

ユニットの疲労効果を取り入れた連想型スキーマモデル

メタデータ	言語: Japanese 出版者: 明治大学教養論集刊行会 公開日: 2012-05-16 キーワード (Ja): キーワード (En): 作成者: 阪井, 和男, 和田, 悟 メールアドレス: 所属:
URL	http://hdl.handle.net/10291/12309

ユニットの疲労効果を取り入れた 連想型スキーマモデル

阪 井 和 男
和 田 悟

—要 旨—

PDP 型スキーマモデルをもとにして、競合するスキーマの交代想起を個々のユニットの疲労度によって実現する方法を提案する。このために、ユニットの継続した興奮に対して、疲労度という内部自由度を新たに導入した。疲労度はユニットの活性化の禁止・許容を支配し、[0, 臨界疲労度] の間でヒステリシスを示すように設定した。その結果、局所的なメカニズムの導入にもかかわらず、長時間にわたって大域的に安定なスキーマの交代想起が実現できた。

第1章 はじめに

従来の認知科学では、人間の情報処理過程を直列的記号処理モデルで説明する試みを中心であった。しかし、このモデルで人間の複雑な認知現象を説明しようとして、モデルを詳細なレベルまで記述すると、処理すべきステップ数は膨大なものになってしまう。そこで、この種のモデルは、コンピュータに比べてはるかに処理の遅い脳の情報処理モデルとして、不適切なのではないか、という認識が生まれてきた。

この認識のもとで、脳の情報処理モデルとしてより適切なモデルを構成する枠組みとして注目を集めている研究がある。それが、ラムエルハートたちの認知科学者を中心とした研究グループによって進められている「並列分散処理 (Parallel Distributed Processing: PDP)」研究である。

PDP 研究によるモデルでは、脳の生理学的な側面から示唆を受けて、単純な構成要素間の並列的な相互作用によって認知過程を説明しようとする。このモデルは一般に「ニューラルネットワーク」と呼ばれ、近年では工学的応用が華々しい。

研究の系譜をたどってみると、ローゼンブラットが提案したパターンの識別方法を学習するパーセプトロン^{1) 2)}に行き着く。パーセプトロンそのものは、その限界がミンスキーとパパート³⁾によって指摘され、米国での研究は衰退に向ったが、日本では甘利俊一、福島、中野薫らによって地道で着実な研究が行われていた。小脳のモデルとして確立されたパーセプトロンと似ているが、その違いは入力層と出力層の間に任意の数の層が入っている点である。このため、内部表現の自由度が大幅に増えている。

現在のネットワークモデルの再興に大きな役割を果たしたものにホップフィールド⁴⁾によるネットワークの定式化がある。ホップフィールドらはこのネットワークを使って、巡回セールスマンのような組み合わせ最適問題に対する近似解を与えている。計算量の壁などで解決困難であった問題に、実際的な解を与えたことに大きな意義がある。

この種のモデルのもっとも重要な利点は、複雑で不確定な外界の構造に対応して、柔軟で連続的に変化していく適応性を備えていることである。従来のエキスパートシステムなど、記号処理による AI の弱点を補うものであり、学習を用いたパターン処理への応用がもっともよく研究されている。

パターン認識や推論・学習・記憶・言語などの人間の認知機能に関する問題を、神経回路網 (ニューラル・ネットワーク) 類似のシステムをベースにして行う研究のことをコネクショニズムという。そして、これによるモデルをコネ

クシヨニストモデル，または並列分散処理（PDP）モデルという。

コネクシヨニズムの名前の由来は，システムの「知識」がユニット間の結合の重みで表されることから名付けられた。処理システムの構成は，脳と同様に単純だが膨大な数のユニット（プロセッサ）を互いに接続したものである。各ユニットは活性量と呼ばれる内部状態を持ち，この値に従って他のユニットに信号を送る⁵⁾。出力された信号が興奮性のものか抑制性のものかは，ユニット間の結合の符号によって決定される。結合が正の場合には出力される信号は興奮性であり，結合が負の場合には信号は抑制性である。1つのユニットには他の多くのユニットからの入力があり，入力値の重み付き総和があるしきい値を超えたときのみ出力するという単純な機能を持っている。そして，多数のユニットの並列計算により，全体として複雑な機能をもつネットワークになっている。このシステムの特徴は，

- i) 多数のユニットが同時に稼働し，超高速並列処理が行なわれる
- ii) いくつかのユニットが誤動作を起こしたり壊れても，全体の機能に影響がでない
- iii) 入力に対する重みとしきい値は，学習によって最適値に変わっていくという自己組織化能力をもっている

などである。

しかし，工学的応用に力点におかれる一般の「ニューラルネットワーク」研究と，このPDP研究とは，次の点で大きく異なっている。PDP研究の関心は工学的応用に応用することではなく，あくまでも生理学的に基礎付けられた認知過程の説明モデルを構成することにある。

1980年代前半にカリフォルニア大学サンディエゴ校でPDP研究を推し進めていたマクレランドとラメルハートたち⁶⁾によるPDPグループの研究は，認知のモデルとしてのネットワークモデルということを明確に表明していた。このモデルの特徴は，認知過程における情報処理の時間的経過や表現の性質，学習機構を適切に説明できることである。彼らは，ネッカーキューブの認知プロ

セスを説明するスキーマモデル^{7) 6)}を提案している。この論文では、PDP 型のスキーマモデルを改良し、ネットワークを構成するユニットに疲労効果を取り入れることで、ネッカーキューブの認知に対応する2つのスキーマの間を交互に想起するモデルを提案する。

本論文の構成は次の通りである。第2章で、さまざまなネットワークモデルの位置付けを明確にし、スキーマ概念の歴史的な変遷をふりかえったのち、PDP モデルにおけるスキーマについてまとめる。第3章は、PDP 型のスキーマモデル^{7) 6)}を紹介し、第4章でネッカーキューブへの応用とシミュレーション結果を示す。ユニットの疲労効果を取り入れた連想型スキーマモデルについては第5章で扱う。第6章は、まとめと議論に当てる。

第2章 ネットワークモデルとスキーマ

はじめに、ネットワークモデルについて紹介し、それらの位置付けを明らかにしておく。そのうち、認知科学の中心的なテーマの1つであるスキーマ概念について、その歴史的な変遷をまとめ、PDP モデルにおけるスキーマモデルについて詳述する。

2.1 ニューラルネットのさまざまなモデル

現在のコネクショニズムを特徴付けているのは、それが認知のモデルとして語られているという点である。ところが、認知モデルの観点からでも、実に多様なコネクショニズムのアプローチが存在する。その全貌をダイアー⁸⁾は次の3つに分けている。以下の内容は「心の計算理論⁹⁾」より引用させていただいた。

2.1.1 ローカリスト・コネクショニスト・ネットワーク

ローカリスト・コネクショニスト・ネットワーク (localist connectionist network) は、コネクショニズムの中ではもっとも記号主義に近い立場をと

る。つまり、ノードのひとつ一つが記号主義でいう記号を表現する。このようなネットワーク上の情報表現を「局所表現 (localized representation)」といい、ひとつのノードがひとつのまとまった概念（単語、意味要素など）を表現する¹⁰⁾。

2.1.2 並列分散処理

このアプローチ、並列分散処理 (parallel distributed processing: PDP) では、概念（語義）はもはや構造化をもつことはなく、単純な要素（微細特徴：micro feature）の組み合わせで表現される¹¹⁾。このような情報表現を分散表現 (distributed representation) と呼ぶ。この分散表現の利点をヒントン¹²⁾にならって以下に挙げる。

- i) 人間は内容が部分的であったり、時には部分的に誤っているような記述からも、ある概念項目を想起できるということが知られているが、人間の記憶がもつこのような内容指定記憶 (content addressing memory) 的性質を容易に実現できる。
- ii) 人間の記憶は、純粹の記憶と、推論の結果再構成された「作話」的記憶の区別が明確ではないという性質をもつが、これを微細特徴間の結合強度のパターンの変化、いわば微細推論 (microinference) でうまく説明できる。
- iii) 学習した知識の汎化を、分散表現上でのサブパターンの変化として自然に実現できる。
- iv) 新しい概念の形成が、ネットワーク上での新しい安定的なパターンの形成として実現でき、また概念の分化も安定パターンの分化という形で同様に扱うことができる。

2.1.3 人工神経系

今までのモデルは、外部の物理世界とのインタフェースとは全く切り離され

た「認知空間」で行われるものであった。しかし、神経科学にも精神科学にも精神的背景をもつコネクショニズムは、感覚器官を通じた外部世界の知覚と、言語の理解、推論などの認知機構との相互作用のモデルを提供できる人工神経系 (artificial neuro system) の可能性がある。

ダイアラーのグループ¹³⁾では、視覚入力とその自然言語表現とを、学習を通じて写像する人工神経系の計算モデルを構築している。

2.2 スキーマの概念

現代の認知科学において、もっとも大事な概念のひとつがスキーマである。スキーマは、認知を理解するための基本構成要素となっている。PDP グループのラメルハートが PDP モデルの研究を始めたのは、スキーマの難しさに取り組み、並列分散処理によるスキーマの解釈を研究し始めたことがきっかけであった⁷⁾。

2.2.1 スキーマの歴史

スキーマ概念の変遷は、文献⁷⁾に、簡潔によくまとめられている。ここに平井有三氏の翻訳による該当部分 (文献⁷⁾ p.377, l.19~p.378, l.3) を若干の修正を加えて引用させていただく。

歴史を通してスキーマは謎に覆われた概念であった。その言葉のカント¹⁴⁾の用い方は刺激的であったが、理解が困難であった。バートレット¹⁵⁾の用い方は、その曖昧さゆえ長い間非難されてきた。ピアジェ¹⁶⁾はスキーマという言葉を用いたが、そのことに関するピアジェ自身の見方に一貫した解釈を見出すことは難しい。

歴史の大部分を通じて、スキーマの概念は余りにも曖昧なものとして実験心理学者の主流によって否定されてきた。その結果、スキーマの概念は1970年代半ばまで大きく避けられていた。スキーマの概念

は、計算機上での明確に規定された実現や、より明確に規定したスキーマの解釈を提案する試みによってよみがえったのである。

このようにして、ミンスキー¹⁷⁾はフレームという概念を仮定し、シャンクとアーベルソン¹⁸⁾はスクリプトという概念に焦点を当て、ポプロウとノーマン¹⁹⁾それにラメルハート²⁰⁾はスキーマの明確な概念を作り出した。その詳細は異なっているが、いずれも本質的には同じ考えである。

2.2.2 スキーマの特性

さて、スキーマのもつ注目すべき特性を次にまとめておこう⁶⁾。スキーマには、

- i) 変数やデフォルト値に対応する概念を表せる
- ii) スキーマは、内部構造をもつ
- iii) スキーマは、部屋・物語・誕生パーティなどの高次の概念や、複雑な概念を表せる規模をもつ

