

構造化ボルツマンマシンによる既学習知識の統合*

奥野 拓^{*1}, 嘉数 侑昇^{*2}

Integration of Learned Knowledge by Structured Boltzmann Machines

Taku OKUNO and Yukinori KAKAZU

This paper proposes a framework for integrating independently learned knowledge by Boltzmann machines. Auto-associative neural networks based on distributed representation have some advantageous properties of dealing with knowledge. However, as to the supplement of knowledge under a dynamically changing environment, or the merging of independently acquired knowledge, they do not work well enough because of difficulty in supplementary learning. Therefore we propose a model for integrating knowledge by connecting independently learned network modules without additional learning. First, localized representation, which is indispensable for realizing interaction between knowledge in different modules, is introduced. Although it will spoil the flexibility of distributed representation, competitive learning adopted for translating representations preserves it to some extent. Secondly, the configuration of the connected network is explicated, and its behavior is illustrated. It is completely realized by activation, competition, and inhibition of Boltzmann machine. Finally, to demonstrate the effectiveness of the proposed model, results of computer simulation on a navigation problem of obstacle avoidance are shown.

Key Words: Artificial Intelligence, Learning, Neural Network, Boltzmann Machine, Competitive Learning, Knowledge Processing

1. 緒 言

近年、ニューラルネットワーク、Q-Learning⁽¹⁾、あるいは遺伝的アルゴリズムに代表されるように、学習により戦略的に対象問題領域の持つ知識を獲得し、それを積極的に利用して問題を解くアプローチが注目されている⁽²⁾⁽³⁾。これらの手法をインプリメントした問題解決システムは、シミュレーションの実行をその前提とする。すなわちシステムはモデル化した問題世界(環境)における問題解決のための、試行(行動)、その試行結果の評価、およびその際の環境の状態記述を、逐次的に実行する。一般に問題解決システムは逐次的に実行したこれらの組の中から一連の望ましい組を、状態・行動対形式の知識として獲得し、システムの内部状態として蓄積する。蓄積されたこれらの知識は問題解決に有効利用されることになる。

しかしながら学習というものは、それが行われた環境への強い依存性を持つ。そのため、学習後に大きな

環境変動がある場合には、一般に以前学習した結果の有用性は保証されず、このため新たに再学習を行う必要が生じる。これは追加学習問題として知られており、これまでに多くの研究がなされてきた⁽⁴⁾。ところが、一般に学習は時間的コストがかかるアプローチであり、またニューラルネットワーク等のように学習項目が明示的に表現されないような場合には、効果的な再学習法が得られていないという現状が存在する。

以上に述べたような、環境変動の存在を前提とした学習問題に対して、ここでは新しく既学習知識の統合という観点からのアプローチを試みる。すなわち、異なる環境において個別的に学習、蓄積された知識群を統合し、問題解決システムが取り扱うべく入力される未経験の新たな環境を認識し、そこでの問題解決に向けて統合化された知識の有効利用を計ろうとする方法である。

以下では、このようなアプローチの一つとして、連想記憶モデルとしてのボルツマンマシン⁽⁵⁾のダイナミクスを利用して、再学習を行わずに既学習知識を統合するネットワークモデル(構造化ボルツマンマシン、SBM)を構築し、種々の検討を加えた結果を報告す

* 原稿受付 平成5年11月1日。

^{*1} 正員, (株)ジャパンテクニカルソフトウェア (〒060 札幌市中央区北4条西12丁目)。^{*2} 正員, 北海道大学工学部 (〒060 札幌市北区北13条西8丁目)。

る。このモデルは、特定の環境において個別に学習されたネットワーク（モジュール）を結合し、その競合・抑制作用を利用して、変化後の環境における入力に対して既学習知識群中で最も有効であるような知識を想起する機構を持つネットワークである。前述したように本アプローチでは、このようなネットワークのダイナミクスを利用して、個々の既学習知識群のみでは対応しきれないような環境変化に対し、複数の既学習知識群に含まれる有効な部分知識群による補完という方法をとる。

以降の章では最初に、結合されたモジュール間における既学習知識間の相互作用に必要な局所表現を導入し、分散表現から局所表現への表現変換を競合学習⁽⁶⁾によって行う枠組みを示す。そして、このような表現変換機構を含む、ボルツマンマシンによる個別学習の単位となるモジュールの構造、学習方法、および学習後の動作について述べる。次に、複数の既学習モジュールを結合して統合ネットワークを構築し、環境からの入力に対するネットワーク各部の動作を順を追って示す。最後に、構築したネットワークの挙動を計算機シミュレーションによって示す。ここでは、簡単な移動物体の障害物回避ナビゲーション問題を例題として用い、異なる成功例を学習した複数のモジュールを結合して新たな問題に適用した結果により、本モデルの有効性を検証する。

