ニューラルネットワーク推論システム†

梶原 信樹††

ニューラルネットワークを細粒度高並列の計算モデルと考え，プログラムによりいくつかの簡単な機能と前向き推論システムを実現し，その情報処理能力とプログラム可能性の検討を行った．ニューラルネットワークは，情報処理の多くの分野に適応可能な細粒度高並列の計算モデルで，その学習能力に期待が集められている．しかし大規模なシステムでは，その機能を白紙の状態からすべて学習することは困難である．あらかじめプログラムできる機能はプログラムし，その後学習によって機能をチューニングしたり，新しい機能を獲得する必要がある．本文では，内部状態を持つ時間連続のモデルを提案しその動特性を分析した．またニューラルネットワークの構造をモジュール化して記述できる記述言語を試作した．時系列認識の実験では，離散的な状態を遷移することにより入力系列を認識するネットワークを構成し動作を確認した．前向き推論の実験では1つの命題を1つのノードに対応させ，ノードの活性度で命題の真偽値を表現した．すべてのルールはあらかじめニューラルネットワークの形に展開され，すべてのルールが並列に実行される並列推論システムである．シミュレーションにより，ある程度の情報の欠落や誤りに対する耐性を持つことが確認された．また推論結果の分析を行ったところ本推論システムは人の推論過程の一部の簡単なモデル化になっているとの結論を得た．

1. はじめに

ニューラルネットワークは，推論，パターン認識，制御等，情報処理の多くの分野に適応可能な細粒度高並列の計算モデルで，その学習能力に期待が集められており多くのモデルおよび学習アルゴリズムが提案されている．しかし現在までに提案されている学習アルゴリズムでは，実現できる情報処理の機能は小規模なパターン変換に限られており，大規模な問題や，時間的に変化するパターンの処理機能を獲得することは困難である．画像処理機能（パターン認識機能），推論機能，制御機能等の多数の機能モジュールからなる大規模なシステムをニューラルネットワークで実現する場合，その機能をすべて白紙の状態から学習によって獲得することは効率が悪い．あらかじめプログラムできる機能はプログラムによって獲得し，その後学習によって機能をチューニングしたり，新しい機能を獲得する必要がある．

本研究はその第1ステップとして，ニューラルネットワークの並列協調動作[2),3)]に注目し，その構造，リンクの重みをプログラムすることによりいくつかの簡単な機能と前向き推論システムを実現しニューラルネットワークの情報処理能力，およびプログラム可能性を検討した[1)]．

2章では，ニューラルネットワークモデルについて述べる．ニューラルネットワークを細粒度高並列の計算モデルと考えた場合，関数の機能だけでは不十分である．より高度な情報処理を行うためには，過去の入力の履歴を記憶しそれに従って動作する順序回路のような機能が必要である．過去の履歴を保持するために各ノードが活性度と呼ぶ内部状態を持ち，他のノードからの影響を受けながら，時間とともに活性度を変化させるニューラルネットワークモデルを提案した．必要な機能をプログラムで実現するためには，ノードの特性を理解しておく必要がある．そのためにノードの動特性も分析した．

3章では，LISP マシン上に試作したニューラルネットワークプログラミング環境，特にネットワーク記述言語について述べる．ニューラルネットワークのプログラムは，ニューラルネットワークの構造（リンクの重み）として格納される．試作した記述言語はニューラルネットワークの構造をモジュール化して記述できる．記述されたプログラムはコンパイラによってニューラルネットワークの構造に展開され，シミュレーションにより動作を確認される．

4章では，ネットワークで実現した機能（時系列認識，前向き推論等）の実験結果について述べる．時系列認識の実験では，離散的な状態を遷移することにより入力系列を認識するネットワークを構成し動作を確認した．知識処理への応用として簡単なルールベースシステムを実現した．ニューラルネットワークモデルで実現したルールベースシステムがいくつか報告されている．Blelloch[6)] は，ルールをあらかじめ静的なネットワーク構造に展開し，複数の推論を並列に実行する並列推論システムを実現している．後向き推論の

† Neural Network Reasoning System by NOBUKI KAJIHARA (C&C System Research Laboratories, NEC Corporation).

†† 日本電気(株) C&C システム研究所

機能も実現しているが，入力演算や出力関数が複数種類必要で，複数のタイプのノードを使用している．Touretzky[7),8)] は，分散表現でプロダクションシステムの照合-実行のサイクルを実現している．変数の機能も実現している．しかし照合-実行のサイクルはあくまでも逐次的で複数の推論が並列には進まない．また，照合-実行のサイクルを制御するためにニューラルネットワーク以外の機構が必要である．Samad[9)] も照合-実行のサイクルを実現している．入力データは逐次的に与えられ出力も逐次的に出てくる．複数の推論が同時に進むような並列性はない．層構造のニューラルネットワークであるが，各層を構成するノードに異なった機能（例えば，不応期，レジスタ機能）が必要である．

本研究では，Blelloch と同様にルールをあらかじめ静的なネットワーク構造に展開し，複数の推論を並列に実行する並列推論システム（前向き推論）を同一の動特性を持つノードを使って実現した．1つの命題を1つのノードに対応させ，ノードの活性度で命題の真偽値を表現した．ノードの活性度は連続値で持つので，これによって確信度付の推論が実現できる．シミュレーションにより，ある程度の情報の欠落や誤りに対する耐性を持つことが確認された．また推論結果に対して簡単な分析を行ったところ本推論システムは人の推論過程の一部の簡単なモデル化になっているとの結論を得た．

2. ニューラルネットワークモデル

制御システム等の応用では，静的なパターンだけでなく時間的に変化する時系列パターンも扱わなければならない．時系列パターンを扱うには，過去の入力の履歴を何らかの形で保持する必要がある．バックプロパゲーション型のモデルでは出力は現在の入力の関数で，モデルそれ自体に過去の履歴を保持する能力はない．したがって時系列パターンをあらかじめ空間パターンに変換したり，1時刻前の出力パターンをラッチしそれを入力にフィードバックする[12),13)] などの工夫が必要である．ホップフィールド型のモデルでは，各ノードが内部状態を持ち，過去の入力の履歴を保持している．

バックプロパゲーション型モデルとホップフィールド型のモデルの違いは，論理回路における組合せ回路と順序回路の違いに比喩できる．

本研究では理論的な解析は困難であるが，より計算能力が高いということでホップフィールド型の内部状態を持つモデルを採用する．

ネットワークはノードとノード間で情報を転送するための重み付の有向リンクから構成される．ネットワークの構造は特に仮定しない．自分自身へのフィードバックリンクも含めて任意の構造が可能とする．

