{m=1}^M \alpha^{(m)} \mathbf{v}^{(m)'} \mathbf{T}_f^{(m)} \mathbf{v}^{(m)} + \frac{1}{2} \sum_{m=1}^M \sum_{\substack{n=1 \\ n \neq m}}^M \beta^{(2,n,m)} \mathbf{Q}_1^{(2,n,m)'} \mathbf{Q}_1^{(2,n,m)} + \frac{1}{2} \sum_{m=1}^M \beta^{(M,1,m)} \mathbf{Q}_2^{(M,1,m)'} \mathbf{Q}_2^{(M,1,m)} \quad (\text{A.4.1})$$

ただし、

$$\mathbf{Q}_1^{(2,n,m)} = \mathbf{v}^{(m)} - \mathbf{w}^{(2,n,m)} \begin{pmatrix} \mathbf{v}^{(n)} \\ \mathbf{v}^{(m)} \end{pmatrix} \quad (\text{A.4.2})$$

$$\mathbf{Q}_2^{(M,1,m)} = \mathbf{v}^{(m)} - \mathbf{w}^{(M,1,m)} \begin{pmatrix} \mathbf{v}^{(1)} \\ \vdots \\ \mathbf{v}^{(M)} \end{pmatrix} \quad (\text{A.4.3})$$

ここで、 $\mathbf{v}^{(m)} \in \mathcal{R}^{N^{(m)}}$ ,  $\mathbf{T}_f^{(m)} \in \mathcal{R}^{N^{(m)}} \times \mathcal{R}^{N^{(m)}}$  は第  $m$  モジュールの状態ベクトルおよび結合行列を表し、また  $\mathbf{w}^{(2,n,m)} \in \mathcal{R}^{N^{(m)}} \times \mathcal{R}^{N^{(n)}+N^{(m)}}$  と  $\mathbf{w}^{(M,1,m)} \in \mathcal{R}^{N^{(m)}} \times \mathcal{R}^N$  はそれぞれクラス  $\mathcal{I}_2, \mathcal{I}_M$  インターネットワークの結合行列である。なお簡単のため、モジュールの結合  $\mathbf{T}^{(m)}$  はあらかじめ求められたものを用いることとし、これを  $\mathbf{T}_f^{(m)}$  とおく。また、エネルギー損失と外部入力に関する項は省略する。

次に、CCHN-LAGI/L における第  $m$  モジュールの状態を、以下のような  $N$  次元ベクトルで書き直す。

$$\hat{\mathbf{v}}^{(m)} = \left[ \underbrace{0 \ \dots \ 0}_{\varphi(m-1)} \ \mathbf{v}^{(m)'} \ \underbrace{0 \ \dots \ 0}_{N-\varphi(m)} \right]' \in \mathcal{R}^N \quad (\text{A.4.4})$$

この式 (A.4.4) および式 (5.1) ~ (5.3) の定義を参考にして式 (A.4.1) を書き直すと以下のようなになる。

$$E = -\frac{1}{2} \sum_{m=1}^M \alpha^{(m)} \hat{\mathbf{v}}^{(m)'} \hat{\mathbf{T}}_f^{(m)} \hat{\mathbf{v}}^{(m)} + \frac{1}{2} \sum_{m=1}^M \sum_{\substack{n=1 \\ n \neq m}}^M \beta^{(2,n,m)} \left\{ \hat{\mathbf{v}}^{(m)} - \hat{\mathbf{w}}^{(2,n,m)} (\hat{\mathbf{v}}^{(n)} + \hat{\mathbf{v}}^{(m)}) \right\}' \left\{ \hat{\mathbf{v}}^{(m)} - \hat{\mathbf{w}}^{(2,n,m)} (\hat{\mathbf{v}}^{(n)} + \hat{\mathbf{v}}^{(m)}) \right\} + \frac{1}{2} \sum_{m=1}^M \beta^{(M,1,m)} (\hat{\mathbf{v}}^{(m)} - \hat{\mathbf{w}}^{(M,1,m)} \mathbf{v})' (\hat{\mathbf{v}}^{(m)} - \hat{\mathbf{w}}^{(M,1,m)} \mathbf{v}) \quad (\text{A.4.5})$$

但し、

$$\mathbf{v} = (\mathbf{v}^{(1)'} \dots \mathbf{v}^{(M)'})' \in \mathcal{R}^N \quad (\text{A.4.6})$$