などの特性があり、これらが表現できるスキーマモデルの構築を目標にすべきである。

2.3 PDP モデルにおけるスキーマ

おそらく最初に考えられるのは、スキーマの概念をユニットの概念に写像することであろう。しかしこの考えでは、スキーマを概念化の強力な道具とする多くの特性がなくなってしまう⁷⁾。つまり、単一のユニットにはスキーマがもつ複雑な概念は表せないのである。ユニットが対応するのは、ヒントン²¹⁾が微細特徴と呼んでいる比較的単純な特徴にしすぎない。スキーマの概念を正当化するには、個々のユニットを超えて全体を眺める必要がある。スキーマを単一のユニットや小規模回路としてではなく、ネットワーク全体がもつ性質としてみなければならぬのである⁷⁾。

ここで扱うスキーマモデルは、拘束条件充足ネットワークと名付けられた PDP モデルとして表せる。そこで次に、拘束条件充足問題の取り扱いを文献⁶⁾に沿ってまとめておく。

2.3.1 拘束条件充足問題

拘束条件充足問題を考えるに当たって、われわれ自身の知識と行動との関係から始めよう。われわれは、すでにさまざまな状況に対する知識を持っている。そして、この知識を使って行動している。つまり、知識とは拘束条件の集りであると考えられる。

この場合、拘束条件充足問題の解とは、可能な限り多くの拘束条件を同時に充足するものである。しかし、すべての拘束条件を完全に充足する解があるとは限らない。むしろ、できるかぎり多くの拘束条件を同時に充足する解こそ求める解である。

つまり、拘束条件がどの程度満足されているかを表す量を扱う必要が生じる。そこで、この量をユニットの活性量に対応させることにすると、活性量は連続値をとらなくてはいけないことが分る。

2.3.2 ユニットの活性量とユニット間の結合

拘束条件充足問題を PDP モデルとして表現する²²⁾ためには、次のように考えればよい。

- i) 仮説をユニットに対応させ、各ユニットの活性量は仮説がどの程度真であるかを表すものとする。
- ii) 仮説同士の関係、すなわち、拘束条件をユニット間の結合の重みと考える。
 - 仮説 A と仮説 B が同時に真であるべきものであれば、仮説 A, B に対応するユニット A, B の間には正の結合がある。
 - 仮説 A, B が両立しないものであればユニット A, B 間には負の結

合があると考える。

iii) 拘束条件充足の重要さの度合いは、結合の強さとして表現する²³⁾。

2つのユニットについて考えてみると、結合が正ならば、両方の活性量が大きければ大きいほど拘束条件は充足の度合いが大きく、逆に、結合が負ならば、2つのユニットの活性量が共に大きければ大きいほど、拘束条件の充足の度合いが小さくなり、適合度は低くなる。したがって、ユニット*i*の活性量を $a[i]$ 、ユニット*i*からユニット*j*への結合の重みを $w[i,j]$ と表せば、ユニット*i*によって課された拘束条件をユニット*j*がどれだけ充足しているかは、

$$w[i,j] a[i] a[j], \quad (2.1)$$

で表すことができる。

2.3.3 外部入力の扱い

拘束条件は、ネットワークに最初から与えられているものだけでなく、ネットワークの外部からの入力も拘束条件として扱えるようにしておきたい。これによって、特定の仮説が真という条件のもとで、どういう解があるかを知ることができる。

外部からの入力の扱いには2通りある。1つは、外部入力の大小にかかわらず、その有無によってのみ、仮説の真偽を決定する方法である。

もう1つの扱いは、外部入力の大小をそのまま仮説の真偽に反映させる方法である。この場合には、他の弱い拘束条件と同じように、外部入力だけでは仮説の真偽を決定しない。外部入力の大きさに応じて仮説の真偽に影響を与えるだけになる。後者の場合、外部から与えられた拘束条件がどの程度充足されているかは、ユニットの活性量と外部入力の積として表わすことができる。

2.3.4 バイアスの役割

ある仮説は真になりやすい、あるいは偽になりやすいという傾向があると想

定するのが妥当な場合もある。仮説自身に与えられたこのような拘束をモデルに反映させるために、各ユニットがバイアスを持つと仮定しよう。

バイアスが正なら、その仮説が真であるほうが望ましいし、バイアスが負の場合には仮説が偽であるほうが望ましい。バイアスに関しても外部入力の場合と同様に、ユニットの活性化量との積によって、拘束条件がどの程度充足されているかを表すことができる。

ここでいうバイアスは、ユニットの興奮条件を与える「しきい値」と対応関係がある。バイアスが高ければ高いほどユニットは興奮しやすくなるが、しきい値は高いと興奮しづらくなる。すなわち、バイアスとしきい値とは、逆の相関があることが分かる。これを式で表わすと、

$$bias[i] = -threshold[i], \quad (2.2)$$

と与えられる。ここで、第 i 番目のユニットに対するバイアスを $bias[i]$ 、しきい値を $threshold[i]$ とした。

2.3.5 適合度関数

ある状況が拘束条件をどれだけ充足しているかを示す度合いを「適合度 (goodness of fit)」と呼ぶと、拘束条件充足問題を解くということは、この適合度を最大にする状況 (活性化量の分布パターン) を見つけることといえる。適合度は次の 3 種類の拘束条件がどの程度充足されているかによって決定される。

- i) ネット入力：他のユニットとの間に与えられている拘束条件
- ii) バイアス：ユニットに与えられている拘束条件
- iii) 外部入力：外部から与えられた拘束条件

よって、特定のユニットの適合度 $goodness[i]$ は現在のネット入力と、そのユニットの活性化量との積

$$goodness[i] = net[i] a[i], \quad (2.3)$$

で与えられる。ここで、ネット入力 $net[i]$ には、次の定義を用いる。

$$net[i] = \sum_j w[i, j] a[j] + input[i] + bias[i]. \quad (2.4)$$

(2.4)式を用いて(2.3)式を書き換えると、ユニット i の適合度は次のように定義される。

$$goodness[i] = \sum_j w[i, j] a[i] a[j] + input[i] a[i] + bias[i] a[i], \quad (2.5)$$

ネットワーク全体としての適合度は、上記の各ユニットに対する適合度の総和として定義されるから、最終的に得られる適合度は

$$goodness = \sum_{ij} w[i, j] a[i] a[j] + \sum_i input[i] a[i] + \sum_i bias[i] a[i], \quad (2.6)$$

から計算できる。

あるユニットに対するネット入力为正ということは、そのユニットへの拘束条件によって、そのユニットの活性量が大きくなることを意味する。逆に、ネット入力を負ならば、ユニットの活性量は小さくなる。

つまり、ネット入力为正ならばそのユニットの活性量が大きくなり、ネット入力を負であれば活性量が小さくなるような、活性値更新の仕組みが与えられていけばよい。そうすると、ユニットの適合度は次第に大きくなり、個別のユニットの適合度の単純な和である全体の適合度もまた、常に増加してゆく。こうした考察はホップフィールド⁴⁾によってなされた。

2.3.6 適合度の増加の限界

実際には、適合度は一般に無限には増加しない。ユニットが活性量の最大値または最小値になりうるのだから、ある点を越えて活性量を増やし続けることは一般にはできない。その結果、全体の適合度も増え続けることはできない。

むしろ、ユニットが最大または最小値に至るまで適合度を増やし続ける。それゆえ、各ユニットは全体の適合度をけっして減少させないように振る舞う。

こうして、全体の適合度は各ユニットが最大値になるまで、あるいは、全てのユニットに対するネット入力がちょうど0になるまで増え続ける。このとき、システムは変化するのをやめ、適合度関数は最大値、つまり、拘束条件充足問題の解を見出したということになる。

適合度関数がこうしたピークに達したとき、適合度はもはや変化できず、ネットワークは安定状態に至ったといえる。厳密にいうと、この解の状態は、適合度関数の大域的ではなく局所的な最大値になったことしか保証しない。

すなわち、これは「山登り」法であり、システムが適合度関数のピークを見つけたことしか保証せず、最も高いピークを見つけることは保証していない。拘束条件充足問題の解は、この関数を最大にする活性量の集合である。

第3章 PDP型スキーマモデル

この節では、PDPモデルにおけるスキーマモデルの詳細について解説する。

3.1 スキーマモデルの特徴記述

スキーマモデルは最も単純な拘束条件充足モデルのひとつであるが、それにもかかわらず、すべての拘束条件充足モデルの動作に対する有益な洞察を与えてくれる。

ここでは、スキーマモデルとして、次に挙げた仮定をおくことにする。

i) スキーマモデルは決定論的である。

確率論的に決まるのではなく、0から1までのいかなる値もとりうる。

ii) 結合マトリクスは対称であり、自分自身には結合しない。

(すなわち、 $w[i, j] = w[j, i]$ であり、かつ、 $w[i, i] = 0$)

iii) ユニットの更新は非同期的に行う²⁴⁾。

更新するユニットはランダムに選び出される。そして、バイアスがラ

ンダムに決定され、ユニットに対するネット入力（ $net[i]$ ）が計算されたのち、ユニットの活性化量が更新される。

3.2 スキーマモデルの更新規則

山登り法のロジックによって示されるのは、ネット入力（ $net[i]$ ）が正であればユニットの活性化量を増やし、負ならば活性化量を減らすということである。それゆえ、つぎのような単純な更新ルールを用いることにする。

$net[i]$ が 0 より大きいとき、

$$a[i](t+1) = a[i](t) + net[i](t) \{1 - a[i](t)\}, \quad (3.1)$$

$net[i]$ が 0 より小さいとき、

$$a[i](t+1) = a[i](t) + net[i](t) a[i](t), \quad (3.2)$$

で与える。(3.1)式の意味は、次の通りである。ネット入力がなければ初期活性化量のままで変化しないが、ネット入力があれば、右辺第2項の効果でネット入力に比例して増大する。ただし、活性化量が1に近づくと第2項の効果はなくなる。一方、(3.2)式が(3.1)式と異なっている点は、ネット入力があった場合の振る舞いである。(3.2)式によると、ネット入力があれば、その値に比例してユニットの活性化量を減少させる。

以上をまとめて書き直すと、

$$a[i](t+1) = (1 - |net[i](t)|) a[i](t) + net[i](t) \theta(net[i](t)), \quad (3.3)$$

となる。ここで、 $\theta(x)$ は階段関数

$$\theta(x) = \begin{cases} 1, & \text{for } x > 0 \\ 0, & \text{for } x < 0 \end{cases} \quad (3.4)$$

である。

ここで、(3.3)式の意味と振る舞いを述べておく。一般の場合、第1項は減衰項であり、ユニットの活性量を指数関数的に減衰させる。これに対し、第2項は外力項に相当し、ネット入力为正の場合のみ働き、強制的に活性量を上昇させるように働く。

この式は、1つのパラメータ（ネット入力： $net[i]$ ）をもつ離散時間系の力学方程式とみなせるが、ネット入力が0の場合に活性量（ $a[i](t)$ ）は不定（初期活性量のまま）となり、ネット入力が1の場合は初期活性量以外はすべて1を与える。それ以外の場合、活性量の不動点の条件は、

$$a'[i](t+1) = a'[i](t), \quad (3.5)$$

で与えられ、不動点は

$$a'[i](t) = \theta(net[i](t)), \quad (3.6)$$

となる。つまり、ネット入力が負なら0、正なら1である（ネット入力が0の場合は上記のように不定）。代表的な値に対して、この活性化規則のシミュレーション結果を図1a～図5bに示した。