2. モジュールの学習

2.1 局所表現の導入 上で述べた学習的アプローチにボルツマンマシンを用いる利点として、汎化能力、部分的な記憶内容の破壊に対するロバスト性、変数やデフォルト値を表現的な拡張をせずに同一の枠組みで自然に扱えること等が挙げられる⁽⁶⁾。これらの特性は、記憶項目がネットワークの結合の重みによって分散的に保持され、項目固有の記憶場所を持たないような表現方法である分散表現⁽⁷⁾に起因するものである。分散表現されたネットワーク間の相互作用は、ユニット間に付加的な結合を与えることによって実現できる。しかしながら、このようにして実現される相互作用は、ユニット間、すなわち記憶項目を構成する属性値間の相互作用であり、記憶項目間の相互作用にはならない。これは、個々のユニットが複数の記憶項目の表現に関与するため、ユニット間の相互作用の重ね合せによって特定の記憶項目間の相互作用は実現できないという理由によるものである⁽⁶⁾。したがって、学習された知識間の相互作用によって知識の統合を行うには、分散的に記憶された知識を局所的な表現に変換する必要が

ある。しかしながら、知識が局所的な表現に変換された時点で、分散表現の利点が失われてしまう。そこで提案モデルでは、局所表現への変換過程にニューラルネットワークの学習則の一種である競合学習⁽⁶⁾を導入し、分散表現の特性をある程度保存するようなネットワークを構築している。以下では、このような表現の変換過程を含み、学習を行う単位であるモジュールの構造と学習について述べる。

2.2 モジュールによる知識の表現

最初に、ネットワークによって表現される知識の形式を定義する。ここでは、知識の分散的な表現を属性の組によって記述する。属性は属性値の集合であるため、分散表現された知識は属性の直積集合の要素として表現される。一方、局所表現された知識は、知識という属性の要素として表わせる。したがって、 n 種類の属性によって表現される分散表現された知識とその局所表現との関係は、属性値の集合をドメイン D として次式のように表わされる。

$$D_S \subseteq D_{B_1} \times \dots \times D_{B_i} \times \dots \times D_{B_n} \dots \dots \dots (1)$$

(1) 式で表現される知識の集合が一つのモジュールによって表現される。モジュールは、(1) 式の右辺を表現するボルツマンマシン（B層）と、左辺を表現する局所（シンボル）表現ユニットの集合（S層）から構成される二層のネットワークであり、層間は競合学習によって結合される。ユニット i の状態値を s_i 、ユニット j から i への結合の重みを w_{ij} で表わすと、モジュールに含まれるユニット i への入力の総和は、次のように計算される。

$$net_i^B = c^B \sum_j w_{ij}^B s_j^B + c^{CS} \sum_j w_{ij}^C s_j^S \dots \dots \dots (2)$$

$$net_i^S = c^{CB} \sum_j w_{ij}^C s_j^B \dots \dots \dots (3)$$

ただし、 B 、 S 、 C はそれぞれ B層、S層、層間結合を表わすインデックスである。また、 c^B 、 c^{CS} 、 c^{CB} は正定数である。

モジュールのすべてのユニットは一般的なボルツマ

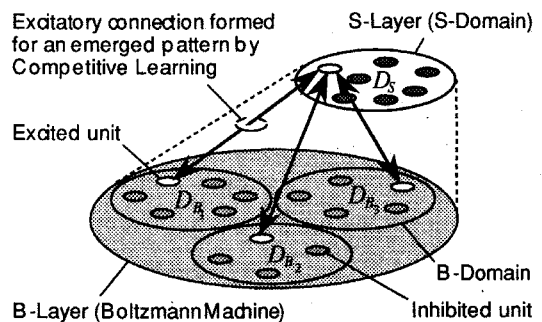


図1 学習モジュールの概念図

ンマシンの状態遷移規則⁹⁾に従って動作する。すなわち、ユニット i の状態が 1 になる確率は、 T を温度パラメータとして次式によって計算される。

$$p(s_i = 1) = \frac{1}{1 + \exp(-net_i / T)} \dots\dots\dots(4)$$

モジュールの構成およびその構成要素とドメインとの対応を図 1 に示す。個々の属性値の有無はネットワークのユニットの発火状態 ($s_i = 1$) ・非発火状態 ($s_i = 0$) に対応づけられる。以降、ドメインに対応するユニットの集合も同様にドメインと呼ぶ。

