

解説

3. アーキテクチャ



3.6 意味ネットワークマシン†

樋口哲也†† 古谷立美††

1. はじめに

意味ネットワークは、人間の記憶のモデルとして 1968 年に Quillian<sup>1)</sup> によって提案されて以来、人工知能や認知心理学の各分野で広く用いられてきたが、近年、NETL やコネクションマシンといった意味ネットワーク指向の大規模並列アーキテクチャが提案、開発されるに至って、専用ハードウェアによる意味ネットワーク処理が注目を集めるようになってきた(5), (8), (10), (11), (14)。

これは AI の実用化や VLSI・CAD 技術の進展を背景として、意味ネットワークのハードウェア化のメリット、つまり、意味ネットワークの構造のとおり意味ネットワークを多数個の単純な PE 群上にマップすることによって、意味ネットワーク処理に潜む高い並列性を利用できる点が認識されてきたためと考えられる。

一方、最近では神経生理学など、脳における情報処理という視点から、フォンノイマンマシンの枠をはなれて新たな高並列アーキテクチャを提案する動きも活発になりつつある。神経ネットワーク計算機ともいべきものがその一例であり、従来の計算機が不得手とする画像や音声の高度な認識を、人間が行うのと近い形で高速度で実現することをめざす。必ずしも意味ネットワーク処理に直接関わるわけではないが、処理要素間の結線状態そのものが情報処理の中心を占めることなど、アーキテクチャ的にみて意味ネットワークマシンとの関連性がかなり高い。

このため本稿では意味ネットワークマシンを広義に捉えて、神経ネットワークマシンを含めた、意味ネットワークに関するさまざまなハードウェア支援の現状を概観し、それらの動向を考察する。

第 2 章では意味ネットワークの一般的特徴と処理の性質、第 3 章では現在提案、開発中の意味ネットワークマシンを具体的に述べる。本稿では神経生理学などからの意味ネットワークマシン、及び神経ネットワーク計算機へのアプローチを分散型システムと呼ぶこととし、第 4 章では分散型システムの概要について述べる。

2. 意味ネットワーク

2.1 意味ネットワークの特徴

意味ネットワークは、リンクとノードを用いた表現形式であり、ノードは「概念」、リンクは二つの概念間に成り立つ「関係」を表している。意味ネットワーク自体は単にグラフ構造であり、ノードやリンクの意味付けはインタプリタで与えられる。したがって意味ネットワークの表現レベルと意味は自然言語処理や知識ベースなど、応用に依存する。

意味ネットワークのもつ大きな利点は、概念の継承 (inheritance)、つまり上位概念のもつ属性を下位の概念へ継承できることである。これによって、図-1 のような階層的知識の表現を直観的にわかりやすく行える。たとえば「カナリアはえさを食べるか」の問いに対して、「カナリア」のノードには何も直接記されていないが、上位概念の「動物」のノードにはえさを食

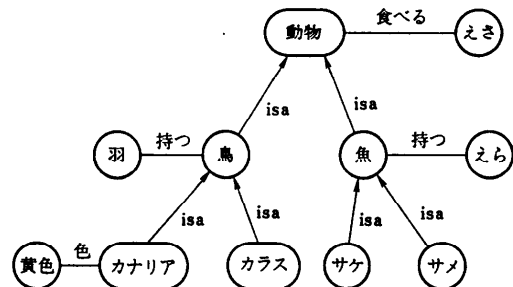


図-1 意味ネットワークの例 (1)

† Semantic Network Machines by Tetsuya HIGUCHI and Tatumi FURUYA (Memory Systems Section, Electronic Computer Division, Electrotechnical Laboratory).

†† 電子技術総合研究所電子計算機部記憶システム研究室

べるとあるため、isa リンクによってこの属性をカナリアに継承し、答えは「正しい」となる。

意味ネットワークはこのような階層的・分類構造的 (taxonomical) な知識を含む宣言的知識の表現にすぐれているため、知識ベース、記憶モデル、自然言語処理など、その応用範囲は広い。しかしながら、手続き的知識の表現に問題があり、このため、意味ネットワークに基づく知識表現言語では、手続き的知識の表現のための枠組をどう取り込むかが共通の課題となっている。意味ネットワークの詳細については文献2)~4)を参照されたい。

2.2 意味ネットワーク処理

意味ネットワークの表す概念や関係は応用ごとにその表現対象、水準が異なるが、共通に見出せる基本的処理として次の三つがあげられる：(1)マーカ伝搬、(2)連想処理、(3)集合演算。これらの処理は、意味ネットワークの各ノードごとに、「マーカビット」と呼ぶ1ビットのフラグを複数個割り当てることによって均一な形で扱うことができる。その様子を図-2を用いて説明する。

図-2は、計算機科学者の集合とノーベル賞受賞者の集合を、簡略化して意味ネットワークで表現したものである。いま「ノーベル賞をもらった計算機科学者は誰か?」という質問が図-2の意味ネットワークに対してなされたとする。この解を得るためには、まず計算機科学者の集合、次にノーベル賞受賞者の集合を求め、そのあと二つの集合の論理積をとればよい。

そこで上述の三つの基本演算から、まず連想処理によって計算機科学者のノードを見つけ、そのノードの、たとえばマーカビット1番をセットする。続いて、マーカ伝搬により、計算機科学者の集合に属する