その特徴をまとめると、

- i) ネット入力の絶対値が1未満の場合は、不動点に向かって単調に近づくだけ（図1a, 図2b）である。
- ii) ちょうど1の場合は、初回の更新でいきなり不動点に落ち着く（図2a, 図2b）。
- iii) ところが1を超えると、減衰振動になる（図3a, 図3b）。
- iv) ネット入力が2の場合は、2周期振動を起こす（図4a, 図4b）。
- v) 2を超えると、減衰項は負性抵抗として働き、2周期解のままで発散していく（図5a, 図5b）。

したがって、活性量が不安定化する条件は

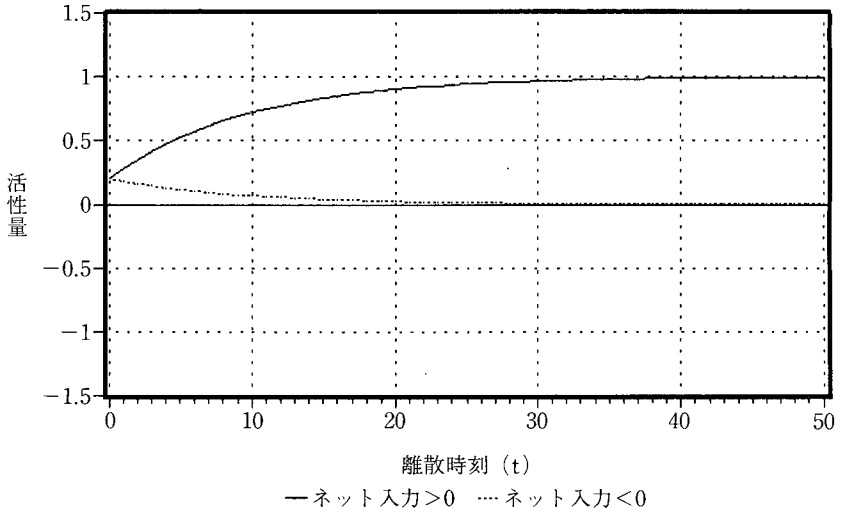


図1a. ユニット活性化規則のシミュレーション
(初期活性量=0.2, ネット入力の絶対値=0.1)

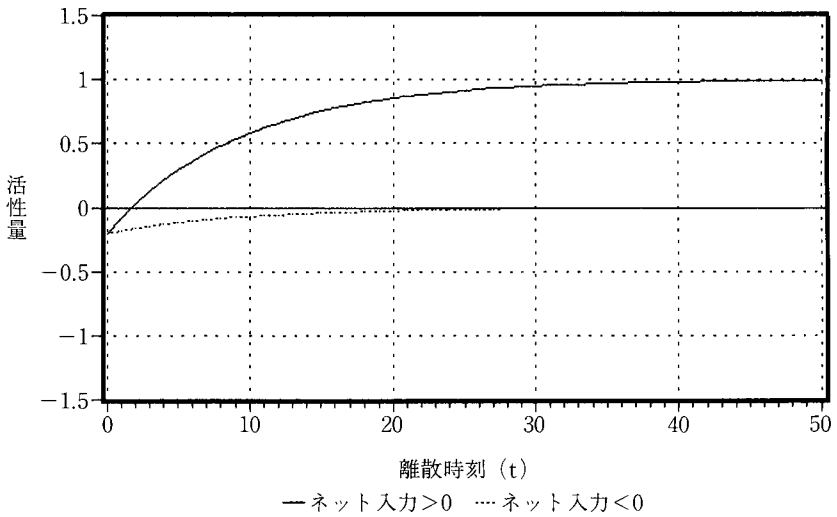


図1b. ユニット活性化規則のシミュレーション
(初期活性量=-0.2, ネット入力の絶対値=0.1)

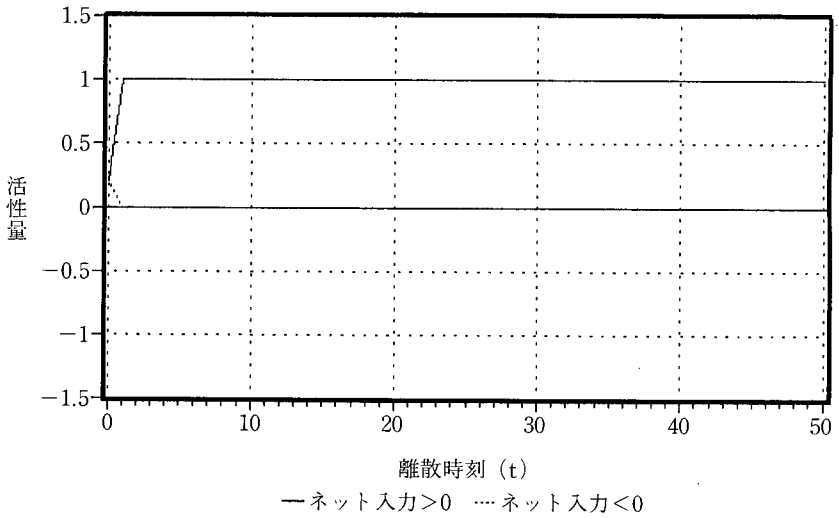


図2a. ユニット活性化規則のシミュレーション
(初期活性化量=0.2, ネット入力の絶対値=1)

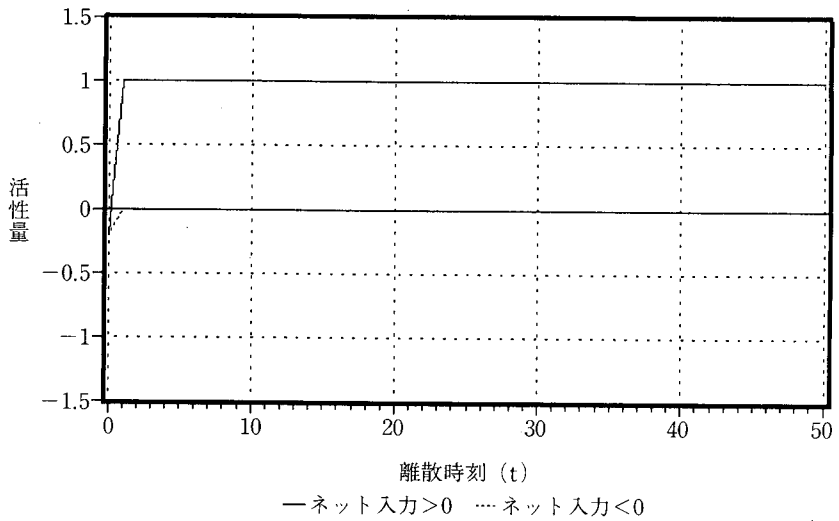


図2b. ユニット活性化規則のシミュレーション
(初期活性化量=-0.2, ネット入力の絶対値=1)

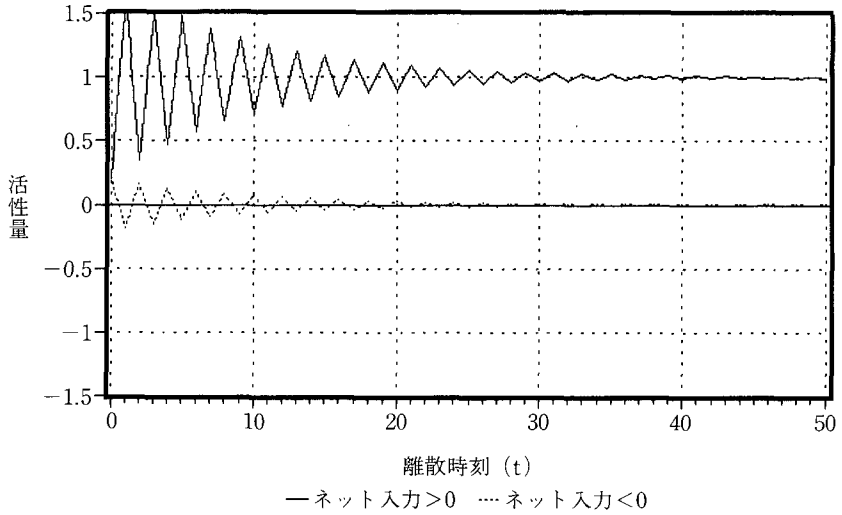


図3a. ユニット活性化規則のシミュレーション
(初期活性化量=0.2, ネット入力の絶対値=1.9)

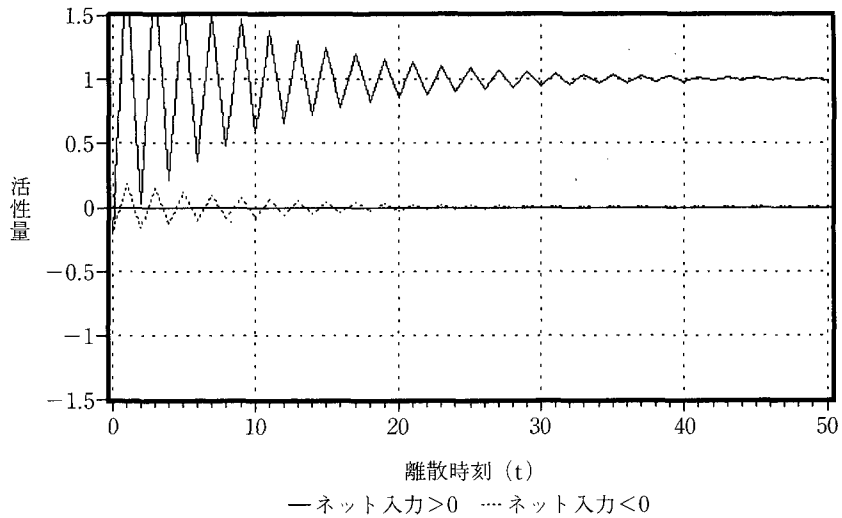
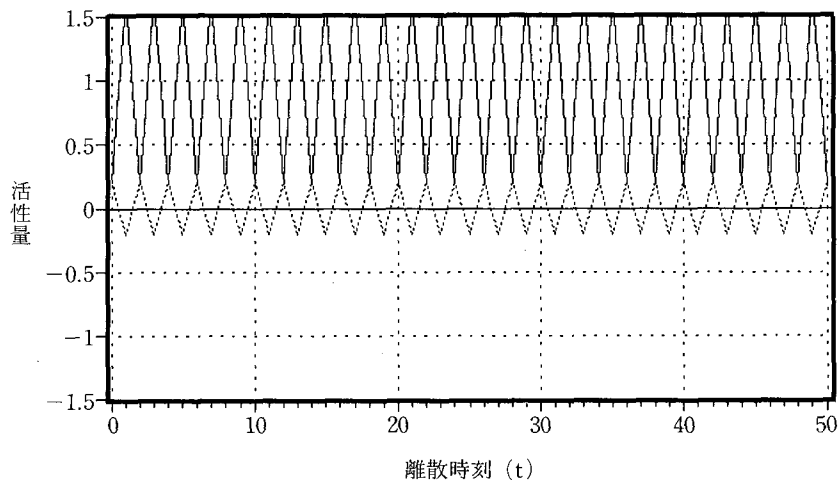
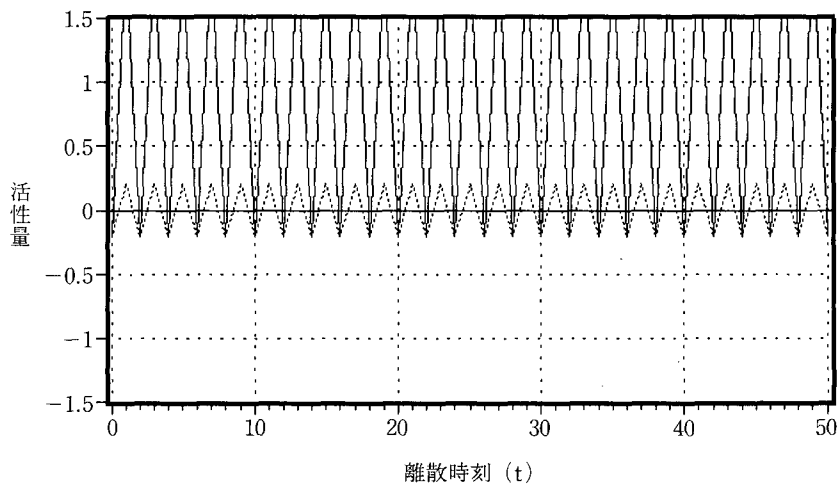