2.1 ノードの動作方程式

各ノードは活性度という内部状態を持ち，自分自身も含めて他のノードの影響を受けて次の微分方程式に従って変化する（図 1）．

$$\tau \mathrm{d}a_x(t)/\mathrm{d}t = v_x(t) - a_x(t), \qquad (1)$$

$$v_x(t) = \sum_y Wxy \times o_y(t),$$

$$o_x(t) = \mu(a_x(t)),$$

t：時間，

$a_x(t)$：ノード x の活性度，

$o_x(t)$：ノード x の出力，

Wxy：ノード y から x へのリンクの重み（正負の実数），

$v_x(t)$：ノード x への投票，

μ：出力関数，

τ：時定数．

$a_x(t)$, $o_x(t)$, $v_x(t)$ は時間 t の関数であるが以後単に a_x, o_x, v_x と書く．v_x は自分自身も含めた他のノードからノード x への影響でこれをノード x への投票と呼ぶ（自分自身からの投票を含めないときはその都度ことわる）．活性度 a_x は投票 v_x に追従する1次系である．τ は時定数で以後 τ を時間の単位とする．出力関数 μ は(2)式，図 2 に示すような非線形飽和特性を持つ増加関数である．以後この(1)式を動作方程式と呼ぶ．

$$\mu(x) = \begin{cases} 0 & ; \ x<0 \\ 2x^2 & ; \ 0<x<0.5 \\ 1-2(1-x)^2 & ; \ 0.5<x<1 \\ 1 & ; \ 1<x. \end{cases} \qquad (2)$$

ノード x の活性度 a_x の意味は情報の表現法によりプログラマが任意に決める．例えば，図1においてノード x をある仮説に対応させると，活性度 a_x はそ

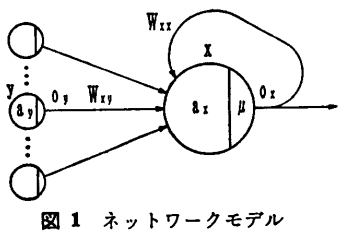


図 1 ネットワークモデル
Fig. 1 Network model.

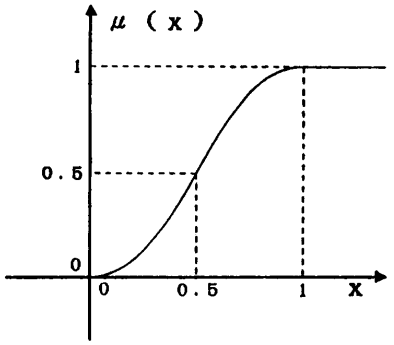


図 2 出力関数 μ
Fig. 2 Output function μ.

の仮説の確信度と解釈することもできる．ノードyは仮説 x を背定 ($Wxy>0$) または，否定 ($Wxy<0$) する証拠で，その確信度 a_y に応じて仮説の確信度に影響を与える．投票 v_x はその時点で仮説 x を検証する証拠の強さである．出力関数μの非線形性により，証拠の確信度 a_y が負のときには他の仮説に影響を与えない．確信度が小さいとき ($a_y<0.5$) には他の仮説に与える影響力は小さく，確信度が大きくなると ($a_y>0.5$) 影響力は大きくなる．一般に，ある仮説は多数の証拠の存在によって検証されるので，1つの証拠の確信度が非常に強いとしても他の証拠の確信度が小さい場合は仮説の確信度は大きくならない．すなわち，証拠の確信度が十分大きくなると ($a_y>1$)，その証拠の影響力は飽和する．動作方程式(1)の右辺第2項は減衰項である，これにより，複数の証拠が同時にそろったときのみ仮説は検証される．

動作方程式(1)は以上のような考察により決定した．しかし，これは普遍性の高い解釈の1つであって，情報をどう表現するかによって，これ以外の意味付をすることが可能である．

2.2 ノードの基本特性

ネットワークをプログラムする場合にはノードの特性を理解する必要がある．動作方程式(1)より導かれるノードの特性を分析する．

図3は自分自身以外のノードからの投票が0の場合のノード x の活性度 a_x と $\tau\mathrm{d}a_x/\mathrm{d}t$ の関係を示したものである．自分自身へのリンクの重み Wxx が $Wt=(2+\sqrt{2})/4$ より大きなノードの曲線は a_x 軸との交点を3つ持つ．このうち原点と右端の交点は安定な点でxの活性度は2つの安定な値をとりうることを示している．例えば $Wxx=1$ であれば x の活性度は0と1で安定である．すなわち $Wxx>Wt$ のノードは双安定のメモリとしての機能を持つ．図3の曲線は他の

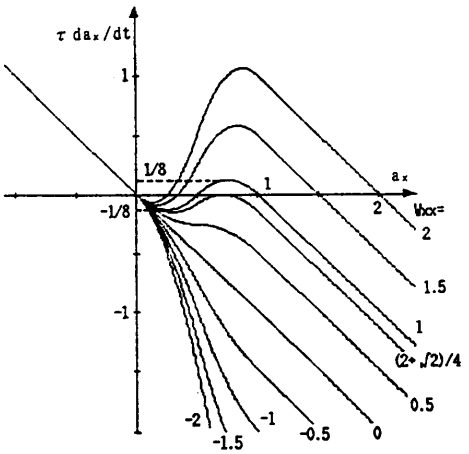


図 3 $\tau\mathrm{d}a_x/\mathrm{d}t=Wxx\mu(a_x)-a_x$
Fig. 3 $\tau\mathrm{d}a_x/\mathrm{d}t=Wxx\mu(a_x)-a_x$.

ノードからの投票 v_x の値によって上下にシフトする．絶対値の大きな投票が十分に長い時間与えられると双安定なノードは，一方の状態から他の状態になる．例えば $Wxx=1$ のノードの活性度が $a_x=0$ の状態のとき投票 $v_x=1$ が約 0.546τ ($v_x=0.2$ なら約 4.708τ) 以上与えられると活性度 a_x は0.5より大きくなり，それ以後投票 v_x が0となっても活性度 a_x は1になる．$v_x<1/8$ であればいくら長い時間与えられても状態を変えない．

ノードのインパルス応答とステップ応答を**図 4, 5**

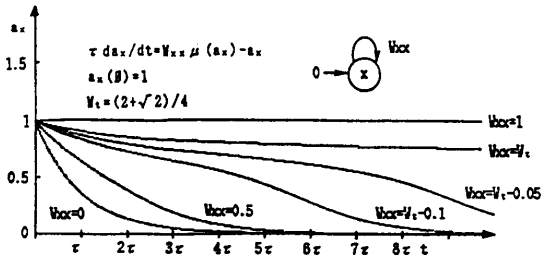


図 4 インパルス応答
Fig. 4 Impulse response.