ここで次の関係を利用すると、

$$\widehat{\mathbf{v}}^{(m)} = \mathbf{H}^{(m)} \mathbf{v} \quad (\text{A.4.7})$$

式 (A.4.5) はさらに以下のように書き換えられる。

$$\begin{aligned} E = & \frac{1}{2} \mathbf{v}' \sum_{m=1}^M \left[ -\alpha^{(m)} \mathbf{H}^{(m)} \widehat{\mathbf{T}}_f^{(m)} \mathbf{H}^{(m)} \right. \\ & + \sum_{\substack{n=1 \\ n \neq m}}^M \beta^{(2,n,m)} \left\{ \mathbf{H}^{(m)} - \widehat{\mathbf{w}}^{(2,n,m)} (\mathbf{H}^{(n)} + \mathbf{H}^{(m)}) \right\}' \left\{ \mathbf{H}^{(m)} - \widehat{\mathbf{w}}^{(2,n,m)} (\mathbf{H}^{(n)} + \mathbf{H}^{(m)}) \right\} \\ & \left. + \beta^{(M,1,m)} (\mathbf{H}^{(m)} - \widehat{\mathbf{w}}^{(M,1,m)})' (\mathbf{H}^{(m)} - \widehat{\mathbf{w}}^{(M,1,m)}) \right] \mathbf{v} \end{aligned} \quad (\text{A.4.8})$$

ここで式 (A.4.8) を以下のように置き換え、

$$E = \frac{1}{2} \mathbf{v}' \mathbf{C} \mathbf{v} \quad (\text{A.4.9})$$

これを時間  $t$  で微分する。ここで  $\mathbf{C}$  が対称行列となることに注意すれば、 $dE/dt$  は以下のように表される。

$$\frac{dE}{dt} = \mathbf{v}' \mathbf{C} \frac{d\mathbf{v}}{dt} + \frac{1}{2} \mathbf{v}' \frac{d\mathbf{C}}{dt} \mathbf{v} \quad (\text{A.4.10})$$

CCHN-LAGI/L の動作方程式は  $dE/dt \leq 0$  が成り立つよう決定され、これは以下の式で表される。

$$c_1 \frac{d\mathbf{u}}{dt} = -\mathbf{C} \mathbf{v} \quad (\text{A.4.11})$$

$$c_2 \frac{d\mathbf{C}}{dt} = -\mathbf{v} \mathbf{v}' \quad (\text{A.4.12})$$

ここで式 (A.4.12) の解を  $\mathbf{C}_f$  とし、これを式 (A.4.11) に代入すれば、自己相関連想記憶モデルの活性ダイナミクスと同じ形式で書けることがわかる。



## 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] W. S. McCulloch and W. Pitts: “A logical calculus of the ideas immanent in nervous activity”, *Bull. Math. Biophysics*, **5**, pp.115-133 (1943).
- [3] D. O. Hebb: *The organization of behavior*, Wiley (1949).
- [4] A. L. Hodgkin and A.F. Huxley: “A quantitative description of membrane current and its application to conduction and excitation in nerve”, *J. Physiol.*, **117**, pp.500-544 (1952).
- [5] F. Rosenblatt: *Principles of neurodynamics – perceptrons and the theory of brain mechanisms*, Spartan (1961).
- [6] B. Widrow and M. E. Hoff: “Adaptive switching circuits”, *1960 IRE WESCON Convention Record*, pp.96-104 (1960).
- [7] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group: *Parallel distributed processing*, MIT Press (1986).
- [8] J. J. Hopfield: “Neural networks and physical systems with emergent collective computational abilities”, *Proc. Natl. Sci. USA*, **79**, pp.2254-2558 (1982).
- [9] J. J. Hopfield and D. W. Tank: “Neural computation of decisions in optimization problems”, *Biol. Cybern.*, **52**, pp.141-152 (1985).
- [10] K. Fukushima: “Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”, *Biol. Cybern.*, **36**, 4, pp.193-202 (1980).
- [11] 堤： “ニューラルネットワーク研究の最新動向”，システム/制御/情報，**36**, 10, pp.619-624 (1992).
- [12] T. Kohonen: *Self-organization and associative memory*, 3rd ed. Springer-Verlag (1989).