図3b. ユニット活性化規則のシミュレーション
(初期活性化量=-0.2, ネット入力の絶対値=1.9)



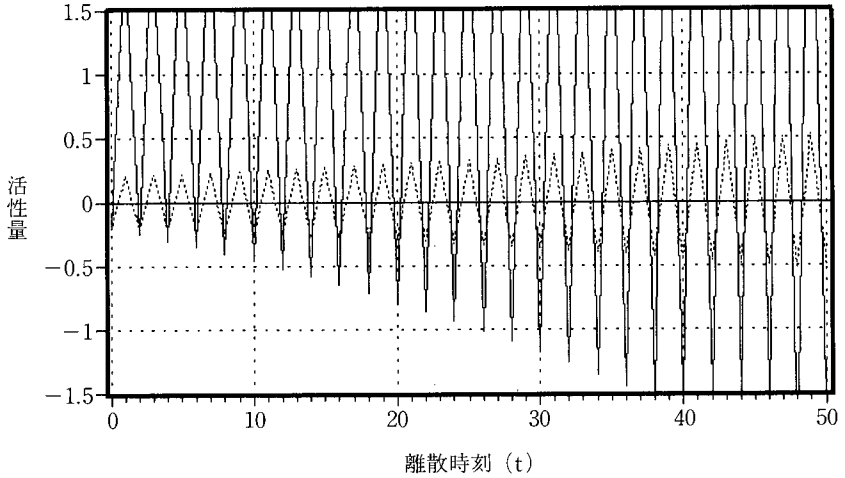
— ネット入力 > 0 --- ネット入力 < 0

図4a. ユニット活性化規則のシミュレーション
(初期活性化量=0.2, ネット入力の絶対値=2)



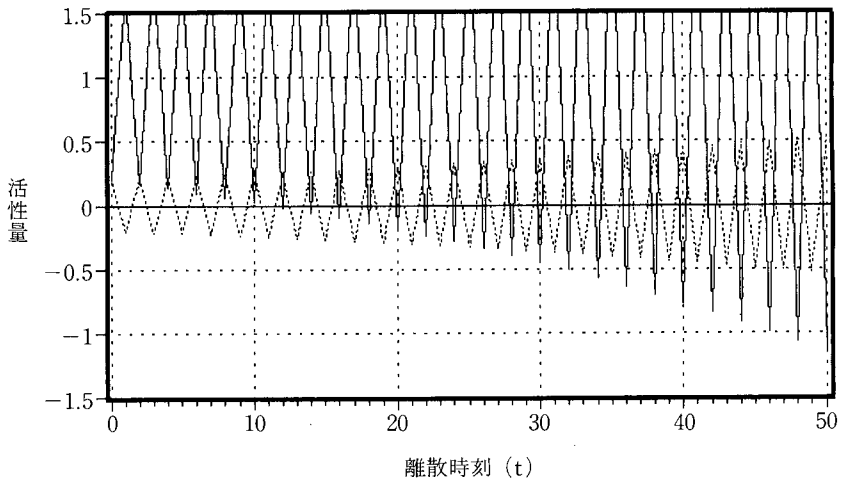
— ネット入力 > 0 --- ネット入力 < 0

図4b. ユニット活性化規則のシミュレーション
(初期活性化量=-0.2, ネット入力の絶対値=2)



— ネット入力 > 0 ... ネット入力 < 0

図5a. ユニット活性化規則のシミュレーション
(初期活性化量=0.2, ネット入力の絶対値=2.02)



— ネット入力 > 0 ... ネット入力 < 0

図5b. ユニット活性化規則のシミュレーション
(初期活性化量=-0.2, ネット入力の絶対値=2.02)

$$|net[i]| > 2, \quad (3.7)$$

で与えられるが、これは、減衰項の係数 $(1 - |net[i](t)|)$ の絶対値が 1 を超えて、活性化が不安定化する条件に他ならない²⁵⁾。

さて、ネット入力とは 3 つのソースから得られる：あるユニットに隣接するユニット、そのユニットのバイアス、そのユニットに対する外部入力の 3 つである。これらの和をとって、次の式を得る。

$$net[i](t) = \sum_j w[i, j] a[j](t) + bias[i] + input[i]. \quad (3.8)$$

スキーマモデルのユニット結合係数には、総和規則 (sum rule) が成立している。この総和規則は、ユニット間の協調・競合、すなわち、スキーマ内ユニットの協調と、スキーマ間ユニットとの競合とのバランスを表わす。総和規則の具体的な表式は、次章のネッカーキューブの例で与える。

第 4 章 スキーマモデルのネッカーキューブへの応用

この章では、スキーマモデルのネッカーキューブへの応用について述べる。ネッカーキューブは競合するスキーマがわずか 2 つしかないため、スキーマモデルのなかでは最少のスキーマをもつもっともシンプルなモデルである。

ここでの議論は、ラメルハートたち⁷⁾によるが、同じ原理で少し異なったネットワークによってネッカーキューブの問題を解析する方法には、フェルドマン²⁶⁾がある。また、少し異なった枠組みによる議論がスモレンスキ²⁷⁾によって与えられている。

4.1 ネッカーキューブの定義

ネッカーキューブとは、**図 6** で示されたような線画による箱の透視図をいう。この図は平面に描かれた線画であるが、見ているうちに立方体に見えてくる。しかし、立方体の見え方には 2 通りあることにすぐに気が付こう。重な

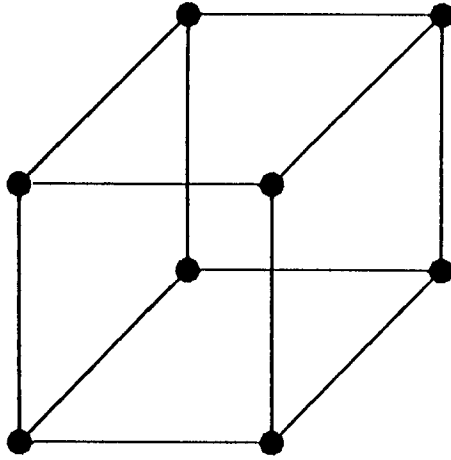


図6. ネッカーキューブ

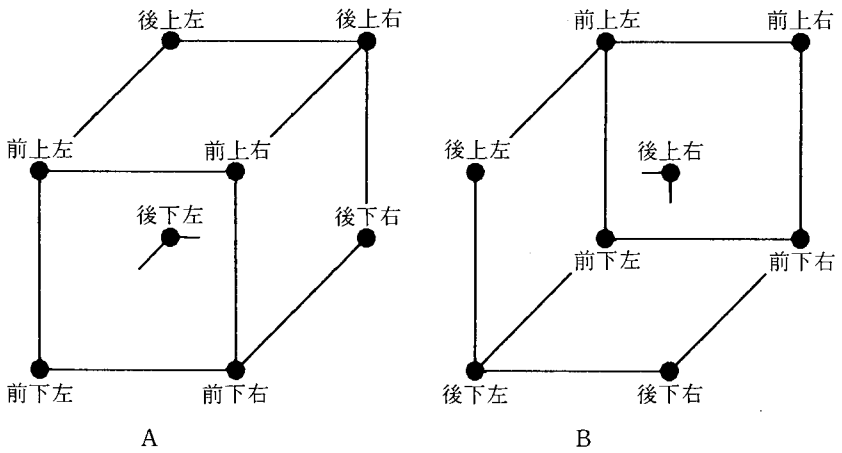


図7. ネッカーキューブの2つのスキーマ

った2つの正方形のうち、左側にある方を前面にあると見るか、あるいは逆に後面にあると見るか、両方の見方がありうる。

この2つの見方を、相反する2つのスキーマに対応づけよう。これを図示したのが図7である。今後、左下の正方形が前面にあるとみなすスキーマをAスキーマ、後面とするスキーマをBスキーマと呼ぶことにする。

ここで、ネッカーキューブのスキーマモデルとの関係について述べよう。スキーマモデルをネッカーキューブに応用するには、2つの競合するネットワーク（AとBスキーマに対応する）を構成すればよい。そして、各々のスキーマに対応するネットワーク内では、構成要素のユニット間に協調関係を設定すればよい。すなわち、図7にある合計16個の頂点をユニットとし、スキーマ間の競合とスキーマ内の協調とを、ユニット間を結ぶ結合係数として与えてやればよいことになる。それぞれのユニットは、ネッカーキューブの各頂点に対する可能な仮説に対応している。

そして、ネッカーキューブがAスキーマに見えたとすると、ネットワークを構成するユニットのうち、Aスキーマに属するユニットのすべてが興奮し一方、Bスキーマ側はすべて興奮しないという状態に対応する。

4.2 ユニット結合と総和規則

さて、ネッカーキューブに対応するネットワークを構成するには、ユニット間の結合係数を決める必要がある。そこでまず、1つのユニットに着目し、このユニットがほかのユニットとどのような協調・競合関係を満たせばよいかをみていくことにする。

Aスキーマの「前上右」ユニットから見た結合関係の模式図を図8に示した。まず、同じAスキーマに属する他のユニットとの結合は、最隣接のユニット間のみに限って興奮性（正の結合係数）とする。つまり、Aスキーマの「前上右」という仮説が真ならば、最隣接ユニットについては、左側は「前上左」、下側は「前下右」、斜め右上は「後上右」という仮説がそれぞれ真である