下位概念の各ノードのマーカビット1番をセットする。つまり、計算機科学者のノードからその下位概念のノードに向けてマーカ伝搬を放射状に開始し、途中にあるノードの各マーカビット1番を次々にセットしていく。

同様にノーベル賞受賞者の集合の要素に対しても、連想処理とマーカ伝搬によって、マーカビットの2番をセットする。

この後、これら二つの集合の論理積を求めるために、マーカビットの1番と2番のANDをとる集合演算を全ノードに発し、その結果、この例ではSimonのノードだけが解となる。

このように意味ネットワークの基本処理はマーカビットを利用して実行できるが、特にマーカ伝搬によって意味ネットワーク特有の並列性が生かされることに注目する必要がある。すなわち、集合は意味ネットワークによって tree 状に表現されることが多いが、その場合ルートノードから放射状にマーカ伝搬が進めば、tree の深さに比例した時間で集合の要素を求めることができる。特に大規模の意味ネットワークの場合には、その効果は著しく、たとえば百万ノードからなる集合が2進木で表されていれば20ステップでマーカ伝搬が終了する<sup>5)</sup>。大規模の意味ネットワークが従来のメモリに配列の形で格納される場合には、スーパーコンピュータを用いたとしても現実的な時間内に解を求めることは難しい。

一方、マーカ伝搬だけでなく、連想処理と集合演算における並列性も高い。すなわち、これらの処理はそれぞれ各ノードに対して同時に行うことができ、意味ネットワーク内の全ノード数分の並列性が存在する。

ただし、これら三つの基本処理における並列性をそ

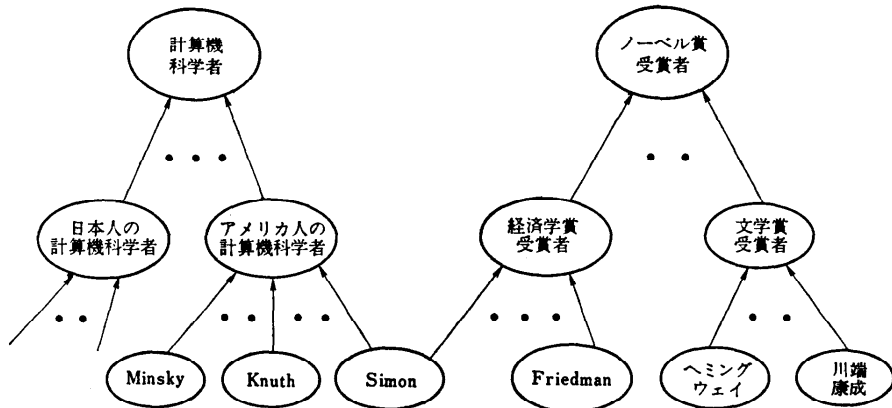


図-2 意味ネットワークの例(2)

のまま生かすには意味ネットワークの各ノードが処理機能をもたねばならない。このため、意味ネットワークのハードウェア化では、各ノードごとに1台の小規模処理装置を割り当てる直接的なマッピングが基本的アプローチの一つとなっている。

### 3. 意味ネットワークマシン

本章ではこれまでに提案、あるいは開発された意味ネットワークマシンの例として、NETL, コネクションマシン, SNAP, CAP, IX をあげる。

#### 3.1 NETL

NETLはカーネギーメロン大学のS. E. Fahlmanによって提案された意味ネットワークマシンである<sup>5),6)</sup>。その特徴は、意味ネットワークのノードやリンクごとにそれぞれ小規模のPEを割り当て、それらのPEを直接意味ネットワークの形のとおり結合することである。

NETLは並列なネットワークメモリを形成し、外部のホスト計算機の制御下でSIMD的に動作する。つまり命令としてNETL内部を伝搬するマーカは一時に一種類であり、目的とする探索や演えきに応じて、ホスト計算機がマーカ列を発生していく。

NETLの構成は、ノードやリンクを表すPE部分と、それらのPE群を結合するネットワーク部分とに分けられる。

PEは図-3に示すように、フラグビットなどのメモリ領域と、コマンドインタプリタからなる。シリアル番号はPEの識別用であり、フラグビットはそのPEの表すノード、またはリンクのタイプを示す。マーカビットはそのPEの現在の状態を示す、一種の短期記憶である。コマンドインタプリタはごく単純なもので、ホスト計算機からのコマンドを、フラグやマーカの状態に応じて受け付ける。

NETLのPEは機能、回路構成とも単純であり、メモリ領域の多いことなど、VLSI化に向く。FahlmanはPEを百万台規模で接続することをゴールとしているが、その根拠の一つはこのPE構成の単純さにある。

一方、PE間を結合するネットワークはHashnetと呼ばれるもので、百万個のPEの同時接続を目的とする。接続の方法としては、ハードウェアとして960×960の接続ネットワークを用意し、それを1,024とおりに時分割することによって、約百万(960×1,024)とおりの接続パスをサポートする。