- [13] J. J. Hopfield and D. W. Tank: "Computing with neural circuits: a model", *science*, **233**, pp.625-633 (1986).
- [14] J. Szentágothai: "The 'module-concept' in cerebral cortex architecture", *Brain Research*, **95**, pp.475-496 (1975).
- [15] 塚原 (編) : 脳の情報処理, 朝倉書店 (1984).
- [16] S. Zeki and S. Shipp: "The functional logic of cortical connections", *Nature*, **335**, pp.311-317 (1988).
- [17] M. S. Gazzaniga: "Organization of the human brain", *Science*, **245**, pp.947-952 (1989).
- [18] D. H. Hubel and T. N. Wiesel: "Functional architecture of macaque monkey visual cortex", *Proc. Roy. Soc. Lond, B*, **198**, pp.1-59 (1977).
- [19] G. Carpenter and S. Grossberg: "The ART of adaptive pattern recognition by a self-organizing neural network", *IEEE Computer*, **21**, 3, pp.77-88 (1988).
- [20] K. Fukushima and T. Imagawa "Recognition and segmentation of connected characters with selective attention", *Neural Networks*, **6**, 1, pp.33-41 (1993).
- [21] J. M. J. Murre, R. H. Phaf, and G. Wolters: "CALM: categorizing and learning module", *Neural Networks*, **5**, pp.55-82 (1992).
- [22] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang: "Phoneme recognition using time-delay neural networks", *IEEE Trans. on Acous., Speech and Signal Process.*, **37**, 3, pp.328-339 (1989).
- [23] 小澤, 松本, 堤: "階層化コホーネンネットの単音節認識への応用", 第33回システム制御情報学会研究発表講演会論文集, pp.261-262 (1989).
- [24] H. Kita, H. Masataki, and Y. Nishikawa: "NN/II: improved version of a network for large-scale pattern recognition tasks", *Proc. of Int. Joint Conf. on Neural Networks (IJCNN91-Singapore)*, Vol.3, pp.2080-2085 (1991).
- [25] 岩田, 当麻, 松尾, 鈴村: "大規模4層ニューラルネット 'CombNET-II'", 電子通信情報学会論文誌 (D-II), **J73-D-II**, 8, pp.1261-1267 (1990).

- 
- [26] M. Kotani, H. Matsumoto, and T. Kanagawa: “Acoustic diagnosis for compressor with hybrid neural network”, *Proc. of Int. Joint Conf. on Neural Networks (IJCNN91-Seattle)*, Vol.1, pp.251-256 (1991).
- [27] M. I. Jordan and R. A. Jacobs: “Hierarchical mixtures of experts and the EM algorithm”, *Neural Computation*, **6**, 2, pp.181-214 (1994).
- [28] J. A. Anderson: “A simple neural network generating an interactive memory”, *Math. Biology*, **42**, pp.339–364 (1972).
- [29] S. Amari: “Neural theory of association and concept-formation”, *Biol. Cybern.*, **26**, pp.175-185 (1977).
- [30] 中野：アソシアトロン，昭晃堂 (1979).
- [31] K. Murakami and T. Aibara: “An improvement on the Moore-Penrose generalized inverse associative memory”, *IEEE Trans. on System, Man, and Cybernetics*, **SMC-17**, pp.699-707 (1987).
- [32] B. Kosko: “Adaptive bidirectional associative memories”, *Applied Optics*, **26**, pp.4947-4960 (1987).
- [33] M. H. Hassoun and A. M. Youssef: “High performance recording algorithm for Hopfield model associative memories”, *Optical Engineering*, **28**, pp.46-54 (1989).
- [34] M. H. Hassoun: “Dynamic heteroassociative neural memories”, *Neural Networks*, **2**, pp.275-287 (1989).
- [35] D. J. Amit, H. Gutfreund, and H. Sompolinsky: “Storing in-finite numbers of patterns in a spin-glass model of neural networks”, *Phys.Rev.Lett.*, **55**, pp.1530-1533 (1985).
- [36] R. J. McEliece et al.: “The capacity of the Hopfield associative memory”, *IEEE Trans. on Information. Theory*, **33**, pp.461-482 (1987).
- [37] S. Amari and K. Maginu: “Statistical neurodynamics of associative memory”, *Neural Networks*, **1**, pp.63-73 (1988).