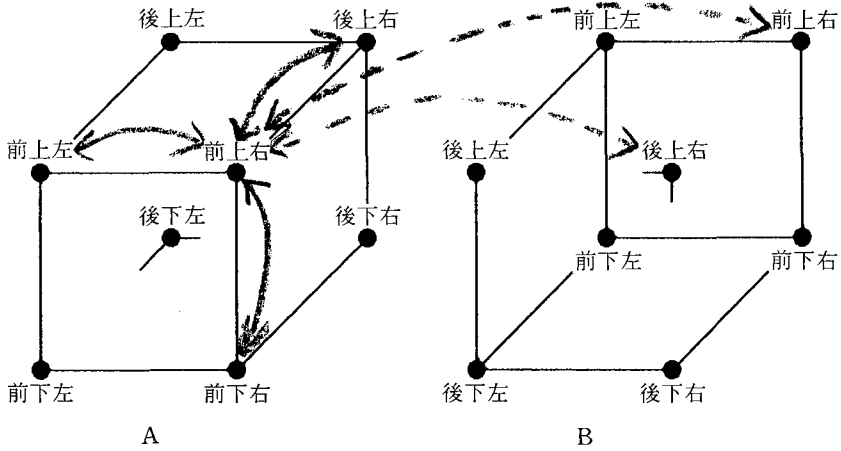


図8. ネッカーキューブのユニット間結合

Aスキーマの「前上右」ユニットから見たユニット間結合。実線の矢印は協調，破線は競合を表す。

ことを意味する。

次に、Bスキーマに属するユニットとの結合は、競合関係をもつとする。今の場合、注目しているAスキーマの「前上右」と競合するBスキーマの仮説は、異なるスキーマであるにもかかわらず同じ解釈を与えるBスキーマの「前上右」と、同形の場所で同じ位置関係を与えるBスキーマの「後上右」の2つである。これは、Aスキーマの「前上右」が真であれば、Bスキーマの「前上右」、「後上右」の両方の仮説とも偽であることを意味する。

さて、注目しているAスキーマの「前上右」のユニットが興奮するか抑制されるかは、結合しているすべてのユニットからのネット入力

$$\begin{aligned}
 net[i] &= \sum_j w[i, j] a[j] \\
 &= \sum_{j=\{A\}} w[i, j] a[j] + \sum_{j=\{B\}} w[i, j] a[j], \quad (i=\{A\}) \quad (4.1)
 \end{aligned}$$

の正負で決まる。ここで、バイアスと外部入力を無視し、第*i*要素のユニット

は A スキーマに属するとした。

もし、今結合している 5 個のユニット (A スキーマに 3 個, B スキーマに 2 個) がすべて興奮し、活性化量 $a[j]=1$ になっていると仮定すると、第 i ユニットの完全な矛盾に陥り、興奮できなくなるはずである。これは、上式が 0 に他ならないことを意味する。

よって、結合係数についての次の総和規則がいえことになる。

$$\sum_{j=\{A\}} w[i, j] = - \sum_{j=\{B\}} w[i, j], \quad (i = \{A\}) \quad (4.2)$$

同じ A スキーマどうしの結合係数を c とおくと、上式は $3c$ になる。したがって、A-B 間の結合係数は、 $-3c/2$ である。同様に、第 i ユニットが B スキーマに属する場合も考慮すると、結局、総和規則は次のようにまとめられる。

$$\sum_{j=\{A\}} w[i, j] = - \sum_{j=\{B\}} w[i, j] = 3c, \quad (i = \{A\}) \quad (4.3a)$$

$$\sum_{j=\{B\}} w[i, j] = - \sum_{j=\{A\}} w[i, j] = 3c. \quad (i = \{B\}) \quad (4.3b)$$

4.3 スキーマモデルによるシミュレーション

この節では、スキーマモデルをネッカーキューブに適用し、シミュレーションを行う。スキーマモデルにおけるユニット活性化量の更新規則は、(3.3)式と(3.8)式とで与えられる。ここで、非同期的更新を採用し、外部入力は 0 とおく。そして、ユニットをランダムに選び、平均値 0 の乱数によってバイアスを設定し、1 回に 1 ユニットの更新を行うことにする。

ここで、ユニット活性化量の定義域については、ラメルハートたち⁷⁾、および、マクレランドとラメルハート⁶⁾によって提供されているシミュレーションソフトに従って、1 を超えないものとした。このため、活性化規則のシミュレーションで議論した減衰振動解や発散振動解に相当するパラメータ領域においては、1 を超える活性化量に更新される可能性があるが、今の場合は、それを強制的に安定な不動点である 1 へ落とし込むことになる。いわば、強制的な安

定化条件を課したことに相当する。この強制安定化条件をはずすと、カオスを含む多様なダイナミクスが現れるが、詳細は別論文²⁸⁾に譲る。

シミュレーション結果は、採用した乱数系列によって、いくつかの最終状態に分類できる。大まかにいえば次の3つに分けられる。

- i) Aスキーマのユニットがすべて興奮し、Bスキーマのユニットはすべて抑制される。
- ii) Bスキーマのユニットがすべて興奮し、Aスキーマのユニットはすべて抑制される。
- iii) Aスキーマのいくつかと、Bスキーマのいくつかが興奮し、残りはすべて抑制される。

ここで、適合度のもっとも高い安定状態は、i)とii)であり、iii)は中間的な適合度を与える準安定状態に対応する。

図9には、ii)の場合の平均活性化量の時系列データを示した。平均は、ユニ

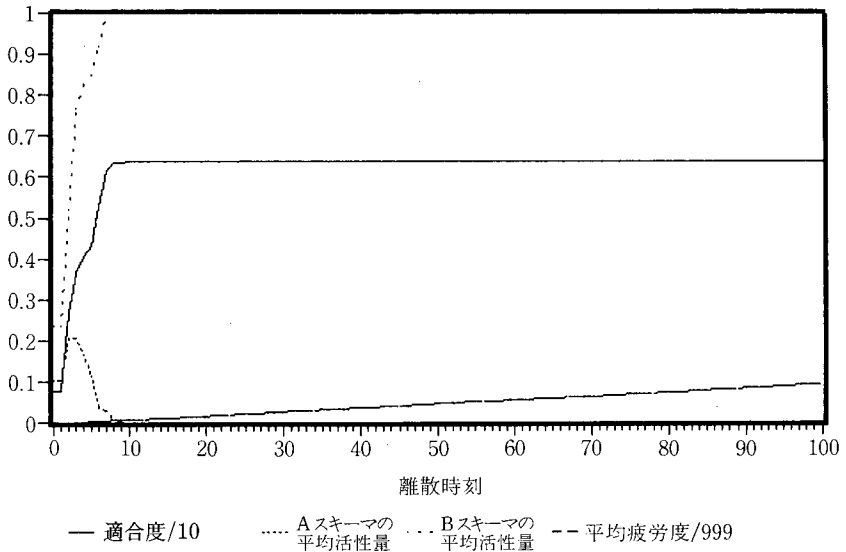


図9. スキーマモデルによるシミュレーション
(臨界活性化量=0.9, 臨界疲労度=999, ユニット結合係数 $c=0.4$)

ット活性化量をスキーマごとに行い、(2.6)式で計算した適合度とともにプロットした。ここで、同スキーマ間の最隣接ユニットどうしの結合係数は $c=0.4$ である。

この図から、次の特徴が読み取れる。

- a) 数ステップの初期経過時間の間は、A・B スキーマの両方とも協調して成長する。
- b) 初期経過時間の後は、両スキーマの競合が始まり、片方のスキーマだけが生き延びる。
- c) 不動点に達したあとは、平均活性化量に変化は見られない。
- d) ネットワーク全体の適合度は、単調に増加する。

第5章 連想型スキーマモデル

スキーマモデルの面白さは、局所的なユニットの結合係数から、大域的なスキーマ構造が現れる点にある。しかしながら、前章で明らかになったスキーマモデルによるシミュレーションは、認知科学的には単純すぎて、あまり面白みのあるものではない。そこで、局所的なメカニズムによって、大域的に2つのスキーマを相互連想する連想型スキーマモデルの構築を検討し、2つのスキーマ間を相互連想的に活性化する連想型スキーマモデルを提案する。

5.1 ユニットへの疲労効果の導入

相互連想のメカニズムを実現するためには、安定な不動点が不安定化する要因を付加する必要がある。ここではその駆動力を、安定な不動点に落ち着いたというあくまで局所的な情報に求めることにする。すなわち、個々のユニットが安定な不動点に居続けるサイクル数に応じて、ユニットに疲労効果を取り入れる。

期待される疲労効果をまとめると、

- i) ユニットの興奮が継続すると、疲労が蓄積される。

- ii) 疲労の蓄積がしきい値を超えると、ユニットは強制的に抑制される。
- iii) ユニットの再興奮には、回復期間が必要である。

(回復期間を超えないと、興奮可能にならない)

となる。この条件を満たすもっとも簡単な方法は、個々のユニットに新たな内部状態(疲労度)を設け、そのダイナミクスを決めてやることである。

ここではインプリメントが簡単なように、疲労度のダイナミクスを決定するアルゴリズムは次のように決める。

- a) ユニットの「疲労度」は、疲労が蓄積され始める「臨界活性化量」と、「臨界疲労度」、「活性フラグ」によって支配される。

「臨界活性化量」は、「疲労度」の蓄積を開始する臨界値である。「臨界疲労度」は、「活性フラグ」をオンからオフにする臨界値である。「活性フラグ」は、強制的な興奮の抑制とその解除を支配する。

- b) 「活性フラグ」がオンのときのみ、ユニットは興奮できる。
- c) 「活性フラグ」の初期状態はオンである。
- d) 「疲労度」は、ユニットが臨界活性化量を超えて1単位時間のあいだ興奮すると1増える。

すなわち、ユニットの「疲労度」とは、活性化量が「臨界活性化量」を超えたサイクル数を意味する。

- e) 「疲労度」が「臨界疲労度」を超えると、「活性フラグ」をオフにする。
- f) 「活性フラグ」がオフになると、ネット入力に関係なくユニット更新時に0を出力する。
- g) 「活性フラグ」がオフで、更新の結果ユニット活性化量が「臨界活性化量」以下になれば「疲労度」を1減らす。

- h) 「疲労度」が0以下になれば、「活性フラグ」はオンになる。

すなわち、「疲労度」の値は疲労回復に必要な更新サイクル数に等しい。

上のアルゴリズムで注意すべきは、「疲労度」の増減は、ユニット更新のタイミングで行われる点である。ユニット更新は、ランダムに選択されたユニッ

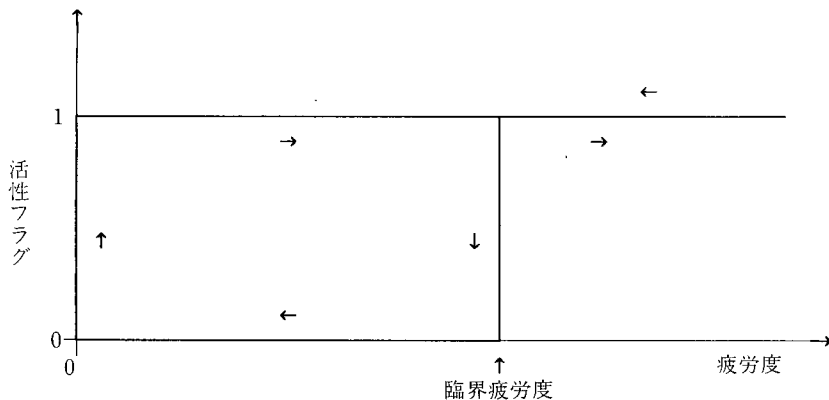


図10. 活性フラグのヒステリシス・ループ
 (「活性フラグ」対「疲労度」)