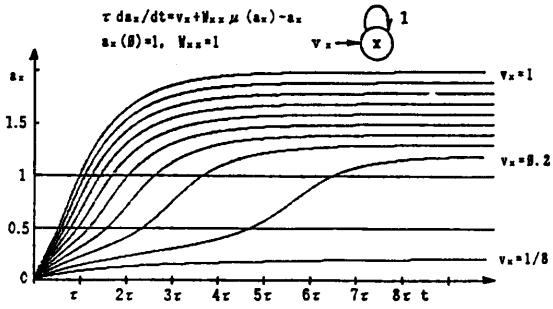


図 5 ステップ応答
Fig. 5 Step response.

に示す．図4のインパルス応答では，$Wxx<Wt$ の場合は，$t\to\infty$ で $a_x\to 0$ となるが，$Wxx>Wt$ の場合は0以外の正の値に収束する．図5のステップ応答は $Wxx=1$ のノードが自分自身以外の他のノードからの投票 v_x をステップ状に与えられたときの応答である．活性度 a_x が0.5を越えれば以後は $v_x=0$ となっても活性度は減衰しない．

3. ネットワークのプログラム

ニューラルネットワークをプログラムするためのプログラミング環境を LISP マシン上に開発した．この環境は，ニューラルネットワークの動作を確認するためのシミュレータ／モニタ，およびニューラルネットワークをプログラムするための記述言語／コンパイラから構成される．

プログラムは，記述言語を使ってネットワークの構造とリンクの重みを記述することにより行う．この記述はコンパイラによってネットワーク構造に変換され，シミュレータで動作を確認することができる．

記述言語はひとまとまりの機能を実現する部分ネットワークをテンプレートとして定義し，さらにそのテンプレートを使ってより複雑なネットワークを定義する機能を持つ．

3.1 シミュレータ

ネットワークを構成するすべてのノードに関して動作方程式(1)を解くことによってニューラルネットワークの動作をシミュレーションする．シミュレータは，動作方程式(1)を次のオイラー法により Δt ごとに逐次的に解く．

$$\Delta a_x(t)=\Delta t\times[v_x(t)-a_x(t)]/\tau.$$
$$a_x(t+\Delta t)=a_x(t)+\Delta a_x(t). \qquad (3)$$

シミュレータはネットワークのノードの活性度，出力の時間変化をモニタする機能を持つ．

3.2 ニューラルネットワーク記述言語

記述言語はネットワークをプログラムするときによく現れる部分ネットワークのパターンをテンプレートとしていくつか用意しており，それらを使ってより複雑なネットワークを記述することができる．

最も基本的なテンプレートとして ifc (IF Coincident then activate) を用意している．ifc のシンタックスと意味を**図6**に示す．$y_1, y_2, \cdots, y_n$，x はノード名，$Wxy_1, Wxy_2, \cdots, Wxy_n$ は数である．ノード名は，各ノードに固有の名前で，この名前によってネットワーク内のノードを指定する．基本的には，ifc があればどんなネットワークでも記述できる．しかし，これだけでは不便である．ひとまとまりの機能を実現する部分ネットワークをテンプレートとして定義し，さらにそのテンプレートを使ってより複雑なネットワークを定義する機能が必要である．

2つのノード x, y の活性度がほぼ同じであれば，活性化するノード $x=y$ は，**図7**のように構成できる．→は正の重みを持つリンク，↦は負の重みを持つリンクである．↦は活性度が常に1のノード *true* からのリンクである．図7のネットワークをテンプレートとして定義するためには次のように記述する．

```
(def-relation equally-active (x y x=y)
  (declare (local x>y y>x))
  (ifc ((x 1) (y -1))
       (x>y))
  (ifc ((x -1) (y 1))
       (y>x))
  (ifc ((*true* 1) (x>y -1) (y>x -1))
       (x=y)))                          (4)
```

equally-active という名前でこのテンプレートが登録される．x, y, $x=y$ はテンプレートの仮引数ノードである．$x>y$, $y>x$ はローカルなノードとして宣言され，プログラムの他の部分からは，このノードを参照することはできない．仮引数とローカルなノード以外，(4)式では *true*, はグローバルなノードである．グローバルなノードはその名前によってどこからでも参照できる．*true* は，活性度が常に1のノードである．(4)式の (declare...) 以降の3つの (ifc...) で部分ネットワークを記述している．この3つの式の順序は equally-active の機能には，まったく影

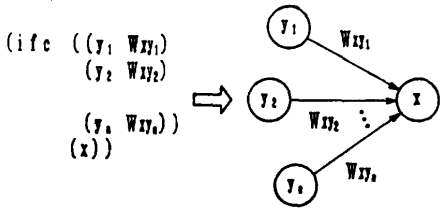


図6 ifc (IF Coincident then activate)
Fig. 6 ifc (IF Coincident then activate).

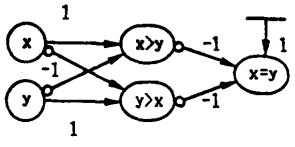


図7 equally-active
Fig. 7 Equally-active.

響を与えない．equally-active は，次のように使う．

(equally-active left-eye right-eye the-front) (5)

これで図7の x, y, $x=y$ をそれぞれ left-eye, right-eye, the-front というノードに，$x>y$, $y>x$ を他から参照できない名前を持つ2つのノードに変えたネットワークを記述したことになる．

def-relation で定義したテンプレートは，より複雑なネットワークを定義するときの部品として使用できる．

現在のところ引数ノードの数が不定個のテンプレートは，def-relation では定義できないので，記述言語の基本的なテンプレートとして LISP で定義している．

ネットワークの記述をしていると，ノード y からノード x へのリンクが重複して記述されることがある．そのときには，各々の記述の平均を Wxy の値とする．すなわち，ある記述1の中でノード y からノード x へのリンクが $Wxy=W1$ と定義され，記述2の中で $Wxy=W2$ と定義されると $Wxy=(W1+W2)/2$ となる．これによって，ネットワーク全体としては，記述1と記述2で定義される動作を平均したような動作をすることを期待した．この方法は，ネットワークのデバッグをやりにくくする原因となるかもしれないが，今までに行った小規模な実験では，問題は生じてない．

4. 実　　験

この章では，システムが用意する基本的なテンプレートについて述べる．また応用例として前向き推論の機能をネットワークで実現する方法と，その実験結果について述べる．注意していただきたいのは，以下で説明するテンプレートは，ニューラルネットワークモデルでどんな機能が実現できそうかを検討するために実験的に構成したもので，これが最良というものではなく，もっと良いものがあるかもしれないということである．実用的なプログラミングシステムとするには，今後さらにテンプレートの拡充が必要である．いくつかのネットワークを図示するが，リンクの重みを特に書いていない場合は重みの絶対値が1のリンクである．また，ノード x の活性度 a_x を単に x と書く．