2.3 モジュールの学習と動作 モジュールの学習は以下のような手順で行われる。まず、学習させる知識群に対応するパターン群を B 層にランダムな順序で呈示し、一般的なボルツマンマシンの学習則⁹⁾に従って学習させる。十分な学習の後、B 層のネットワークをボルツマンマシンの状態遷移規則に従って自由動作させると、各パターンがほぼ学習時に呈示した確率で想起されるようになる。

次に、こうして想起されたパターン群に対し、一般的な競合学習則⁹⁾に従い層間の興奮性結合を形成する。このとき、予め S 層には十分な数のユニットを準備しておき、層間の任意のユニット間に完全結合を与えた状態で学習を行う。十分な学習の後、B 層において想起された各パターンに対して S 層の特定のユニットが一つだけ発火するようになる。学習後に、発火しなかった S 層のユニットおよびそれらに関する層間結合を取り除くことにより、S 層には、呈示パターンと同数のユニットが残る。ここで、S 層のユニットは、非発火時に層間の興奮性結合を通して B 層を抑制するための状態値として -1 を取るものとする。

学習後のモジュールは上述のように一つのボルツマンマシンとして動作させる。B 層では通常のボルツマンマシンと同様に、ユニットの状態値の固定による部分パターンからの完全パターンの想起、すなわち属性

値をキーとした知識の検索が可能である⁹⁾。一方 S 層では、B 層において既学習パターンが想起された時には、競合学習によって対応づけられた一つのユニットが発火する。さらに、競合学習の効果として、未学習パターンの想起時には複数のユニットが発火し、S 層における表現が分散化される⁹⁾。この効果により、表現変換後においても分散表現の性質をある程度保存することを可能としている。

3. 統合ネットワーク

3.1 状態・行動の表現 ここで対象としているような、状態の観測に基づく行動決定の問題において扱われるドメインは、観測によって得られる環境からの入力と、環境に対する各種の効果器の出力とに分けることができる。前者を状態ドメイン D_{state} 、後者を行動ドメイン D_{action} でそれぞれ置き換えると、式 (1) は以下のように書き換えられる。

$$D_{state} = \prod_{i=1}^m D_{B_i} \dots\dots\dots(5)$$

$$D_{action} = \prod_{i=m+1}^n D_{B_i} \dots\dots\dots(6)$$

$$D_S \subseteq D_{state} \times D_{action} \dots\dots\dots(7)$$

統合ネットワークにおけるモジュール間の相互作用は、これら 2 つのドメインを単位として考えることができるため、以下ではこれらのドメイン間に結合を与えることによって統合ネットワークを構築する。

3.2 複数の既学習モジュールの結合 統合ネットワーク (SBM) の概念図を図 2 に示す。図中のドメイン間に描かれた矢線は、ユニット間に与えられた結合の方向性を表わす。図中の入出力サイトは、モジュールの B 層から内部結合を除いたユニットの集合であり、環境からの入力および環境への出力を、結