トに対して行われるため、このアルゴリズム自体の中には、大域的に同期しやすい傾向は一切ない。むしろ大域的には、「疲労度」のバラつきが大きくなりやすい傾向がある。

この疲労度ダイナミクスは、「活性度」に対して「活性フラグ」をヒステリシス・ループ上で動かすことに相当する。これを図10に示した。

5.2 連想型スキーマモデルによるシミュレーション

この節では、前節のアルゴリズムを組み込んだ連想型スキーマモデルのシミュレーション結果を示し、議論する。はじめに、比較的短時間に相当する200単位時間(200サイクル)の時系列プロファイルを示し、次に長時間(1024サイクル)のプロファイルを示して議論する。

5.2.1 短時間の時系列プロファイル

前節のアルゴリズムを組み込んだ連想型スキーマモデルのシミュレーション結果を、図11a～図14bに図示する。いずれもユニット結合係数 $c=0.4$ である。

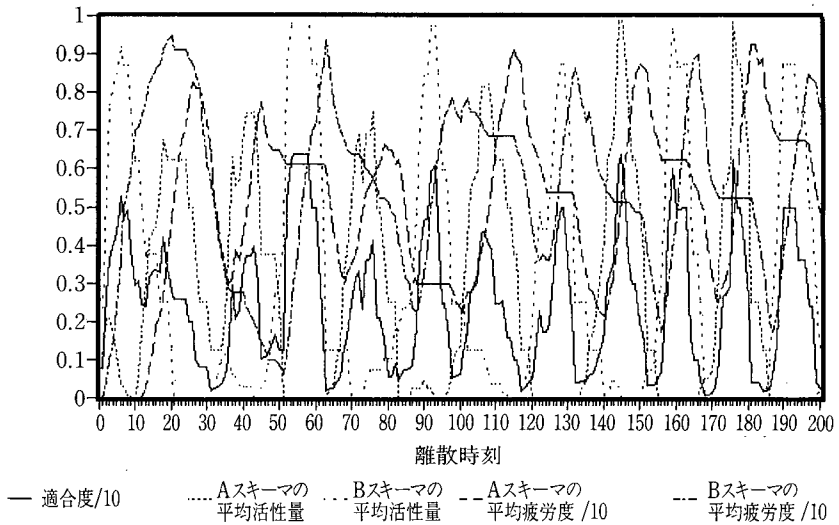


図11a. 連想型スキーマモデルによるシミュレーション
 (臨界活性化量=0.9, 臨界疲労度=10, ユニット結合係数 $c=0.4$, 平均想起率=0.02)

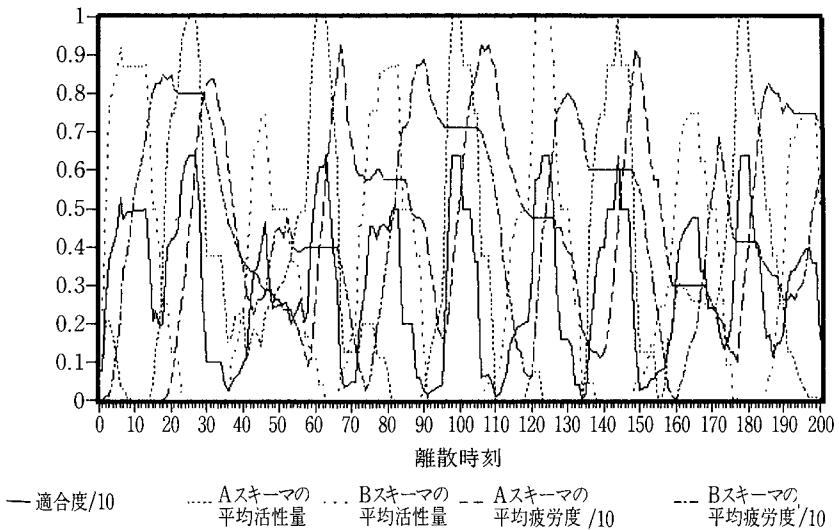


図11b. 連想型スキーマモデルによるシミュレーション
 (臨界活性化量=1.0, 臨界疲労度=10, ユニット結合係数 $c=0.4$, 平均想起率=0.05)

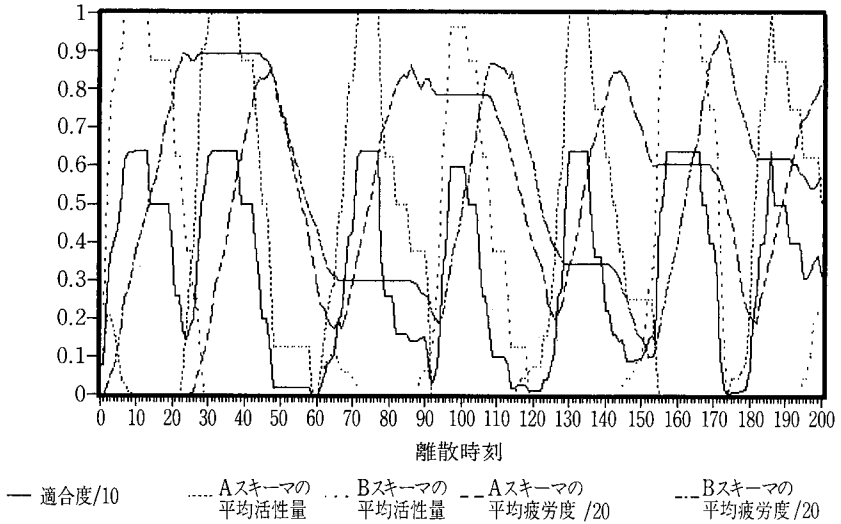


図12a. 連想型スキーマモデルによるシミュレーション

(臨界活性化量=0.9, 臨界疲労度=20, ユニット結合係数 $c=0.4$, 平均想起率=0.14)

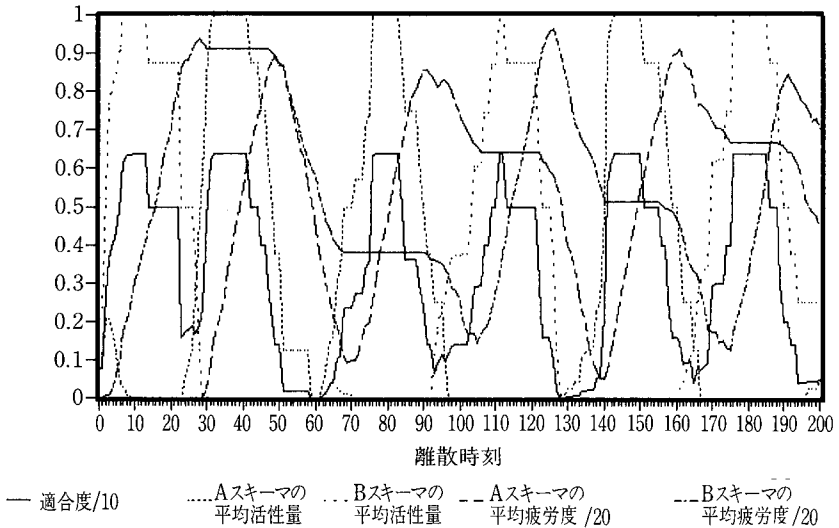


図12b. 連想型スキーマモデルによるシミュレーション

(臨界活性化量=1.0, 臨界疲労度=20, ユニット結合係数 $c=0.4$, 平均想起率=0.20)

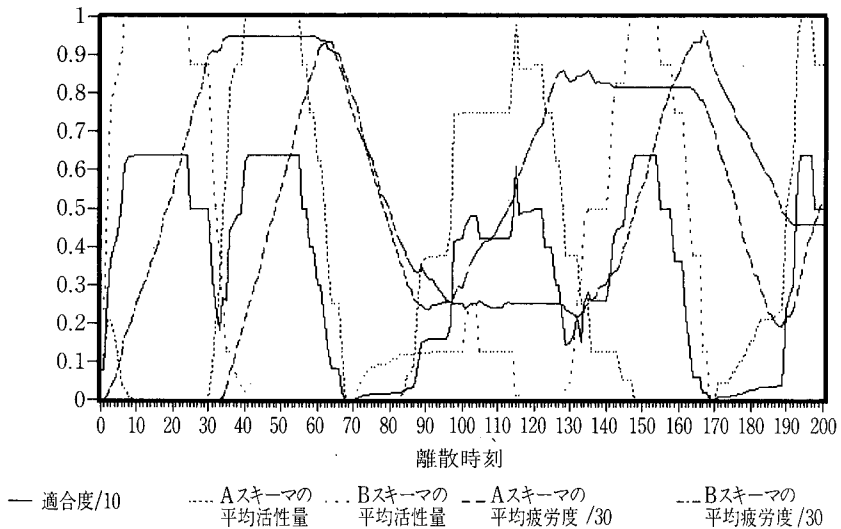


図13a. 連想型スキーマモデルによるシミュレーション

(臨界活性化量=0.9, 臨界疲労度=30, ユニット結合係数 $c=0.4$, 平均想起率=0.20)

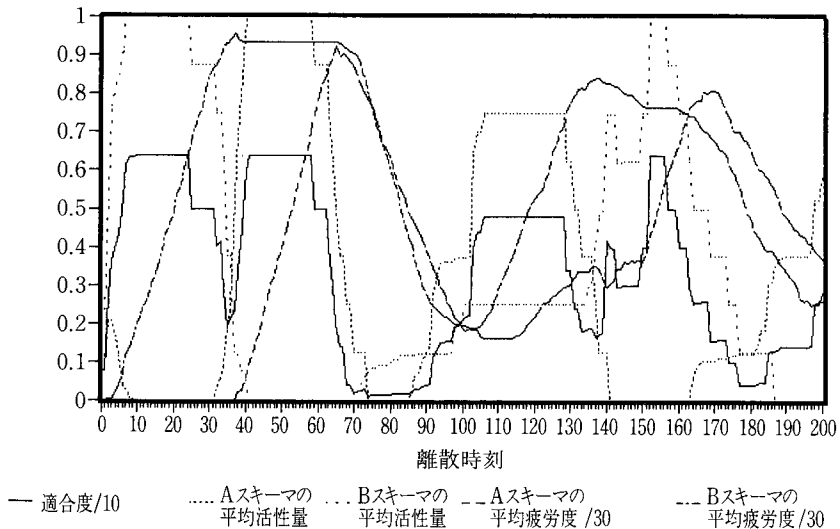


図13b. 連想型スキーマモデルによるシミュレーション

(臨界活性化量=1.0, 臨界疲労度=30, ユニット結合係数 $c=0.4$, 平均想起率=0.19)