4.1 wta (Winner Takes All) ネットワーク

いくつかの競合する仮説は，図8のようなネットワークで構成できる．各ノード a, b, c が競合する仮説を表す．v_a, v_b, v_c は，それぞれノード a, b, c への投票である．各ノードは自分自身へは重み1のリンクを持ち，他のノードへは重み -1 のリンクを持つ．a, b, c はそれぞれ v_a, v_b, v_c の投票を受けて活性度を変化させるが，最終的には最大の投票を集めたノードだけが活性化され他は抑制される．ある1つのノード，例えば a が活性化した後に $v_a=v_b=v_c=0$ となってもa は活性を保持する．すなわち wta ネットワークは，多安定のメモリの機能を持っているといえる．v_a, v_b, v_c がほぼ同じ値であれば，どのノードも活性化できない．

図8で，ノード a が活性化している状態 ($a_a=1$, $a_b=a_c=-1$) から b が活性化している状態 ($a_b=1$, $a_a=a_c=-1$) に変化させるためには $v_a<-1/8$, $v_b>1/8$, $v_c=0$ としてしばらくこの投票を保持すれば良い．図3の $\tau \mathrm{d}a_x/\mathrm{d}t$ の特性を見ると $Wxx=1$ のノードは，他からの投票が 1/8 より大きければ活性化し $-1/8$ より小さければ活性を保持できないことがわかる．$v_a<-1/8$ にすることによりa は活性を保持できなくなる．そしてa からの抑制がなくなり $v_b>1/8$ が与えられることによりb は活性化し他を抑制する．上の動作は状態を変化させるための十分条件であって必要条件ではないことに注意されたい．

図9は，wta ネットワークの実験に使ったネットワークである．ノード carp-win, tigers-win, giants-win, dragons-win, whales-win, swallows-win で競合する仮説を表し，wta ネットワークを構成している．ノード carp, tigers, giants, dragons, whales,

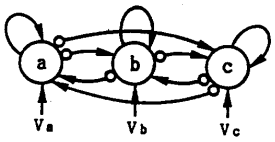


図 8 wta ネットワーク
Fig. 8 wta (Winner Takes All) network.

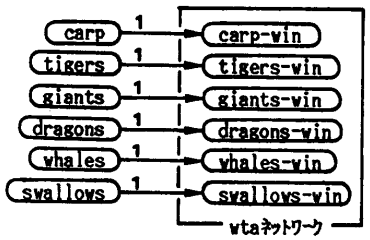
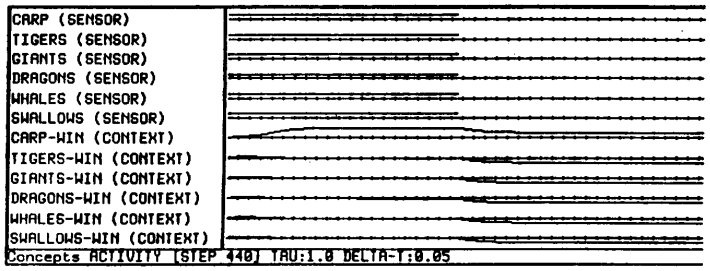
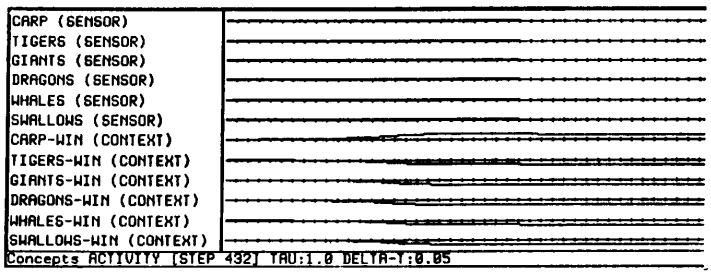


図 9 セントラルリーグ
Fig. 9 Central league.

swallows は入力ノード（センサノード）で，外部から指定した値に活性度を固定する．実験結果を**図 10** に示す．図 10 はシミュレータのモニタ画面の一部である．グラフの縦軸が活性度（軸と軸の間隔が 4），横軸が時間で 1 目盛りは 0.5τ である．図 10 (a) は，センサ carp の活性度を 1.0 それ以外を 0.8 にして実行した場合である．スタートして約 4τ で carp-win が活性化し他のノードは抑制される．その後センサノードの活性度を 0 にしても，carp-win の活性度は保持される．図 10 (b) は，carp の活性度を 0.3 それ以外を 0.2 にしたときの結果である．やはり carp-win が活性化するが活性化するまでの時間は約 10τ である．



(a) carp=1.0, それ以外は 0.8



(b) carp=0.3, それ以外は 0.2

図 10 セントラルリーグの実験結果
Fig. 10 Results of "Central league."

4.2 ifs (IF Sequent then activate)

一連のノード $a_1, a_2, \cdots, a_n$ が順に活性化したときに活性化されるようなノード x を構成したいことがある．例えば，h, e, l, l, o というノードがこの順に活性化したときに hello というノードが活性化するようにしたい．このような機能を実現するために ifs (IF Sequent then activate) というテンプレートを用意しており

(ifs h e l l o hello) (6)

で上の機能を定義できる．

内部状態を持つニューラルネットワークで時系列を認識する方法はいくつか提案されている．外部からの刺激に駆動されて相互結合した 1 次元のノード配列上を活性化パターンが伝わることにより認識する方法[4]や，離散的な状態を遷移することにより認識する方法[5]がある．ここでは，後者の方法で ifs を構成した．その構成方法を以下に述べる．ifs を構成するために 2 種の部品 ifs-aux 1, ifs-aux 2 を用意する (**図 11**)．ifs-aux 1 と ifs-aux 2 は y から x への抑制性のリンクがあるかないかが異なるだけである．これはトリガ t によってある状態 x から他の状態 y へ遷移するネットワークである．x が活性化していないと $t1$ が $t2$ を抑制し t が活性化しても y は活性化しない．x が活性化すると $t2$ への抑制がなくなる．そこでトリガ t が活性化すると，$t2$ を介して y が活性化する．ifs-aux 2 の場合は y によって x は抑制される．このとき $t2$ も抑制されるが y はすでに活性を保持できるまで活性化している．この 2 種類の部品を使うと (ifs a b c d abcd) は，**図 12** (a) のように構成できる．

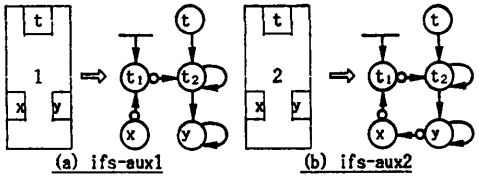


図 11 ifs の構成部品
Fig. 11 Components of ifs (IF Sequent then activate).