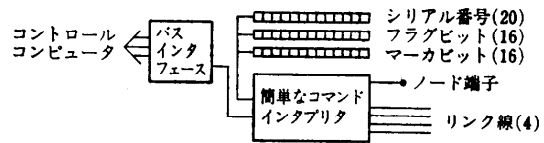


図-3 NETLのPE

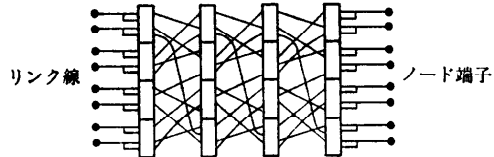


図-4 Hashnetの構成

Hashnetの基本構成を図-4に示す。同図の左側は各PEからの出力で、右側は各PEへの入力である。4つの層に分かれており各層間はランダムに接続されている。各層は1,920個の、15×15のセクタセルを含んでおり、セクタセル内部の4ビットを書き換えることで15とおりの接続パスを作ることができる。この4ビットは、各タイムスライスごとに書き換えられて全く別な接続パスを作る。物理的には一時に960の接続パスを作ればよいのに、実際には2倍に当たる1,920個のセクタセルを含むのは、新しい接続パスを作るときにネットがふさがっている状態を防ぐためである。

新しい接続パスの確立はNETLでは新しい知識の追加を意味する。その方法としては、まず入力端側から、フリーのセクタセルをマークするための信号をブロードキャストし、目標とする出力端に到達させる。次にその出力端から入力へと逆たどりし、途中で通過するセクタセル内のビットをセットしていく。

またネットの時分割を行うには、セクタセルの書き換えのほかに、フラグ、マーカ、シリアル番号も書き換えなければならない。加えて、異なったタイムスライス間での移動を行うことも必要であり、これは、たとえばあるPEにマーカをいま送ったつもりでも、実際にはそのPEへの接続パスが設定される別の時刻になってからそのマーカが送られることがあるためである。そのため各セクタセルには1,024ビットのシフトレジスタが用意され、任意のタイムスライスにマーカを送れるようになっている。

NETLは最初に意味ネットワークのハードウェア化の有用性を示した点で意義をもち、アーキテクチャの面でも、Hashnetや、VLSI化に適したPE構成など、独自性を有する。ただし、ハードウェア化の試算

はあるものの、実装には至っていない。

3.2 コネクションマシン

コネクションマシンは、1981年に Hillis によって提案された意味ネットワークマシンである。ただし、昨年 Thinking Machines 社で開発されたプロトタイプ CM-1 では、1 データごとに PE を割り当てるというこのマシンの特徴を生かして、意味ネットワークだけでなく、画像処理<sup>33)</sup>、知識ベースシステム<sup>34)</sup>、VLSI シミュレーションなどもその応用範囲に含めている。以下では、初期の提案<sup>7)</sup>でなく CM-1<sup>8)</sup> を対象として、その構成、機能を述べる。

CM-1 は、ホスト計算機、及びマイクロコントローラの制御下で SIMD 的に動作する 65,536 個のセルの集まりである (図-5)。32 ビットの加算で 1,000 MIPS の性能をもつ。各セルは 1 ビットの ALU と 4,096 ビットのメモリからなり、CM-1 全体で 32 MB のメモリを有する。

CM-1 は 1.3 m 角の立方体の中に 4 台のラックを含み、各ラックに 1,024 ユニットの収める。1 ユニットの PE 16 個と交信ネットワーク用ルータ 1 個を集積した CMOS・LSI チップ 1 個と、4 個の SRAM チップとからなる。そのユニットを全体で 4,096 個含む。

CM-1 には二つの通信機構がある。第一のものは各チップに一つずつあるルータ同士を結合するルータネットワークである。ルータネットワークの形態は、Boolean  $n$ -cube である。すなわち、4,096 個のルータのおのおのが 12 本のワイアをもち、ルータの番地を 0 から 4,095 とすると、それぞれのワイアが  $2^k (k=0, 1, \dots, 11)$  番地だけ相対的に離れたルータにつながる。

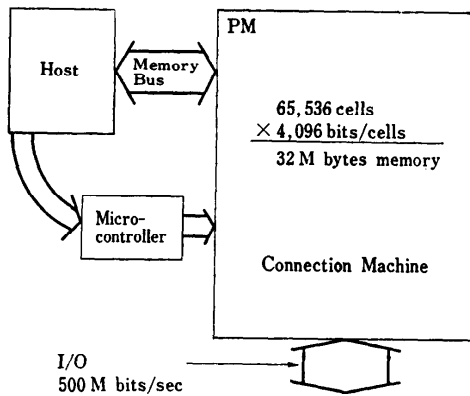


図-5 コネクションマシンの全体構成

このため、CM-1 では約 2 万 4 千本の配線がルータ間になされている。Boolean  $n$ -cube は、PE 間の距離が総 PE 数の対数のオーダーで済む利点をもち、また PE 間の通信パターンの局所性を生かすことができる。第二の通信機構は各 PE をメッシュ結合したものである。LSI チップ内の 16 個の PE は  $4 \times 4$  のメッシュで接続されておりチップ間にまたがるものはチップの NEWS ピンと呼ばれるピンを介して実現される。PE 間のローカルな通信にはこのメッシュが使われる。

PE の内部構成は図-6 のようになっている。1 ビット ALU、汎用及び専用で計 16 ビットの内部フラグ、そして外部メモリを用いて、PE の基本処理は次の 3 サイクルで実行される：(1)オペランドとして外部メモリから 2 ビットと内部フラグ 1 ビットを読み出す、(2)指定された操作に基づき、2 ビットの結果を生成する、(3)結果を外部メモリと内部フラグへ書きこむ。