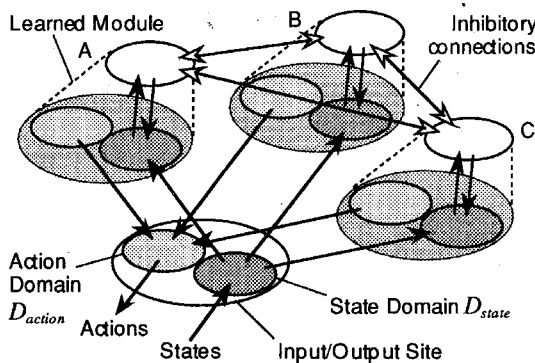


図 2 統合ネットワーク (SBM) の概念図

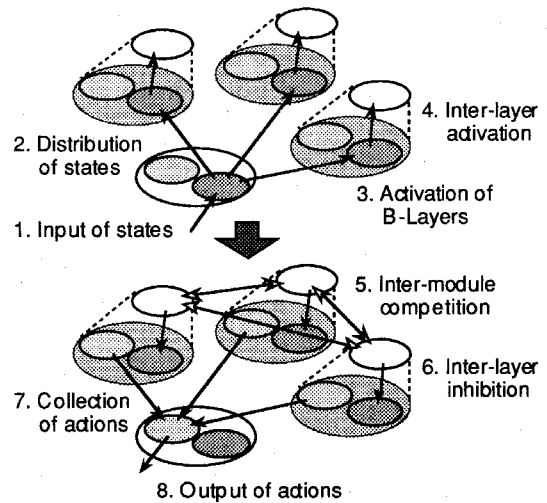


図 3 SBM の動作

合される全てのモジュールで共通とするために設けられている。入出力サイトと各モジュールは、対応するドメイン間で図のように結合され、結合の重みは、対応するユニット間で興奮性、それ以外のユニット間で抑制性としている。一方、S層間には、競合作用を行うための抑制性の結合が任意のユニット間に与えられている。この競合作用は環境からの入力のみ依存して行われるため、各モジュール内の層間結合は状態ドメインとS層との間のみ形成している。

以上のような構造をもつ統合ネットワークに含まれるユニット*i*への入力の総和は次の様に計算される。

$$net_i^{Bs} = c^B \sum_j w_{ij}^{Bs} s_j^B + c^{CS} \sum_j w_{ij}^{CS} s_j^S + c^{Is} \sum_j w_{ij}^{Is} s_j^S \dots (8)$$

$$net_i^{Ba} = c^B \sum_j w_{ij}^{Ba} s_j^B \dots (9)$$

$$net_i^S = c^{CB} \sum_j w_{ij}^{CB} s_j^B + c^S \sum_{k \neq i} w_{ij}^{Sk} s_j^S \dots (10)$$

$$net_i^{Ia} = c^{Ia} \sum_k \sum_j w_{ij}^{Ia} s_j^{Ba_k} \dots (11)$$

$$w_{ij}^{Is} = \begin{cases} 1 & \text{if } a_i^{Bs} = a_i^{Is} \\ -1 & \text{otherwise} \end{cases} \dots (12)$$

$$w_{ij}^{Ia_k} = \begin{cases} 1 & \text{if } a_i^{Ba_k} = a_i^{Ia} \\ -1 & \text{otherwise} \end{cases} \dots (13)$$

$$w_{ij}^{Sk} = -1 \dots (14)$$

ただし、Bs, Ba, Is, Iaは、それぞれB層のD_{state}およびD_{action}、入出力サイトのD_{state}およびD_{action}を表わすインデックスであり、インデックスの添え字kはk番目のモジュールを表わし、c^{Is}, c^{Ia}は正定数である。また、a_iは、ユニット*i*に対応づけられた属性値を表わす。

以上において既学習モジュールを結合するために与えたに興奮性・抑制性の結合の重みは式(12)~(14)に示されるように、それぞれ全モジュールに共通な定数値を与えており、学習内容には依存しない。また、層間結合w_{ij}^Cは競合学習により自己組織的に形成するため、結果として、SBMはボルツマンマシンに記憶されている分散表現された知識のみに基づいて動作するネットワークとして構築されている。次に、環境からの入力に対するSBMの挙動の概略を示す。

3-3 統合ネットワークの動作 SBMの動作の概念的な流れを図3に示す。環境から入出力サイトへ入力された状態値は、外部入力として各モジュールの状態ドメインへ送られる。各モジュールでは、その外部入力が状態値の固定と同様の効果を与える。すなわち、各モジュールのB層はそれぞれ独立に、外部入

力をキーとして、全体の制約を最大に充足させるようなパターンを高い確率で想起する。B層で想起されたパターンは層間結合を通してS層のユニットを発火させる(以上、図3上図参照)。

S層では、他モジュールとの間の抑制性結合により競合が起こり、全てのモジュールの中で最も高い確率で発火するユニットを含むモジュールが競合に勝ち残る。それ以外のモジュールのS層のユニットは抑制されて非発火状態となり、層間結合を通してB層のユニットを抑制する。最終的に、各モジュールの行動ドメインの発火パターンが入出力サイトに伝えられるが、競合に負けたモジュールのB層は抑制されているため、勝ち残ったモジュールの行動ドメインの想起パターンが環境への出力となる(以上、図3下図参照)。以上が動作の概略であるが、SBM全体が一つのボルツマンマシンであり、全てのユニットが非同期的に状態を更新するため、実際には各動作が順に行われるわけではない。