- [38] S. V. B. Aiyer, M. Niranjana, and F. Fallside: "A theoretical investigation into the performance of the Hopfield model", *IEEE Trans. on Neural Networks*, **1**, pp.204-215 (1990).
- [39] M. Okada: "A Hierarchy of macrodynamical equations for associative memories", *Neural Networks*, **8**, 6, pp.833-838 (1995).
- [40] 金道: "連想記憶のダイナミックスの幾何学的解析", 電子情報通信学会論文誌 D-II, **J78-D-II**, 12, pp.1877-1889 (1995).
- [41] S. Amari: "Characteristics of sparsely encoded associative memory", *Neural Networks*, **2**, pp.451-457 (1989).
- [42] 森田, 吉澤, 中野: "自己相関連想記憶の想起過程とその改良", 電子情報通信学会論文誌 D-II, **J73-D-II**, 2, pp.232-242 (1990).
- [43] 掛谷, 金道: "符号反転記憶法による自己相関連想記憶の改良", 神経回路学会誌, **1**, 1, pp.20-26 (1994).
- [44] 喜多: personal communication (1997)
- [45] D. Casasent and B. Telfer: "High capacity recognition associative processors", *Neural Networks*, **5**, pp.687-698 (1991).
- [46] K. Tsutsumi: "Cross-coupled Hopfield nets via generalized-delta-rule-based inter-networks", *Proc. of Int. Joint Conf. on Neural Networks (IJCNN90-San Diego)*, Vol.II, pp.259-265 (1990).
- [47] J. J. Hopfield: "Neurons with graded response have collective computational properties like those of two-state neurons", *Proc. Natl. Aca. Sci. U.S.A.*, **81**, pp.3088-3092 (1984).
- [48] K. Tsutsumi and H. Matsumoto: "Neural computation and learning strategy for manipulator position control", *Proc. of IEEE Int. Conf. on Neural Networks (ICNN-87)*, Vol.IV, pp.525-534 (1987).
- [49] K. Tsutsumi: "A multi-layered neural network composed of backprop. and Hopfield nets and internal space representation", *Proc. of Int. Joint Conf. on Neural Networks (IJCNN-89-Washington DC)*, Vol.II, pp.365-371 (1989).

- 
- [50] M. I. Jordan: “Serial order: a parallel, distributed processing approach”, *Tech. Report ICS-8604*, University of California, San Diego (1986).
- [51] J. L. Elman: “Finding structure in time”, *Tech. Report CRL-8801*, University of California, San Diego (1988).
- [52] R. J. Williams and D. Zipser: “Gradient-based learning algorithms for recurrent connectionist networks”, *Tech. Report NU-CCS-90-9*, Northeastern University (1990).
- [53] F. J. Pineda: “Dynamics and architecture for neural computation”, *Journal of Complexity*, **4**, pp.216-245 (1988).
- [54] K. Doya and S. Yoshizawa: “Adaptive neural oscillator using continuous-time back-propagation learning”, *Neural Networks*, **2**, pp.375-385 (1989).
- [55] B. A. Pearlmutter: “Dynamic recurrent neural networks”, *Tech. Report CMU-CA-90-196* (1990).
- [56] M. Sato: “A real time algorithm for recurrent analog neural networks”, *Biological Cybernetics*, **62**, pp.237-241 (1990).
- [57] 小谷, 小澤: “ニューラルネットワーク応用の最新動向”, *システム/制御/情報*, **36**, 10, pp.669-677 (1992).
- [58] B. L. M. Happel, J. M. J. Murre: “Design and evolution of modular neural network architectures”, *Neural Networks*, **7**, pp.985-1004 (1994).
- [59] 船橋: “ニューラル・ネットワークによる連続写像の近似的実現について”, *電子情報通信学会技術報告*, MBE88-9 (1988).
- [60] J. Hertz, A. Krogh, and R. G. Palmer: *Introduction to the theory of neural computation*, Addison-Wesley (1990).
- [61] 甘利, *神経回路網の数理*, 産業図書, 1978.
- [62] H. Kakeya and T. Kindo: “Eigenspace separation of autocorrelation memory matrices for capacity expansion”, *Neural Networks*, **10**, pp.833-843 (1997).