この基本処理に必要なパラメータ、すなわち外部メモリのアドレスや ALU 操作を指定する真理値表の値は、マイクロコントローラが与える。

CM-1 の命令は次のようにして実行される。まずホスト計算機から高レベルのマクロ命令が発せられる。次にこれをマイクロコントローラが解釈してナノ命令に展開し、最後に CM-1 の各 PE がそのナノ命令を実行する。

たとえばホスト計算機で 8 ビットの加算命令が発せられると、CM-1 の ALU は 1 ビットであるため、マイクロコントローラはこのマクロ命令を 8 つの 1 ビット加算用ナノ命令に展開し、これらを順々に CM-1 全体にブロードキャストする。

CM-1 のプログラミングは Lisp や C などの高級言

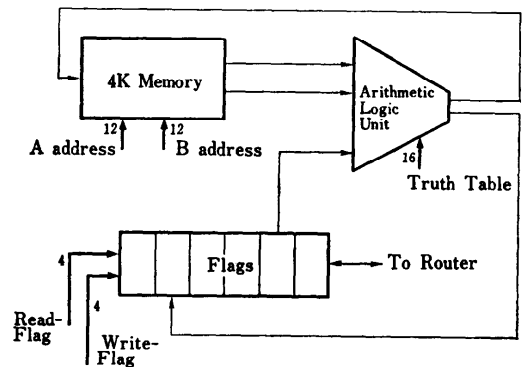


図-6 CM-1 の PE

語をコネクションマシン用に拡張したもので行う。たとえば Lisp の拡張版は Cm Lisp と呼ばれ、そこでは一つのデータを CM-1 の各 PE ごとに写像するために、xector と呼ぶデータ構造をとり入れている。xector はインデックスと値の対を要素とするデータ構造で、図-7 は3要素の xector を示す。xector の要素は別々の PE に分散されるため、たとえば二つの xector の加算の場合、たとえ要素数が何千あったとしても SIMD 的に1回の加算時間ですむ。また xector を用いた並列処理を記述しやすくするために  $\alpha$  notation,  $\beta$  notation と呼ぶ記法も導入されている。

3.3 SNAP

SNAP は南加の Moldovan らによって提案されたマシンである<sup>9)</sup>。基本構成は図-8 に示すように、2次元アレイ状に配置した PE 間をローカルバスとグローバルバスで結合したものである。PE の構成は図-9 であり、連想メモリ (CAM), 処理ユニット (PU), 通信ユニット (CU) の三つの部分からなる。1PE には、SN の1ノードが格納される。CAM はノードを記憶するセルメモリと、ノードに接続されるリンクを記憶するポインタメモリからなる。

PU は PE を制御する部分で、簡単なマイクロプログラム方式のコントローラである。AND, OR, NOT, SET, TEST, RESET, MASK, MATCH, CAR, CDR などの命令をもち、PE 内のフラグ (マーカビット) を使って、後に示す意味ネットワーク操作命令を実行する。CU は通信制御部で、4近傍 PE との通信のほか、グローバルバスを用いた通信を管理する。

意味ネットワークの作成や意味ネットワークとして表現された知識に対する質問は、次に示すような基本命令を用いてなされる。SEARCH は連想命令で、引数で示されるノードのフラグを立てる。F-SEARCH, R-SEARCH は、SEARCH で求まったノードに結合されているノードをさがす命令である。F-PROP, R-PROP はマーカ伝搬命令で、あるノードから、あるリンクによって結合されてい