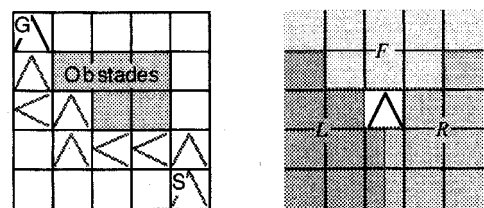
以上のメカニズムにより、SBMは、複数の既学習ネットワークから外部入力をキーとして検索を行うネットワークであることが見て取れる。ただし、ボルツマンマシンの制約充足機構⁽⁵⁾により、キーと完全に一致する項目が存在しない場合でも、既学習項目から尤もらしい結果を推論する機構を潜在的に含んでいる。

次章では、このようなSBMの挙動を、簡単な移動物体のナビゲーション問題を例とした計算機シミュレーションによって示す。

4. 移動物体のナビゲーション

4. 移動物体のナビゲーション

4-1 問題設定 ここでは、ナビゲーションのタスクとして障害物回避問題を用いる。図4aで示され



(a) マップと経路の例 (b) ゴールの方向



(c) 障害物の位置 (d) 可能な行動

図4 移動物体の障害物回避ナビゲーション問題

るようなマップ上の移動物体を考える。マップ上には、スタート (S) とゴール (G) が与えられ、障害物はマップ中のハッチング部分として与えられる。移動物体に与えられるタスクは、区画上をスタート地点から障害物を回避しながらゴール地点に移動することである。マップ上では移動物体およびその軌跡を“^”で表わす。図中の軌跡はこのマップに関する最短経路の一つを示している。移動物体の観測可能な状態は、現在の進行方向に対するゴールの方向と隣接する障害物の位置の2種類のみとする。ただし、ゴールの方向は、図4b中央にある“^”を移動物体の現在位置と方向に重ね合わせたとき、図中のL, F, Rのどの領域にゴールの位置が含まれるかを判定する。また、障害物は、図4c中央下の移動物体に対し、L, FL, F, FR, Rで示される隣接5区画の各々について障害物の有無を判定する。ただし、そのとき図4bで示したゴールの含まれる領域外の区画にある障害物は無視し、障害物無しとみなす。したがって障害物の状態は、ゴールの方向がL, Rの場合に対して各1通り、Fの場合に対して7通りの組

み合わせとなり、合計9状態となる。移動物体の可能な行動は、図4d中央下の移動物体に対して、左(L), 前方(F), 右(R)の隣接3区画への移動のみとし、移動後は各区画に示される方向を進行方向とする。以上の設定は次のように表現される。

$$D_{state} = D_{goal} \times D_{obstacle} \dots\dots\dots (15)$$

$$D_{action} = D_{move} \dots\dots\dots (16)$$

$$D_{goal} = \{L, F, R\} \dots\dots\dots (17)$$

$$D_{obstacle} = \{L, FL, FL \wedge F, F, FR \wedge F, FR, R, FL \wedge FR, FL \wedge FR \wedge F, C\} \dots\dots\dots (18)$$

$$D_{move} = \{L, F, R\} \dots\dots\dots (19)$$

ここで、(8)式において、3番目の状態は図4cのFLとFの位置に同時に障害物が観測された状態を表わし、最後の状態は観測可能な位置に障害物が存在しない状態を表わす。以上のような問題設定の下で行った計算機実験の結果の一例を次節で示す。