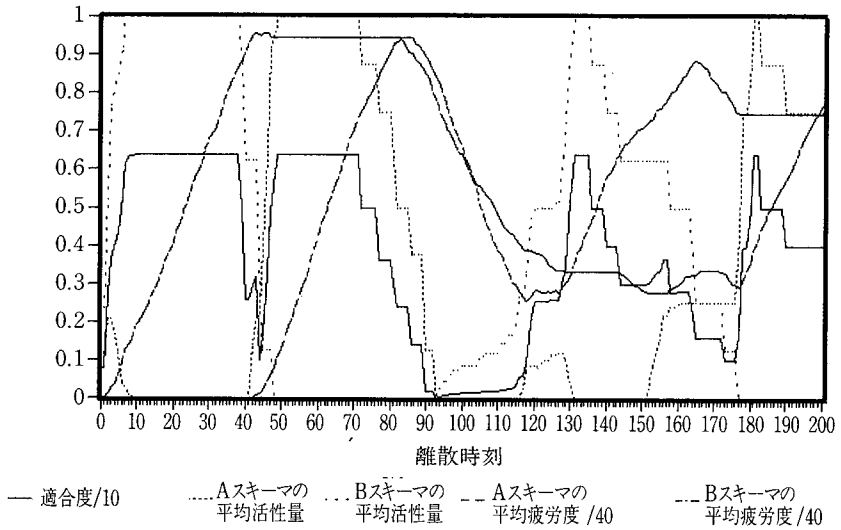


図14a. 連想型スキーマモデルによるシミュレーション

(臨界活性化量=0.9, 臨界疲労度=40, ユニット結合係数 $c=0.4$, 平均想起率=0.27)

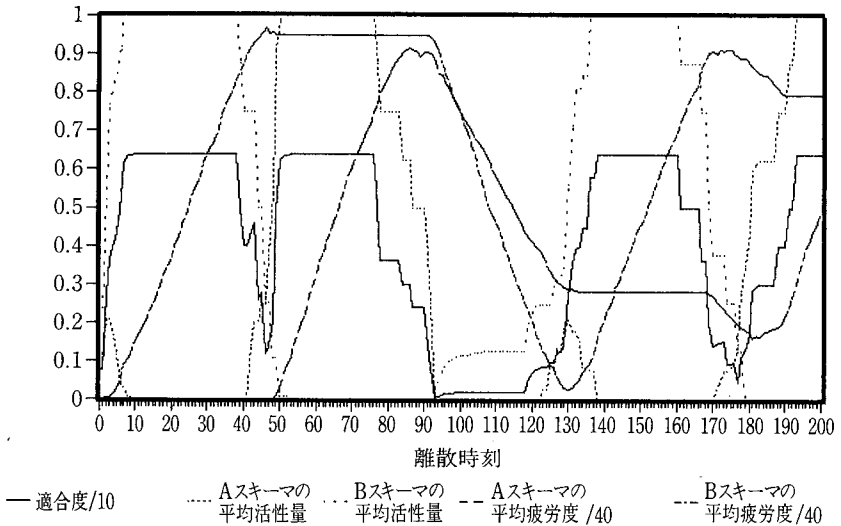


図14b. 連想型スキーマモデルによるシミュレーション

(臨界活性化量=1.0, 臨界疲労度=40, ユニット結合係数 $c=0.4$, 平均想起率=0.41)

ここで、平均想起率とは、観測時間に対する完全想起時間（A・B どちらかのスキーマを完全に想起している時間）の割合である。すなわち、平均想起率とはスキーマの交代想起の成功率のことであり、平均想起率が高ければ高いほど、スキーマの交代想起がうまくいっていることを意味する。

これらの図から特徴的なことを挙げると、次のようにまとめられる。

- i) 臨界活性量は、1に近い方が平均想起率が高い。
- ii) 臨界疲労度が、大きい方が平均想起率は高い。
- iii) 臨界活性量の効果よりも、臨界疲労度の効果の方が顕著である。
- iv) 初期の推移時間の終わり頃、両スキーマの疲労度がほぼ同時に落ち込んでいる。
- v) 初期の推移時間を除けば、順序だった交代想起が成功している。
(あるいは、少なくともその傾向がでている)
- vi) 大域的な想起の安定性が保たれている。
(途中で完全想起に失敗しても、そのあと回復する)

これらのなかで特徴的なことは、最後に挙げた大域的な交代想起の安定性である。前節で述べたように、取り入れたユニットの疲労効果は、あくまで局所的なものにしか過ぎず、大域的な安定性まで保証するものではなかった。したがって、この大域的な安定性の起源は、ユニット間結合にしかありえない。やはり局所的なユニット間結合だけで、大域的安定性が生まれたのは、ユニット結合係数が十分安定な不動点を作る条件に合致しているためである。

5.2.2 長時間の時系列プロファイル

短時間の時系列プロファイルによると、もっとも大域的な安定性の悪いデータは、図11aに示したプロファイルである。平均想起率はわずか0.02で、臨界活性量0.9、臨界疲労度10であった。このプロファイルを長時間で見たとき、平均想起率が悪化していくのか、それとも悪化しないのかをここで調べてみよう。

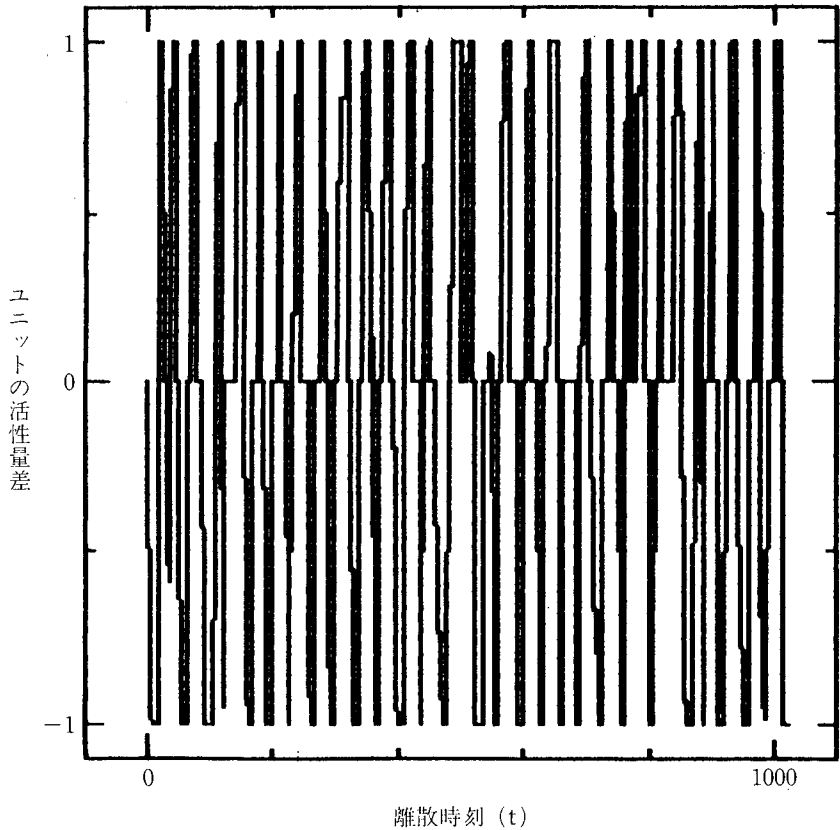


図15a. 抑制性結合で結ばれたユニットの活性量差の
長時間プロフィール

(臨界活性量=0.9, 臨界疲労度=10, ユニット結合係数 $c=0.4$)

ユニットの活性量=1のとき, Aスキーマの「後上左」ユニットが興奮し,
ユニットの活性量=-1のとき, Bスキーマの「前上左」ユニットが興奮して
いる。

はじめに, 各スキーマに属する特定のユニットを1つずつ取り出し, それらの活性量の差を1024サイクルにわたって図15aにプロットした。Aスキーマに属するユニットには「後上左」を, Bスキーマは「前上左」を採用し, 両者の差を縦軸に取った。したがって, 縦軸で1の値はAスキーマ側の特定ユ

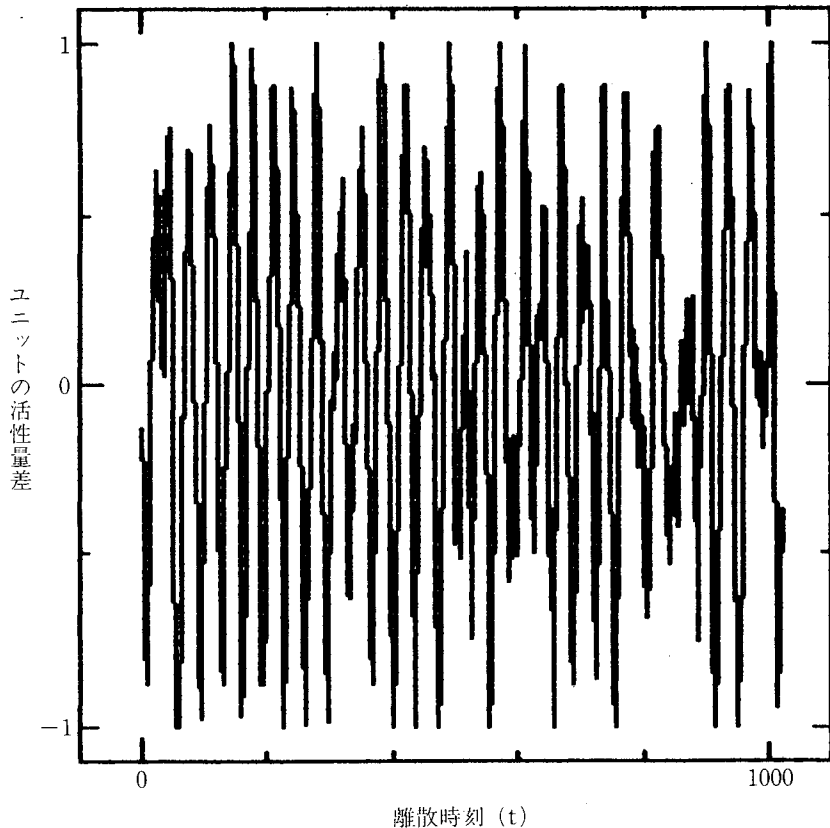


図15b. A・Bスキーマの平均活性量差の長時間プロフィール

(臨界活性量=0.9, 臨界疲労度=10, ユニット結合係数 $c=0.4$)

ユニットの活性量=1のとき, Aスキーマの全ユニットが興奮し,
 ユニットの活性量=-1のとき, Bスキーマの全ユニットが興奮している。

ユニットが興奮してBスキーマは抑制, -1はBスキーマ側が興奮してAスキーマが抑制されていることになる。これを見ると, 長時間にわたる交代想起が非常にうまくいっていることが分かる。