(SKY→BLUE GRASS→GREEN APPLE→RED)  
図-7 xector の例

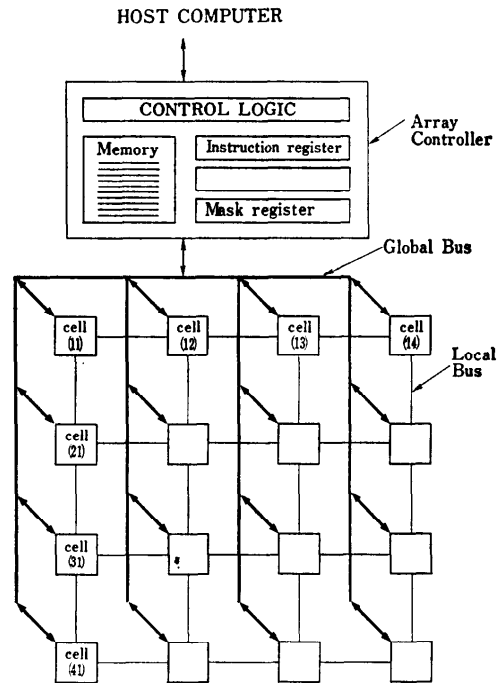


図-8 SNAP の全体構成

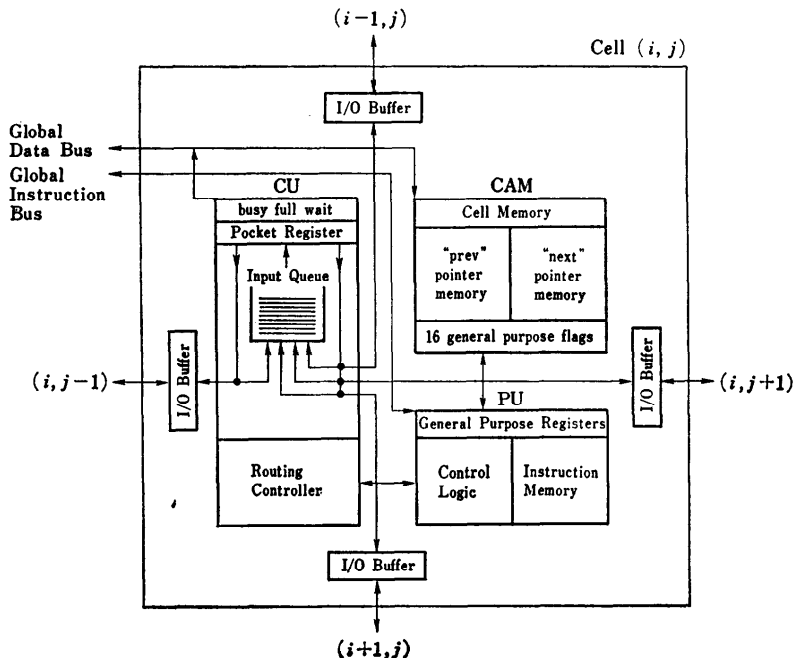


図-9 SNAP の PE



関係を作ろうとすると、父と母が横軸にあるため、面倒なことになる。

### 3.5 IX

電総研では、自然言語処理や知識ベースなど、意味ネットワークとしてモデル化される知識を柔軟に表現し高速に処理する意味記憶システム IX (イックス) を開発している。IX は意味ネットワークに基づく知識表現言語 IXL<sup>12)</sup> と、IXL で記述される知識を高速処理するマシン IXM よりなる<sup>13), 14)</sup>。

図-12 が IXM の構成であり、多数の PE とそれらの間を結合するピラミッド型ネットワークよりなる。PE 内には意味ネットワークを格納する連想メモリがある。このマシンでは、意味ネットワーク処理の基本となる連想と集合演算が PE の連想メモリで行われ、マーカーの並列伝搬はネットワークで実現される。

これまで述べてきたマシンは 1PE に 1ノードを割り当てていたのに対し、IXM では意味ネットワークを複数のノードからなる“部分意味ネットワーク”に分割して、部分意味ネットワークを一つの PE の連想メモリに格納している。部分意味ネットワークを PE の連想メモリに格納することにより、連想と集合演算が完全並列に行われる利点が得られる。また、1PE に複数のノードを割り当てるのは IXL で記述される手続き的知識を IXM 上で並列処理するために高機能 PE が必要となるためでもある。

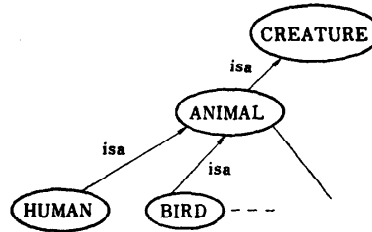
図-13 は、連想メモリへの意味ネットワークの格納の様子を示している。連想メモリの各語は、identifier, destination, リンク、マーカービットからなる。

またこのピラミッド型ネットワークでは、ファンアウトの大きいノードからのマーカー伝搬を高速化するため、各ネットワークプロセッサにも連想メモリを用意してマーカーの並列コピーを実現している。

IXM にはホスト計算機から IXL コマンドが SIMD 的に与えられ、IXL コマンドは PE 内で解釈される。この解釈は、マーカーの到着によって非同期的に駆動されるため、各 PE で独立に行われる。この点に着目すると動作は MIMD と見ることもできる。

## 4. 分散表現型システム

3章で述べたマシンはいずれも基本的には一つの場合を一つの処理装置に割り当てており、個々の情報が局所的に存在する。その意味で 3章のマシンは局所表



identifier field(21 bit)	destination field(21 bit)	link name	marker bit fields(10 bit)
ANIMAL	HUMAN	RISA	-----
ANIMAL	BIRD	RISA	-----
ANIMAL	CREATURE	isa	-----
HUMAN	ANIMAL	isa	-----
CREATURE	ANIMAL	RISA	-----
BIRD	ANIMAL	isa	-----

図-13 連想メモリへの意味ネットワークの格納

現型 (local representation) である。

これに対して、神経科学や生理学など脳の機能に関する研究分野では、特定の記憶事項を多くの処理要素(神経細胞)の上で分散して記憶し、それら処理要素間の結合の活性化(activation)のパターンの違いによって個々の記憶事項を識別する考え方がある<sup>15)~23)</sup>。コネクショニストモデル、神経ネットワーク計算機など、呼び方はさまざまであり、また個々に違いはあるものの、ここでは文献 31) に従い、それらをまとめて分散表現型 (distributed representation) システムと呼ぶことにする。

この分散表現型システムでは、新しい記憶事項の追加が、処理要素間の新しい結合路の設定、あるいは結合の強さの増減といった形でなされる。不完全な入力からでも記憶事項を想起できることなど、人間の脳についてこれまでわかっていることと整合しやすいし、また処理要素の一部が故障しても特定の記憶が失われにくいといった利点があることから、画像や音声の高度のパターン認識をめざして最近この分散表現型システムの提案が盛んになりつつある。すでに VLSI 化、商用化の行われたものもある<sup>24), 29), 35)</sup>。