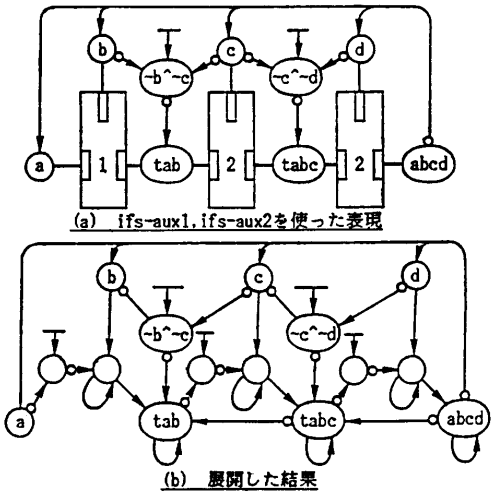


図 12 (ifs a b c d abcd)
Fig. 12 (ifs a b c d abcd).

図 12 (b) は (a) を展開した結果である．tab は a, b が順に活性化したことを記憶するローカルなノードで

ある．実験結果を図 13 に示す．横軸の目盛りは 5τ である．sh, se, sl, so は，それぞれキーボードから h, e, l, o のキーが押されたときに 2.5τ だけ活性度が1になるセンサノードである．hello は h, e, l, l, o とキー入力があったときに活性化する．

lc-ifs-1, 5, 9 は，それぞれ he, hel, hell の系列が活性化したときに活性化されるローカルなノードである．図 13 の 1, 2, 3 は，それぞれ hell, helo, hello と入力した場合で，不完全な1と2では hello は活性化しない．正しく入力が行われた3の場合だけ活性化する．

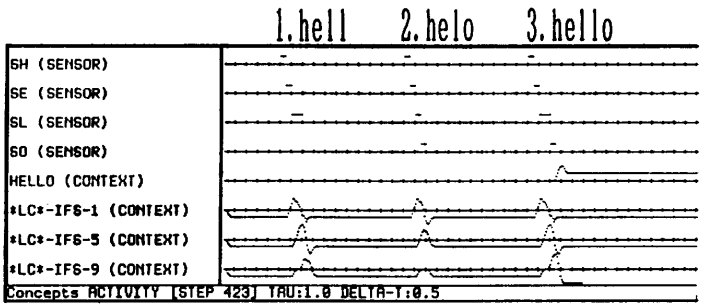


図 13 (ifs sh se sl sl so hello) の実験結果
Fig. 13 Results of (ifs sh se sl sl so hello).

ここで実現した ifs は正しい系列が入力されたときのみ出力ノードが活性化する．しかしより柔軟な認識のためには，系列の一部が欠けたり，余分なものが挿入された不完全な系列に対してもその程度に応じて出力が活性化するような特性を実現することが今後の課題である．

4.3 前向き推論

簡単な前向き推論の機能をネットワークで構成し実験した．例題は，Winston[10] の動物当てシステムのプロダクションルールを使用した．このネットワークは与えられた動物の特徴からその特徴を持つ動物名を推論する．図 14 に推論に必要なネットワーク zoo の定義を示す．(if...then...) は1つのプロダクションルールに対応し図 15 のようにネットワークに変換される．is-tiger* (仮説) は，自分自身へ重み1のリンクを持つので他からの投票が 1/8 以上になれば活性化する．is-tiger* の証拠となる4つのノードの活性度がすべて1以上になれば is-tiger* への投票は $W=0.2(>1/8)$ になり is-tiger* は活性化する．これで AND-OR 推論木の AND ノードを構成したことになる．実際には仮説をサポートするすべての証拠の 62.5% (0.125/0.2) 以上が集まればその程度に応じて仮説は活性化する．すなわち，本システムは，厳密な推論ではなく，確信度付の推論を実現している．

zoo をネットワークに展開すると図 16 のようになる．図 16 のネットワークは，図 14 をプロダクションルールとして実行したときに作られる推論木と，ほとんど同型のネットワークである．このことから zoo の定義をネットワークに展開した時点で既に必要な推論は半分済んでいるといえる．この展開によって 40 個

```
defrelation zoo ()
  ;;; 哺乳類
  (if (has-hair) then (is-mammal))
  (if (gives-milk) then (is-mammal))
  ;;; 鳥類
  (if (has-feathers) then (is-bird))
  (if (flies lays-eggs) then (is-bird))
  ;;; 肉食獣
  (if (eats-meat) then (is-carnivore))
  (if (has-pointed-teeth has-claws
       has-forward-eyes)
   then (is-carnivore))
  ;;; 有蹄類
  (if (is-mammal has-hoofs)
   then (is-ungulate))
  (if (is-mammal chews-cud)
   then (is-ungulate even-toed))
  ;;; 種の決定
  (if (is-mammal is-carnivore
       has-tawny-color has-dark-spots)
   then (is-cheetah*))
  (if (is-mammal is-carnivore
       has-tawny-color has-black-stripes)
   then (is-tiger*))
  (if (is-ungulate has-long-legs has-a-long-neck
       has-tawny-color has-dark-spots)
   then (is-giraffe*))
  (if (is-ungulate
       has-white-color has-black-stripes)
   then (is-zebra*))
  (if (is-bird does-not-fly has-long-legs
       has-a-long-neck is-black-and-white)
   then (is-ostrich*))
  (if (is-bird
       does-not-fly swims is-black-and-white)
   then (is-penguin*))
  (if (is-bird flies-well)
   then (is-albatross*))
  ;;; Winner Takes All ネットワーク
  (wta is-tiger* is-cheetah*
       is-giraffe* is-zebra*
       is-ostrich* is-penguin* is-albatross*)
  ;;; モータの駆動
  (ifc (is-tiger*) (is-tiger))
  (ifc (is-cheetah*) (is-cheetah))
  (ifc (is-giraffe*) (is-giraffe))
  (ifc (is-zebra*) (is-zebra))
  (ifc (is-ostrich*) (is-ostrich))
  (ifc (is-penguin*) (is-penguin))
  (ifc (is-albatross*) (is-albatross)))
```

図 14 動物当てニューラルネットワークの記述
Fig. 14 Description of the network which reasons about animal species.

```
(if
 (is-mammal
  is-carnivore
  has-tawny-color
  has-black-stripes)
then
 (is-tiger*))
```

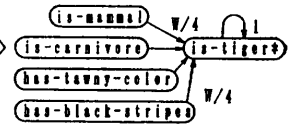


図 15 プロダクションルールのネットワーク表現 ($W=0.2$)
Fig. 15 Network representation of a production rule ($W=0.2$).