- [63] S-W. Zhang, A. G. Constantinides, and L-H. Zou: "Further noise rejection in linear associative memories", *Neural Networks*, **5**, pp.163-168 (1992).
- [64] S. Ozawa, K. Tsutsumi, and N. Baba "Association performance of cross-coupled Hopfield nets for correlated patterns", *Proc. of Int. Joint Conf. on Neural Networks (IJCNN93-Nagoya)*, Vol.II, pp.2335-2338 (1993).
- [65] I. Kanter and H. Sompolinsky: "Associative recall of memory without errors", *Physical Review A*, **35**, pp.380-392 (1987).
- [66] D. W. Tank and J. J. Hopfield: "Simple "neural" optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit", *IEEE Trans. on Circuits and Systems*, **CAS-33**, 5, pp.533-541 (1986).
- [67] S. Ozawa, K. Tsutsumi, and N. Baba: "The estimation of cross-coupled Hopfield nets as an interactive modular neural network, *Proc. Int. IEEE Conf. on Neural Networks (ICNN94-Orlando)*, Vol.III, pp.1340-1345 (1994).
- [68] T. Hatanaka: "Adjustment of the basin size in autoassociative memories by use of the BPTT technique", *Proc. Int. Joint Con. on Neural Networks (IJCNN91-Singapore)*, Vol.II, pp.1723-1728 (1991).
- [69] 森田, 吉澤, 中野: 非単調ダイナミクスを用いた構造をもつパターンの連想記憶, *電子情報通信学会論文誌 D-II*, **J75-D-II**, 11, pp.1884-1891 (1992).
- [70] 北村: "創発的機能形成のシステム理論に向けて", *計測と制御*, **35**, 7, pp.492-494 (1996).

# 本論文に関する発表論文

## 学術論文

1. 小澤，堤： “モジュール化ニューラルネットモデルの提案とその連想記憶能力の評価”，電子情報通信学会論文誌 D-II, **J77-D-II**, 6, pp.1135-1145 (1994).
2. 小澤，堤，馬場： “クロス結合ホップフィールドネットから導出される連想記憶モデルとそのノイズ空間ダイナミクスの役割”，電気学会論文誌 C, **117-C**, 9, pp.1253-1258 (1997).
3. 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 (1997.12 掲載予定).
4. S. Ozawa, K. Tsutsumi, and N.Baba: “An artificial modular neural network and its basic dynamical characteristics”, *Biological Cybernetics* (in press).
5. S. Ozawa, K. Tsutsumi, and N. Baba: “A high performance associative neural memory utilizing the noise-subspace dynamics”, *Neural Processing Letters* (submitted).

## 国際会議論文

1. S. Ozawa, K. Tsutsumi, and H. Matsumoto: “Basic dynamical properties of cross-coupled Hopfield nets”, *Proc. of International Joint Conf. on Neural Networks (IJCNN91-Singapore)*, **III**, pp.1949-1954 (1991).
2. S. Ozawa, K. Tsutsumi, and H. Matsumoto: “Association dynamics of cross-coupled Hopfield nets with many-to-many mapping internetworks”, *Artificial Neural Networks 2*, Elsevier, pp.375-378 (1992). (Proc. of International Conf. on Artificial Neural Networks, ICANN92-Brighton, UK)
3. S. Ozawa and K. Tsutsumi: “Association performance of cross-coupled Hopfield nets for correlated patterns”, *Proc. of International Joint Conf. on Neural Networks (IJCNN93-Nagoya)*, **II**, pp.2335-2338 (1993).

4. S. Ozawa, K. Tsutsumi, and N. Baba: "The estimation of cross-coupled Hopfield nets as an interactive modular neural network", *Proc. of International Conf. on Neural Networks (ICNN94-Orlando)*, **III**, pp.1340-1345 (1994).
5. S. Ozawa, K. Tsutsumi, and N. Baba: "A modular neural network with local and global interactions", *Proc. of World Congress on Neural Networks (WCNN95-Washington DC)*, **I**, pp.471-476 (1995).

#### 著書

1. 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).
2. 西川, 北村 (編): ニューラルネットと計測制御, 朝倉書店 (1995).  
(第2, 7章を分担執筆)

#### 解説論文

1. 小谷, 小澤: "ニューラルネットワーク応用の最新動向", システム/制御/情報, **36**, 10, pp.669-677 (1992).
2. 馬場, 小澤: "ニューロ技術のロボットへの応用", 日本ロボット学会誌, **11**, 1, pp.44-88 (1993).