従来のフォンノイマンマシンにおけるアドレスやアルゴリズムの概念がなく、処理要素間の結合そのものが情報処理の中核となっていることなど、従来とは全く異なった高並列アーキテクチャとして興味深い。本章ではこの分散表現型に属するものについて述べる。

### 4.1 Hinton マシン

ハードウェア化に至らず概念的な提案にとどまっているが、分散表現型システムのイメージを与えるために Hinton の初期の研究<sup>25)</sup>について述べる。ここでは Hinton マシンと略称する。

Hinton マシンは神経細胞に対応するパーセプトロンの機能を各 PE にもたせることを前提にしたもので、意味ネットワーク内の異なるノードやリンクを同一の PE 群の上に写像するマシンである。まずパーセプトロンについて簡単に述べたあと、Hinton マシンの概要を示す。

パーセプトロンは学習することのできるパターン認識装置であり、神経細胞のモデルに基づいている<sup>27)</sup>。つまり神経細胞は多入力1出力の演算素子であり、入力  $X([x_1, x_2, \dots, x_n])$  と重み  $W$  に基づく次式の計算を行って、しきい値  $S$  を越えたときに on になるものとする。

$$x_1 * w_1 + x_2 * w_2 + \dots + x_n * w_n > S$$

このような素子を、たとえば網膜の各細胞として並べ、ある特定の入力パターンのときだけに出力するように重み  $W$  を調整する。この重み付けの調整が学習に対応する。意味ネットワークを Hinton マシンで表現するには、おのおのがパーセプトロン素子に対応する PE を複数個集め、それらの上で複数の概念や関係を表す。同一の PE 群上に異なった概念を覚えさせるためには重みを調整する。

Hinton マシンでは、たとえば図-14 のように PE 群を4つのグループに分ける。意味ネットワークの一つの有向グラフは2項関係を表すので、その2項に対応するものとして、ROLE 1, ROLE 2, 関係に対応するものとして REL と呼ぶ PE のグループを設定する。第4のグループは、命題 (proposition) の意味で PROP と名付ける。

意味ネットワークの記憶には次の形の命令を発する。

(STOREALL' ((JOHN FATHER LEN)  
(JOHN SISTER MARY)  
(KATE FATHER BILL)))

この命令は、図-15 の意味ネットを構成する三つのパターンが入ったときだけそれぞれ入力と同じパターンが再生されるように、三つのパターンに共通なただ1とおりの重み  $W$  を決定するものである。

意味ネットの呼び出しは、PROP 内のデータを利用する。たとえば John の父は誰かという問い合わせ、

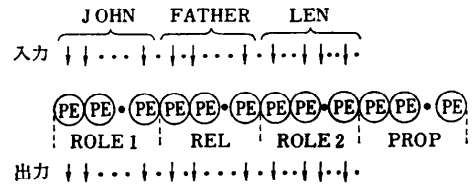


図-14 Hinton マシンの概念図

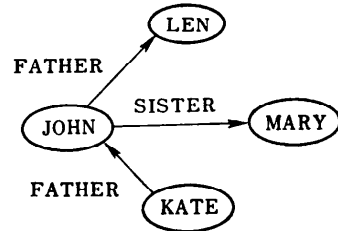


図-15 意味ネットワークの例 (3)

(JOHN FATHER?) は、(JOHN FATHER LEN) を入力したときに決定された PROP の値を用いて答えが返される。PROP の値は一種のハッシュコードともいえる。一方、インヘリタンスは ROLE1 の情報を使って実現される。

Hinton マシンは製作されていないが、このような重み付けによるパターン認識の考え方を具体化したものとして Brunel 大学の WISARD がある<sup>29)</sup>。1981年にプロトタイプが作られた後、産業用の自動監視システムなどにおける特定画像の認識を目的に、CRS 1000 システムとして商用化されている。

### 4.2 Hopfield モデルに基づくシステム

CALTEC, ならびに ATT では、アルゴリズムを実行するのではなく回路の電氣的エネルギーを最小化することによって最適解へ収束するアナログ回路の設計が CALTEC の Hopfield の理論に基づいて行われ、VLSI 化が進んでいる<sup>21), 22), 24)</sup>。

Hopfield の理論では神経ネットワークにおける計算を次のようにとらえる。すなわち、神経ネットワークに入力電圧が加えられるとネットワークが励起状態になり、その結果ネットワークが、問題に応じて定められたエネルギー関数に従って、最も近い極小値に移行し、この極小値における出力電圧が計算結果となる。つまり、問題を表現している初期状態から結果を表す最終状態への軌跡が計算に対応するわけで、フォンノイマンマシンにおけるアルゴリズムの概念はない。

実際の回路では神経細胞をオペアンプに、また神経細胞間の接続を、抵抗を介したオペアンプ出力のフィードバックループに対応させている。CALTEC で



は4ビットのA/Dコンバータを作成しており、アナログ入力電圧に応じて4ビットの出力を出す。入力とオペアンプ出力のフィードバックの交点に抵抗があり、これらの抵抗値によりエネルギー関数が定まる。マトリックス状に配置できることからLSI化に向いており、ATTでは256,000本の相互接続が可能な512個のPE(神経細胞を模擬する)を含むチップを作成している。またCALTECではC. Meadが視覚システムを目的に、ASSOCMEMと呼ぶ、484の相互接続の可能な22PEを含むチップを作成している<sup>35)</sup>。