4.2 計算機実験 前節で定義した問題に対し、成功した試行の例として、図5a~cに示されるような3種類のマップを用意し、軌跡の各位置における状

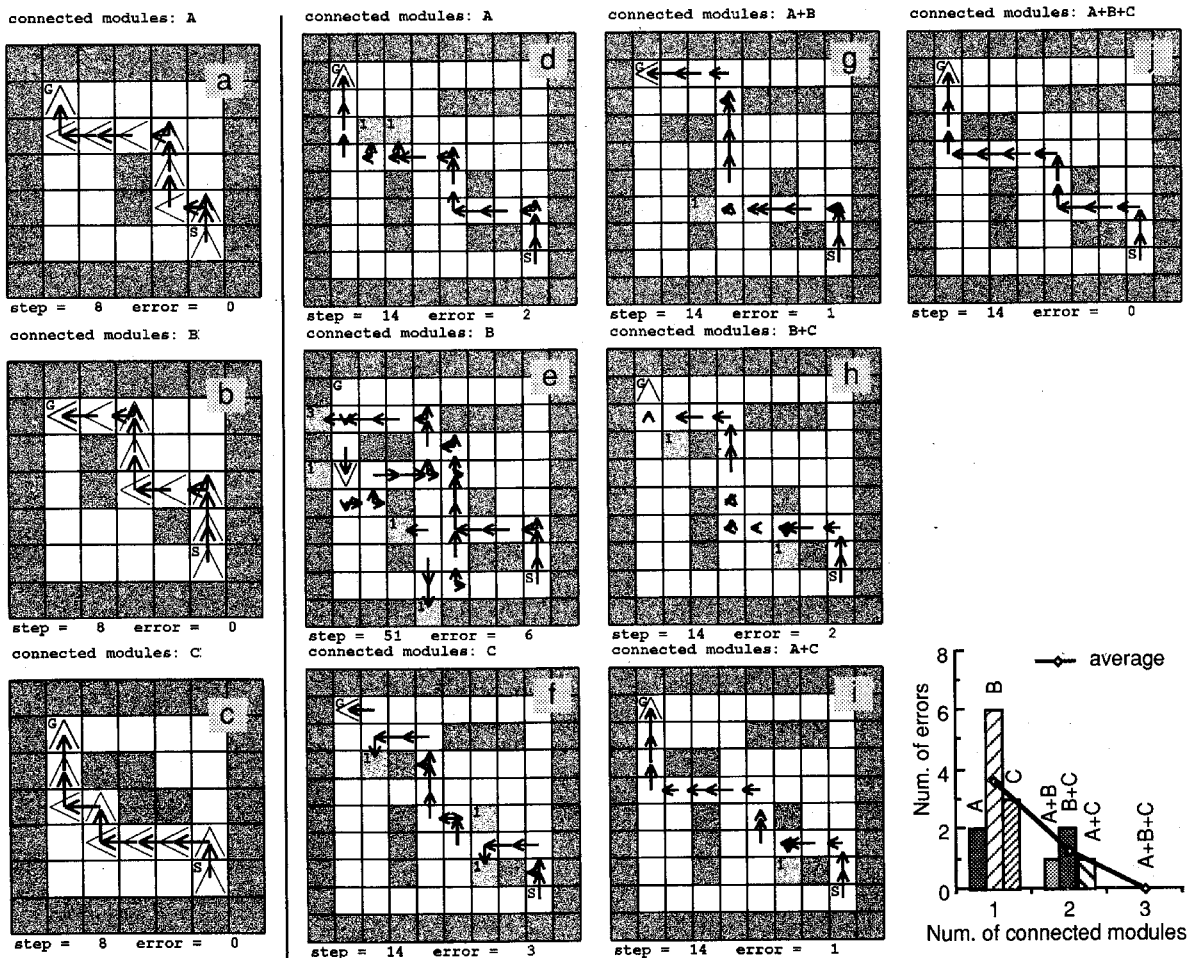


図5 学習マップと未学習マップに対する試行結果

ツマンマシンの推論的な性質を備えた検索能力を既学習知識の統合に適用することの有効性が確認できた。

本モデルは、増加していく知識、あるいは独立に獲得された複数の知識を、実時間で扱うことのできる手法として提案したが、長期にわたって変動し続ける環境においては、冗長あるいは不要な知識の削除や再編成等が必要であり、その実現には再学習が不可欠であると考えられる。したがって、今後の課題としては、実際の応用へ向けて、再学習を含めた総合的な枠組みの構築が考えられる。

参考文献

- (1) Watkins, C., *Learning from Delayed Rewards*, PhD Thesis, (1989), Cambridge University.
- (2) 小森・小谷内, 日本ロボット学会誌, 9-1 (1991), 100.
- (3) 長谷川, 日本機械学会ロボティクス・メカトロニクス講演会講演論文集, B (1992), 113.
- (4) たとえば, Inomata, S. and Nakayama, K., *IEICE Tech. Rep.*, NC90-23 (1990), 77.
- (5) Hinton, G.E. and Sejnowski, T.J., In *Parallel Distributed Processing*, I (1986), 282, MIT Press.
- (6) Rumelhart, D.E. and Zipser, D., In *Parallel Distributed Processing*, I (1986), 151, MIT Press.
- (7) Hinton, G.E., ほか 2 名, In *Parallel Distributed Processing*, I (1986), 77, MIT Press.
- (8) Okuno, T. and Kakazu, Y., *Proc. of World Congress on Neural Networks*, I (1993), 569.
- (9) 安西, 認識と学習, 岩波書店, 1989.