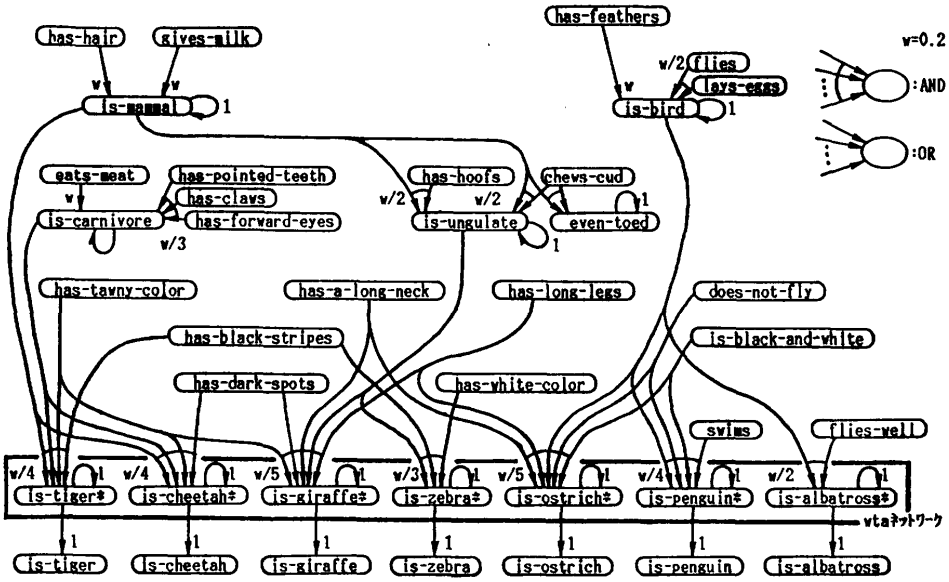


図 16 動物当てのネットワーク ($W=0.2$)
Fig. 16 The network which reasons about animal species ($W=0.2$).

のノードと102本のリンクが作られた．図16のノードのうち他からのリンクを持たないものがセンサノードで動物の特徴（21個ある）に対応している．21個の特徴のうち指定したいくつかの特徴に対応するセンサノードの活性度を1に固定する．動物名に対応して7つの出力ノード（モータノード）is-tiger, is-cheetah, ..., is-albatross がある．7つのモータノードのうち活性度が0.9以上になったノードに対応する動物名が出力される．残りの 12 個のノードはプロダクションルールにより作られる中間結果に相当する．12 個のノードのうち 7 個（is-tiger*, is-cheetah*, ⋯, is-albatross*）は7つの動物名に対応し，これが重み1でモータノードに影響を与える．そして，この7つのノードは，wta ネットワークを構成しており，同時に唯1つのノードしか活性度が大きくなれない．実験結果を**図 17, 18, 表 1** に示す．図17は動物の特徴としてトラ（Tiger）の特徴を与えた場合（表1の場合1）である．横軸の目盛りは τ である．まず is-mammal（哺乳類）と is-carnivore（肉食獣）がほぼ同時に活性化し，そして最後に is-tiger* が活性化している．is-tiger* によって is-tiger が活性化される．is-tiger の活性度は 11.4τ 後に0.9を越える．is-tiger* の活性化とともに他の動物が抑制されていることがわかる．is-mammal と is-carnivore が同時に活性化することは注目に値する．単純な例ではあるが，2つの推論が同時に並列に行われたことを示す．図 14 の zoo の中のすべてのルールは常に並列に実行されていると

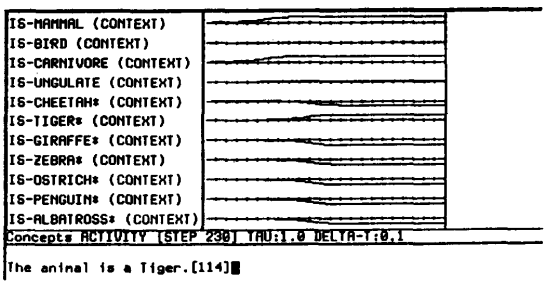


図 17 トラの特徴を与えた結果
Fig. 17 Results of reasoning from features of tiger.

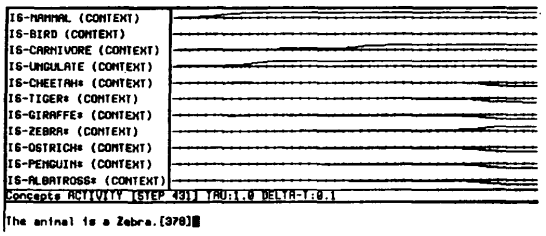


図 18 シマウマ-1，トラ-2 の特徴を与えた結果
Fig. 18 Results of reasoning from features of both zebra and tiger.

もいえる．図 18 はトラの特徴から2つ取り除いたものとシマウマ（Zebra）の特徴から1つ取り除いたものを両方与えた場合（表1の場合 18）である．最終的にはシマウマが活性化するが is-zebra の活性度が 0.9 を越えるのは 37.8τ 後である．

表1はいろいろな場合の結果をまとめたものである．活性化するまでの時間 T は，動物の特徴が与えら

表 1 動物当ての実験結果
Table 1 Results of reasonings.

T:Tiger C:Cheetah G:Giraffe Z:Zebra O:Ostrich P:Penguin A:Albatross

場合	1 T	2 C	3 G	4 Z	5 O	6 P	7 A	8 *	9 C*4	10 A*3	11 O*G	12 T-1	13 T-2	14 T-2	15 T-2	16 T-3	17 Z-1	18 Z-1 * T-2	19 Z-1 * T-2	20 Z * T-2	21 Z * T-1	22 Z-1 * T	23 Z*T
Has Hair	0	0	0	0				0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
Has Feathers					0	0	0	0		0	0												
Gives Milk	@	@	@	@				0															
Flies							@	0	*														
Does Not Fly					0	0		0			0												
Flies Well							0	0		0													
Lays Eggs					@	@	@	0															
Eats Meat	@	@						0															
Has Pointed Teeth	0	0						0	0			0	0	0	0	0		0	0	0	0	0	0
Has Claws	0	0						0	0			0	0	0	*	*		0	0	0	0	0	0
Has Forward Eyes	0	0						0	0			*	*	*	*	*		*	*	*	*	0	0
Has Hoofs			0	0				0	*		0						0	0	0	0	0	0	0
Chews Cud			@	@				0															
Has Tawny Color	0	0	0					0	0		0	0	0	*	0	*		*	0	*	0	0	0
Has White Color				0				0									0	*	0	0	0	*	0
Is Black and White					0	0		0	*	*	0												
Has Dark Spots		0	0					0	0		0												
Has Black Stripes	0			0				0				0	*	0	0	0	*	0	*	0	0	0	0
Has Long Legs			0		0			0	*	*	0												
Has A Long Neck			0		0			0		*	0												
Swims						0		0															
活性化した動物	T	C	G	Z	O	P	A	*	C	A	O	T	*	T	T	*	Z	Z	*	Z	Z	T	T
活性化時間T(τ)	11.4	11.5	10.9	13.3	9.5	9.7	11.1	-	12.5	13.5	11.9	15.5	-	31.8	16.6	-	31.2	37.8	-	13.3	14.5	11.4	14.1
@も与えた場合T2(τ)	8.9	8.9	9.1	9.2	8.5	8.5	8.4																