### 4.3 ボルツマンマシン

ボルツマンマシンはHintonらによって提案された高並列マシンであり<sup>30)-32)</sup>、前節のHopfieldの理論と同様にその動作はエネルギー関数に基づく。ただし各神経細胞の状態が確率的に定まる点が異なる。

その構成は相互に結合された多数( $N$ 個)のユニットからなり、各ユニットは“0”、“1”のいずれかの状態をとる。このような構成で、各ユニット( $U_i$ )の状態を $S_i(1=1\cdots N)$ としたときのハードウェア全体のエネルギー(Hamiltonian)を

$$E = -\frac{1}{2} \sum_{ij} W_{ij} S_i S_j - \sum_i (\eta_i - \theta_i) S_i$$

と定義する。ここで $W_{ij}$ は $U_i$ と $U_j$ の間の相互作用の強さ、 $\eta_i$ は $U_i$ の入力の強さ、 $\theta_i$ は $U_i$ の入力に対するしきい値である。これらの変数の具体的な意味付けはボルツマンマシンを使って解く問題によって定まる。このように $E$ を定義したことにより、他のユニットの状態を保ったまま、 $U_i$ の状態を1から0へ変えたときの全エネルギーの変化量は、回路網の対称性( $W_{ij} = W_{ji}$ )を仮定すると

$$\Delta E_i = \sum_j W_{ij} S_j + \eta_i - \theta_i$$

となることがわかる。そこで、この $\Delta E_i$ によって定まる確率

$$P_i = 1/(1 + \exp(-\Delta E_i/T))$$

( $T$ は動作の激しさを表すパラメータ(温度)で適当に定める)

で、ユニット $i$ の状態を1に設定する。この確率的な状態決定を各ユニットについて繰り返すことにより、各ユニットの状態は全エネルギー $E$ の最小値を与えるものに、熱力学の平衡状態への収束原理に従って確率的に収束することが知られている。

ボルツマンマシンとはこの $E$ を最小にする $S_1 \cdots S_N$ の組み合わせを求めることのできるマシンである。 $E$ を求めるアルゴリズムは各ユニットの値を変えながら全

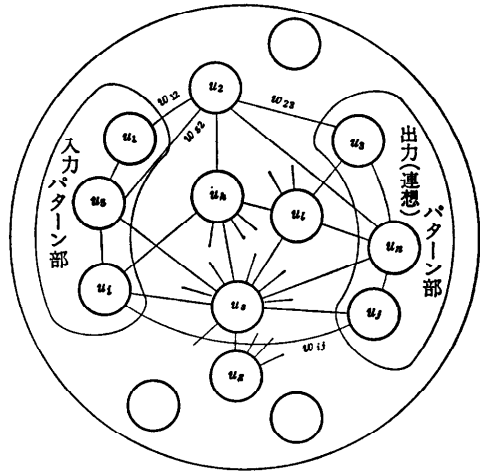


図-16 ボルツマンマシンのイメージ

PEへの影響を調べていく操作を繰り返すもので、並列処理が可能となる。

このようなマシンがあると、 $W_{ij}$ をうまく決めてやることにより連想機能が実現できる。たとえば図-16のようにユニットの一部を入力パターン部、別の部分を出力パターン部とし、それ以外の部分を固定してしまう。こうしておいて、入力パターン部に、ある入力パターン(たとえば図-16で $u_1=0, u_5=1, u_6=0$ )を与えると、これに対応して出力パターン部のユニットの状態たとえば図-16で $s_3, s_4, s_7$ が、 $E$ を最小にするようなものとして求まる。

これは一種の連想機能である。連想機能があると、あるユニットの集合を意味ネットワークのノードに対応させていくことにより、意味ネットワークマシンを実現できる可能性が生まれる。このマシンの実現には多くの問題があると思われるが、アイデアとしておもしろいものである。このほか、ハードウェア化には至っていないが、主にアメリカで分散表現型システムに関する研究が盛んに行われている<sup>15), 16), 23), 24)</sup>。

## 5. おわりに

本稿では意味ネットワークに関わるハードウェア支援を広義に捉え、その動向を局所表現型、及び分散表現型の視点から概観した。意味ネットワークに潜む並列性はハードウェア化しやすい利点があるため、今後大規模知識ベースなど人工知能の実用化の要求が高まるにつれ、高並列アーキテクチャとしての意味ネットワークマシンの重要性はさらに認識されるものと考え

られる。

また分散表現型マシンは神経生理学や認知心理学との学際的な係わり合いをもち、脳により一歩近づいた並列計算機として興味深い。音声や画像の認識など、従来の計算機がやや不得手とするような高度のパターン認識を高速度で実現できる可能性がある。ただしアルゴリズムの記述によって動作が規定できるわけではないので、逆に数値計算などの問題の処理はむずかしいと考えられ、したがって従来の計算機とは補完的な関係になるものと考えられる。