次は, 同じパラメータだが特定のユニットを取り出すのではなく, 各スキーマ内で平均した活性量を図15bに示した。図15aの特定ユニットと比較する

と、明らかに平均想起率は落ち込んでいる。これは、図15a が互いに直接抑制性の結合をしているユニットどうしのプロファイルであったため、比較的に抑制性の相関が強く効いていたからである。これに対して図15b では、直接の抑制性結合がなく相関が弱くて比較的独立に動いているユニットの影響も、平均操作のなかに丸め込まれているため、平均想起率は小さくなっている。

しかしながら、完全想起に失敗したあとでも、再び完全想起を回復するようすが見られる。この場合もやはり、長時間にわたる交代想起には成功している。

第6章 まとめと議論

はじめに、PDP 型スキーマモデルを概観し、最少の競合するスキーマをもつネッカーキューブを例に、ダイナミクスの解析を行った。そして、局所的な疲労効果をユニットに導入することで、大域的なスキーマの交代想起が可能であることを示した。

本論文では、ユニットの活性量を1以下のものに限定し、1を超えれば強制的に1へ引き戻して固定していた。しかし、前章の議論から除外された領域にこそカオスを含めた多様なダイナミクスが隠されている。この点については、現在研究中であり、詳細については今後の課題であるが、予備的な解析結果の報告²⁸⁾は本紀要の別論文として掲載されている。

また、ユニットに新しい内部自由度（内部状態）を導入し、これを用いてヒステリシスを実現したが、甘利（[29], p. 80-81.）によれば、自己結合型のユニットによってヒステリシスを再現できることが指摘されている。この点を考慮したモデルも検討に値しよう。

—謝 辞—

本論文の作成に当たって、以下のコンピュータとソフトウェアに大変お世話

になった。ここに感謝の意を込めて記載し、礼を述べさせていただく。パーソナルコンピュータ「PC-9801NS20（日本電気社製）」。シミュレーション用ソフトの開発には、「LOTUS 1-2-3 Ver. 2.2 J（米 Lotus Development 社製）」、「Turbo C Ver. 2.0（米 BORLAND 社製）」、および、「N 88-日本語 BASIC (86) Ver. 6.0（日本電気社製）」。数式処理プログラム、「DERIVE Ver. 2（米 Soft Warehouse 社製）」。グラフ作成と数値計算のサポートに、オンラインソフトウェアの「Ngraph Ver. 5.（石坂 智, JUNET: isizaka@nsis86.cl.nec.co.jp; NIFTY-Serve: NAH01761)」。論文の打ち込みには「Mifes Ver. 5.0（メガソフト社製）」、清書は、オンラインソフトウェアのテキストフォーマッタ「ntf Ver. 2.0（新島 智之, ASCII-pcs: pcs14235; NIFTY-Serve: MGG01464)」。

参考文献

- 1) Rosenblatt, F.: “Two theorems of statistical separability in the perceptron”, In *Mechanisation of thought processes: Proceedings of a symposium held at the National Physical Laboratory*, November 1958, Vol. 1, London: HM Stationery Office, (1959), pp. 421-456.
- 2) Rosenblatt, F.: “Principles of neurodynamics”, New York, Spartan, (1962).
- 3) Minsky, M., and Papert, S.: “Perceptrons”, Cambridge, MA, MIT Press. [斉藤正男訳「パーセプトロン」東京大学出版会, 1971]
- 4) Hopfield, J. J.: “Neural networks and physical systems with emergent collective computational abilities”, *Proceedings of the National Academy of Science*, Vol. 79, (1982), pp. 2554-2558.
- 5) 活性化量と出力の関数には大きく分けて2通りある。一つは、活性化量をそのまま出力するものであり、もう一つはしきい値活性化量がある程度のレベルに達しないときには出力しないものである。
- 6) McClelland, James L., and Rumelhart, David E.: “Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises”, In *Computational Models of Cognition and Perception*, (ed.) Jerome A. Feldman, Patrick J. Hayes and David E. Rumelhart, The MIT Press, Cambridge, (1988), Chap. 3.
- 7) Rumelhart, D. E., Smolensky, P., McClelland, J. L., and Hinton, G. E.: 「PDP モデルにおけるスキーマと直列思考過程」, In PDP モデル (認知科学とニューロン回路網の探索), D.E. ラメルハート・J.L. マクレランド・PDP リサーチグループ, 甘利

- 俊一監訳，産業図書，(1989)，第9章，pp. 367-419. [In *Parallel Distributed Processing*, Rumelhart, D. E., McClelland, J. L., and the PDP Research Group, The MIT Press, Cambridge, MA, Vol.2, Chapter 14, (1986).]
- 8) Dyer, M. G.: "Symbolic neuroengineering for natural language processing: A multilevel research approach", *Technical Report UCLA-AI-88-14*, Artificial Intelligence Laboratory, (1988), UCLA.
 - 9) 往住彰文: 「心の計算理論」東京大学出版会，1991年，第4章。
 - 10) Cottrell, G. W.: "A connectionist approach to word sense disambiguation", (1989), Pitman.
 - 11) McClelland, J. L., and Kawamoto, A. H.: "Mechanisms of sentence processing: Assigning roles to constituents", In J. L. McClelland, D. E. Rumelhart, and the PDP Research Group, *Parallel distributed processing: Explorations in the micro-structure of cognition*, Vol. 2, (1986), MIT Press.
 - 12) Hinton, G. E., McClelland, J. L., and Rumelhart, D. E.: "Distributed representations", In D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, *Parallel distributed processing: Explorations in the microstructure of cognition*, Vol. 1, (1986), MIT Press.
 - 13) Nenov, V. I., and Dyer, M. G.: "DETE: Connectionist/Symbolic model of visual and verbal association", *Proceedings of IEEE 2nd Conference on Neural Networks*, (1988), San Diego.
 - 14) Kant, E.: "Critique of pure reason", 2nd ed., N. Kemp Smith, Trans., London, Macmillan, (1963). (Original work published 1787) [篠田英雄訳: 「純粹理性批判」(上), (中), (下) 岩波文庫, (1961-1962); 原佑訳: 「純粹理性批判」(上), (中), (下) (高坂正顕・金子武蔵監・原佑編: 「カント全集」第4,5,6巻) 理想社, (1981).]
 - 15) Bartlett, F. C.: "Remembering", Cambridge, England, Cambridge University Press, (1932). [宇津木保・辻正三訳: 「想起の心理学」誠信書房, (1983).]
 - 16) Piaget, J.: "The origins of intelligence in children", New York, International University Press, (1952). [波多野完治・滝沢武久訳: 「知能の心理学」みすず書房, (1967).]
 - 17) Minsky, M.: "A framework for representing knowledge", In P. H. Winston (ed.), *The psychology of computer vision*, New York, McGraw-Hill, (1975), pp. 211-277. [白井良明・杉原厚吉訳: 「知識を表現するための枠組」(『コンピュータビジョンの心理』所収) 産業図書, (1979).]
 - 18) Schank, R. C., and Abelson, R. P.: "Scripts, plans, goals, and understanding", Hillsdale, NJ, Erlbaum, (1977).
 - 19) Bobrow, D. G., and Norman, D. A.: "Some principles of memory schemata", In D. G. Bobrow and A. Collins (eds.), *Representation and understanding: Studies in cognitive*

science, New York, Academic Press, (1975), pp. 131-149. [淵一博監訳「人工知能の基礎 (知識の表現と理解)」近代科学社, (1978).]

- 20) Rumelhart, D. E.: "Notes on a schema for stories", In D. G. Bobrow and A. Collins (eds.), *Representation and understanding*, New York, Academic Press, (1975), pp. 211-236. [淵一博監訳「物語の構図についてのノート」(『人工知能の基礎』所収) 近代科学社, (1978).]
- 21) Hinton, G. E.: "Implementing semantic networks in parallel hardware", In G. E. Hinton and J. A. Anderson (eds.), *Parallel models of associative memory*, Hillsdale, NJ, Erlbaum, (1981), pp. 161-188.
- 22) 並列ネットワークを使って拘束条件充足問題を解くというアイデアを最初に示したのは、ヒントン [Hinton, G. E.: "Relaxation and its role in vision", Unpublished doctoral dissertation, (1977), University of Edinburgh.]である。
- 23) ここでの問題は「最も重要な拘束条件をできるだけ多く充足するような状況を見出すこと」であるから、扱われる拘束条件も必ず充足されねばならないような強いものではなく、充足されるのが望ましい「弱い拘束条件 (weak constraints)」 [Blake, A.: "The least disturbance principle and weak constraints", *Pattern Recognition Letters*, Vol. 1, (1983), pp. 393-399.]である。
- 24) 活性量の更新ルールに関する問題点—同期/非同期更新⁶⁾
ここにちょっとした問題があることに注意されたい。2つのユニットが同時にネット入力を評価されている次のような場合を考えよう。
 - (1) どちらのユニットもオフであり、両者の間には大きな負の重みがあるとする。
 - (2) 各ユニットが小さな正のネット入力をもっていると仮定する。この場合どちらのユニットもオンになる。しかし、負の結合によって結合しているから、どちらもオンになるやいなや、全体の適合度は減少する。次にユニットが更新されるとき、今度はどちらもオフになる。こうして、オン・オフの循環が繰返される。
この問題に対しては、次の2つの解決法がある。
 - (1) 「非同期的」更新：標準的な解決は一回には1つ以上のユニットの更新を許さないことである。この場合、ユニットのいずれか一方がオンになり、他方がオンになるのを妨げることになる。
 - (2) 「同期的」更新：ユニットの活性量を非常にゆっくりと増加させる。これにより、ユニットはお互いにオンになりつつあるということを「感じる」ことができ、適切なバランスを得ることができる。
- 25) 参考文献⁶⁾によると、ここで与えた活性化規則 (3.3式) は自然と活性量を0から1の間に拘束すると書かれているが、ネット入力が1を超える値をもつような特殊な場合は、明らかに活性量も1を超えることに注意すべきである。
- 26) Feldman, J. A.: "A connectionist model of visual memory", In G.E. Hinton and J.

- A. Anderson (eds.), *Parallel models of associative memory*, Hillsdale, NJ, Erlbaum, (1981), pp. 49–81.
- 27) Smolensky, P.:「動的システムにおける情報処理 (調和理論の創設)」In PDPモデル (認知科学とニューロン回路網の探索), D.E. ラメルハート・J.L. マクレランド・PDP リサーチグループ, 甘利俊一監訳, 産業図書, (1989), 第6章, pp. 214–281. [In *Parallel Distributed Processing*, Rumelhart, D. E., McClelland, J. L., and the PDP Research Group, The MIT Press, Cambridge, MA, Vol.1, Chapter 6, (1986).]
- 28) 阪井和男・片山硬・大岩幸太郎・和田悟:「平均場近似によるPDP スキーマモデルのカオスダイナミクスの理論」, 明治大学教養論集, 自然科学, 1992年3月 pp. 67–104。
- 29) 甘利俊一:「神経回路網の数理 (脳の情報処理様式)」産業図書, 1978, 第4章。