0:与えた特徴 @:誤りではないが冗長な特徴 +:誤って与えた特徴 *:足りない特徴

れて，動物名に対応するモータノードの活性度が0.9以上になるまでの時間をτを単位として測ったものである．正しく特徴を与えた場合には 8.4τ から 13.3τ の間に対応する動物名が正しく活性化する．21個のすべての特徴を与えた場合は，どの動物名も活性化しない．これは，wta ネットワークの効果である．証拠の数が増えても結論は，減少するという非単調な特性を示している．本システムは確信度付の推論を行う．したがって，正しい特徴の組合せに，いくつかの誤った特徴を付け加えたり，または，必要な特徴を取り除いた場合にも，結論を出すことができる．

表 2 は動物の特徴（証拠）が動物名（結論）の推論にどの程度寄与しているかを示した物である．この表は次のようにして作成した．

(if (a b c) then (x)),

(if (x y) then (z)). (7)

というルールを考える．仮説 z は2つの証拠 x, y によってサポートされているので，x と y の z に対する寄与の程度はそれぞれ 1/2 とする．x は a, b, c の3つの証拠によってサポートされているので，a, b, c の x に対する寄与の程度は 1/3 である．a, b, c の z に対する寄与の程度は 1/2 と 1/3 を乗じて 1/6 とする．**表 3** は表1の場合 1〜23 で与えた特徴の集合とその動物名に対する寄与の程度を計算したものである．表2，3の計算はノードの特性は静的で線形としており，また wta ネットワークの効果も考慮

表 2 特徴（証拠）が動物名（結論）をサポートする強さ
Table 2 Contributions of an animal feature to animal species.

特 徴	Tiger	Cheetah	Giraffe	Zebra	Ostrich	Penguin	Albatross
Has Hair	0.250	0.250	0.100	0.167	-----	-----	-----
Has Feathers	-----	-----	-----	-----	0.200	0.250	0.500
Gives Milk	0.250	0.250	0.100	0.167	-----	-----	-----
Flies	-----	-----	-----	-----	0.100	0.125	0.250
Does Not Fly	-----	-----	-----	-----	0.200	0.250	-----
Flies Well	-----	-----	-----	-----	-----	-----	0.500
Lays Eggs	-----	-----	-----	-----	0.100	0.125	0.250
Eats Meat	0.250	0.250	-----	-----	-----	-----	-----
Has Pointed Teeth	0.083	0.083	-----	-----	-----	-----	-----
Has Claws	0.083	0.083	-----	-----	-----	-----	-----
Has Forward Eyes	0.083	0.083	-----	-----	-----	-----	-----
Has Hoofs	-----	-----	0.100	0.167	-----	-----	-----
Chews Cud	-----	-----	0.100	0.167	-----	-----	-----
Has Tawny Color	0.250	0.250	0.200	-----	-----	-----	-----
Has White Color	-----	-----	-----	0.333	-----	-----	-----
Is Black And White	-----	-----	-----	-----	0.200	0.250	-----
Has Dark Spots	-----	0.250	0.200	-----	-----	-----	-----
Has Black Stripes	0.250	-----	-----	0.333	-----	-----	-----
Has Long Legs	-----	-----	0.200	-----	0.200	-----	-----
Has A Long Neck	-----	-----	0.200	-----	0.200	-----	-----
Swims	-----	-----	-----	-----	-----	0.250	-----

していないので実験のニューラルネットワークの動作とは必ずしも一致しないがある程度の目安を与える．

表3の下線~~~で示した動物名がニューラルネットワークによる推論結果である．ほとんどの場合で，与えられた特徴の集合から最大のサポートを受けている動物名と推論結果が一致する．

場合11はキリン (Giraffe) とダチョウ (Ostrich) の特徴を両方与えた場合である．表3では，キリン，ダチョウのサポートはどちらも1.0であるが推論結果はダチョウである．これは2つの推論の深さの違いが影響している．キリンを検証するにはそれが有蹄類 (Ungulate) であること，ダチョウを検証するにはそれが鳥類 (Bird) であることを検証しなければならない．有蹄類の検証には2段，鳥類の検証には1段の推論が必要である．このため is-ostrich* のほうが is-giraffe* より先に活性化する．wta ネットワークの特性により活性化した is-ostrich* が is-giraffe* を含めて他の動物に対応するノードを抑制してしまう．

表3 特徴の集合（表1の場合 1〜23）が動物名をサポートする強さ
Table 3 Contributions of animal feature set to animal species.

場合	Tiger	Cheetah	Giraffe	Zebra	Ostrich	Penguin	Albatross
1 T	1.000	0.750	0.300	0.500	-----	-----	-----
2 C	0.750	1.000	0.500	0.167	-----	-----	-----
3 G	0.500	0.750	1.000	0.333	0.400	-----	-----
4 Z	0.500	0.250	0.200	1.000	-----	-----	-----
5 O	-----	-----	0.400	-----	1.000	0.750	0.500
6 P	-----	-----	-----	-----	0.600	1.000	0.500
7 A	-----	-----	-----	-----	0.200	0.250	1.000
8 *	1.500	1.500	1.200	1.333	1.200	1.250	1.500
9 C+4	0.750	1.000	0.800	0.333	0.500	0.375	0.250
10 A+3	-----	-----	0.400	-----	0.800	0.500	1.000
11 O+G	0.500	0.750	1.000	0.333	1.000	0.750	0.500
12 T-1	0.917	0.667	0.300	0.500	-----	-----	-----
13 T-2	0.667	0.667	0.300	0.167	-----	-----	-----
14 T-2	0.667	0.417	0.100	0.500	-----	-----	-----
15 T-2	0.833	0.583	0.300	0.500	-----	-----	-----
16 T-3	0.583	0.333	0.100	0.500	-----	-----	-----
17 Z-1	0.250	0.250	0.200	0.667	-----	-----	-----
18 Z-1 + T-2	0.667	0.417	0.200	0.667	-----	-----	-----
19 Z-1 + T-2	0.667	0.667	0.400	0.667	-----	-----	-----
20 Z + T-2	0.667	0.417	0.200	1.000	-----	-----	-----
21 Z + T-1	0.917	0.667	0.400	1.000	-----	-----	-----
22 Z-1 + T	1.000	0.750	0.400	0.667	-----	-----	-----
23 Z + T	1.000	0.750	0.400	1.000	-----	-----	-----