### 参 考 文 献

- 1) Quillian, M. R.: Semantic Memory, In "Semantic Information Processing" (M. Minsky ed.), MIT Press (1968).
- 2) Barr, A. and Feigenbaum, E. A. (田中, 淵監訳): 人工知能ハンドブック, 共立出版, pp. 219-230 (1983).
- 3) 国分明男, 他: 意味ネットワークに基づく知識表現法とそのハードウェア化, 電子技術総合研究所調査報告, No. 212 (1985).
- 4) Findler, N. V. (Ed.): Associative Networks: The Representation and Use of Knowledge by Computers, Academic Press, New York (1979).
- 5) Fahlman, S. E.: NETL—A System for Representing and Using Real-World Knowledge, MIT Press (1979).
- 6) Fahlman, S. E.: Design Sketch for a Million-Elements NETL Machine, Proc. of First Annual National Conf. on AI, pp. 249-252 (Aug. 1980).
- 7) Hillis, G. E.: The Connection Machine, TR-646, MIT AI Lab. (1981).
- 8) Hillis, G. E.: The Connection Machine, MIT Press (1985).
- 9) Moldovan, D. I. and Tung, Y. W.: Semantic Network Array Processor, Technical Report PPP-84-2, University of Southern California (Apr. 1984).
- 10) Morton, S. G., Abreu, E. and Tse, F.: ITT CAP—Toward a Personal Supercomputer, IEEE Micro, pp. 37-49 (Dec. 1985).
- 11) Brooks R. and Lum, R.: Yes, An SIMD Machine Can Be Used for AI, Proc. of Ninth IJCAI, pp. 73-79 (Aug. 1985).
- 12) 半田剣一, 樋口哲也, 古谷立美, 国分明男: 意味記憶システム IX—知識表現言語 IXL による関係の表現, 情報処理学会第 30 回全国大会論文集, pp. 1513-1514 (1985).
- 13) 樋口哲也, 古谷立美, 楠本博之, 半田剣一, 国分明男: 並列連想記憶を用いた意味ネットワークマシン, 電子通信学会技術研究報告, EC 85-55 (1986).
- 14) Higuchi, T. et al.: The IX Supercomputer for knowledge based systems, FJCC, pp. 1041-1049 (1986).
- 15) Anderson, J. A. and Hinton, G. E.: Models of Information Processing in the Brain, in "Parallel Models of Associative Memory" (Hinton & Anderson eds.), Erlbaum, Hillsdale, NJ (1981).
- 16) Feldman, J. A.: コネクションズ, 日経バイト, No. 12, pp. 161-168 (Sep. 1985).
- 17) Hinton, G. E.: Distributed Representation, Technical Report CMU-CS-84-157, Carnegie-Mellon University (Oct. 1984).
- 18) T. Kohonen (中谷訳): 連想記憶, サイエンス社 (1977).
- 19) 中野 肇: アソシアトロン, 昭晃堂 (1979).
- 20) Fukushima, K. et al.: Neocognitron: a Neural Network Model for a Mechanism of Visual Pattern Recognition, IEEE Trans. Syst. Man Cybern. Vol. SMC-13, No. 5, pp. 826-834 (1983).
- 21) Hopfield, J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities, Proc. Nat. Acad. Sci., Vol. 79 (Apr. 1982).
- 22) Hopfield, J.: Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons, Proc. Nat. Acad. Sci., Vol. 81 (May 1984).
- 23) Rumelhart, D. E. and McClelland, J. L.: Parallel Distributed Processing: Exploitations in the Microstructure of Cognition, Bradford Books/MIT Press, Cambridge, MA. (1985).
- 24) Computer That Come Awfully Close to Thinking, Business Week (June 2. 1986).
- 25) Hinton, G. E.: Implementing Semantic Network in Parallel Hardware, in "Parallel Models of Associative Memory" (Hinton & Anderson eds.), Erlbaum, Hillsdale, NJ (1981).
- 26) Minsky, M. and Papert, S. (斉藤訳): パーセプトロン, 東京大学出版会 (1971).
- 27) A 1-BIPS System Takes New Tack in Parallelism, Electronics, May 5 (1986).
- 28) Hammerstrom, D. et al.: The Cognitive Architecture Project, Computer Science & Engineering, The Oregon Graduate School (Feb. 1986).
- 29) Aleksander, I. Thomas, W. and Bowden, P.: WISARD—A Radical Step forward in Image Recognition, pp. 120-124, Sensor Review (July 1984).
- 30) Hinton, G. E. et al.: Boltzmann Machines: Constraint Satisfaction Networks That Learn, Technical Report CMU-CS-84-119, Carnegie-

- Mellon University (May 1984).
- 31) Fahlman, S. E. and Hinton, G. E.: Massively Parallel Architectures for AI: NETL, THIS-TLE, Boltzmann Machines, Proc. AAAI, pp. 109-113 (1983).
- 32) 麻生英樹: パターン情報処理装置としての脳, 数理科学, pp. 56-61 (Nov. 1985).
- 33) M. Drumheller: Connection Machine Stereo-  
matching, pp. 747-753, AAAI (1986).
- 34) G. E. Blelloch: A Massively Concurrent Rule-Based System, pp. 735-741, AAAI (1986).
- 35) M. Sivilotti, et al.: A Novel Associative Memory Implemented using Collective Computation, pp. 329-342, Chapel Hill Conf. on VLSI (1985).

(昭和61年6月5日受付)