場合 13 はトラの特徴から2つ取り除いた場合である．これはチータ (Cheetah) の特徴から2つ取り除いたものと同じである．表3を見るとトラとチータへのサポートはどちらも 0.667 である．この場合は is-tiger* も is-cheetah* もまったく同じ投票を受け，wta ネットワークによりお互いに抑制しあうので結局どちらも活性化できない．

場合 16 は，トラの特徴から3つ取り除いた場合である．表3ではトラが最大のサポートを受けているが，実際の推論では結論が出なかった．証拠が少なすぎて IsTiger* が活性化できなかったためである．

場合 18 (図18) は，表3ではトラ，シマウマへのサポートはどちらも0.667である．推論結果はシマウマである．結論が出るまでの時間は 37.8τ で非常に長い．推論の段数を考慮しても与えた特徴はどちらに有利かは明確でない．is-tiger*，is-zebra* ともに同じ程度のサポートを受け，しばらくは両方の活性度が拮抗しながら大きくなり，ある時点でその拮抗が解け少しだけ有利であった is-zebra* が活性化したと考えられる．

以上いくつかの場合について推論結果の分析を行った．本推論システムの挙動は人が不十分な情報から判断を下す過程と類似点があり，人の推論過程の一部の簡単なモデル化になっているともいえる．

本アプローチでは推論の速度はルールの数ではなく，推論の段数に依存する．診断型のルールベースシステムではルール数の増大に対して，推論の段数はある程度以上は深くならない．ルール数が増大するとそれに連れてニューラルネットワークの規模も大きくなる．現在，ニューラルネットワークのシミュレーションは，従来型の計算機上で逐次的に行っているのでシミュレーション速度はニューラルネットワークの規模（ノード数，リンク数）に比例して遅くなる．しかし，ニューラルネットワークの並列性を引き出せるハードウェアがあれば大規模なルールベースでも高速な推論が実現できる．

本推論システムのルールは変数を扱うことができない．変数の機能がなくても EMYCIN や Prospector のようなルールベースシステムは実現できる[6),9)]．しかし，一般には変数の機能がないと知識の表現力に制限ができる．並列推論システムとしての特徴を失わずに変数の（それと等価な）機能を実現することが今後の課題である．

5. おわりに

情報処理の基本要素としてノードが備えるべき特性を検討し，時系列パターンを扱うために内部状態を持つ時間連続のニューラルネットワークモデルを提案した．またニューラルネットワークのプログラミング環境を開発した．記述言語は，ひとまとまりの機能を実現する部分ネットワークをテンプレートとして定義し，さらにそのテンプレートを使ってより複雑なネットワークを定義する機能を持つ．

前向き推論の実験では，誤ったデータや，不十分なデータに対してもシステムが動作しうることを示した．4.3節で構成したANDノード以外にもANDノードとORノードの中間的なノードを構成することもできる．今後は，後向き推論を行うネットワークについて検討を行う．

現在はニューラルネットワークプログラミング環境はLISPマシン上にインプリメントしているが今までに行った小規模な実験でもノードやリンクの数が多くなると実行速度が非常に遅くなる．これは逐次方式の計算機で並列な計算をシミュレートしているためである．より大規模な実験や実際のアプリケーションの開発をするにはニューラルネットワークの並列性を引き出せるハードウェアが必要である．現在1PE（プロセッサエレメント）あたり，1Mリンク/秒の処理速度を有する専用並列マシン[11]を開発中である．

謝辞 本研究を行う機会を与えていただくとともに，有益なご意見をいただいた，大阪大学基礎工学部の辻三郎教授，日本電気(株) C&Cシステム研究所小池誠彦部長に感謝いたします．

参考文献

1) 梶原信樹，辻　三郎：ニューラルネットワーク推論システム，情報処理学会知識工学と人工知能研究会資料，45-10 (1986)．
2) Feldman, J. A. and Ballard, D. H.: Connectionist Models and Their Properties, *Cognitive Science*, Vol. 6, pp. 205-254 (1982).
3) Feldman, J. A., Fanty, M. A. and Goddard, N. H.: Artificial Neural Systems, *IEEE Comput.*, Vol. 21, No. 3, pp. 91-103 (1988).
4) 二見亮弘，星宮　望：相互結合型神経回路網の状態遷移に基礎をおく時系列パターン認識の神経回路モデル，電子情報通信学会論文誌，Vol. J71-D, No. 10, pp. 2181-2190 (1988)．
5) 大森隆司，佐藤克己：時系列を認識・想起する神経回路のモデル，第2回生体生理工学シンポジウム，1A-1-3 (1987)．
6) Blelloch, G. E.: *CIS: A Massively Concurrent Rule-Based System*, AAAI (1986).
7) Touretzky, D. S. and Hinton, G. E.: Symbols among the Neurons: Details of a Connectionist Inference Architecture, IJCAI, pp. 238-243 (1985).
8) Touretzky, D. S. and Hinton, G. E.: A Distributed Connectionist Production System, *Cognitive Science*, Vol. 12, pp. 423-466 (1988).
9) Samad, T.: Towards Connectionist Rule-based Systems, *ICNN*, Vol. 2, pp. 525-532 (1988).
10) Winston, P. H. and Horn, B. K. P.: 情報処理シリーズ4 LISP（白井良明，安部憲広共訳），pp. 237-239，培風館 (1982)．
11) 梶尾信樹，中田登志之，松下　智，小池誠彦：並列ニューラルネットワークシミュレータ，情報処理学会計算機アーキテクチャ研究会資料，88-ARC-71 (1988)．
12) Jordan, M. I.: Supervised Learning and Systems with Excess Degrees of Freedom, MIT COINS Technical Report, 88-27 (1988).
13) Servean-Schreiber, D., Cleeremans, A. and McClelland, J. L.: Encoding Sequential Structure in Simple Recurrent Networks, CMU Technical Report, CMU-CS-88-183 (1988).

(平成元年5月1日受付)
(平成元年12月12日採録)

梶原　信樹（正会員）

昭和56年山口大学工学部卒業．昭和58年大阪大学大学院基礎工学研究科修士課程修了．昭和61年同博士課程修了．日本電気(株)入社．C&Cシステム研究所にて，並列回路シミュレーションマシン，並列ニューラルネットワークシミュレーションマシンの研究開発に従事．AI，並列アーキテクチャ，ニューラルネットワークに興味を持つ．電子情報通信学会，人工知能学会，神経回路学会各会員．