

ニューラルネットワーク推論システム

梶原 信樹

辻 三郎

(大阪大学基礎工学部)

1. まえがき

ニューロン状素子を使ったネットワーク構造で、制御、推論を行なうシステムについて述べる。ネットワークは、活性度を持つノードと重みつきのリンクで構成される。各ノードは重みつきのリンクを介して互に影響を与え合いながら活性度を変化させる。人工知能システムでは、処理の並列化が大きな課題となっているが、このようなネットワーク構造は、基本的に並列に動作する。また、同じ構造を持つ多数のノードによって構成されるネットワークは、将来のVLSI技術によるハードウェア化も期待される。

ニューラルネットワークによって情報処理を行なわせる研究はパーセプトロンを始め数多いが、それらはリンクの重みを学習により変えて認識や連想を行なわせるものが殆どであった⁴⁾。本研究は、推論などを行なう知識システムをニューラルネットワークで実現する一つのステップとして、与えられた処理機能をニューラルネットワーク上にプログラムする方式をシミュレーションにより検討した。このためリンクの重みは予め設定した値に固定されているが、ノードの活性度の変化によって所要の処理が実現される(学習による重みの変更は次の研究課題である)。ネットワークのプログラムを行なうには、よく現れる部分ネットワークのパターンをモジュール化して、さらに複雑なネットワークのプログラムをするときにそのモジュールを部品として使用できる機能が必要である。本研究ではネットワークをプログラムするときのソフトウェア的な側面についても述べる。ネットワークのプログラムのツールとして、よく利用する部分ネットワークのパターンをモジュール表現し、それらを利用して複雑なネットワークをシミュレーションするシステムを構築した。これは推論、認識にのみネットワーク構造を使うのではなく外界とのインタフェースを行なうセンサ、モータをもネットワーク構造の中に入れて一つの完結したシステムを目指している。最終的にはネットワークの大まかなプログラムは人間が行ない後は、与えられた環境に自動的に適応するというシステムを目標にしている。

2. システム構成

システム構成を図1に示す。ネットワークを構成するノードには、センサ、モータ、コンテキストの3種類がある。センサは、環境の物理量、状態に反応し活性度を変化させる。センサの活性度は、重みつきリンクによってネットワーク内の他のノードに影響を与える。モータは、ネットワーク内の他のノードから影響を受けて活性度を変化させる。そして、その活性度に応じて環境の物理量、状態に変化を与える。コンテキストは、ネット

ワークの本体で、システムの動作を決定するのは、この部分である。コンテキストノードは、他のノードから影響を受け、また他のノードへ影響を与える。しかし、センサノードは環境のみに影響を受け、他のノードからは影響を受けない。モータノードは環境に影響を与える以外は他のノードに影響を与えない。ノードの活性度は正負の実数値をとるが、この活性度の意味としては特定のものとは考えてない。実世界の曖昧性を扱うために導入したノードの持つ属性である。ノードが一つの仮説を表わしているとすると、そのノードの活性度は、仮説の信頼度と考えることもできる。またノードがセンサであれば、その活性度はセンサの計測する物理量を反映した値となる(後で述べる様にセンサの活性度と環境の関係は自由に定義できる)。

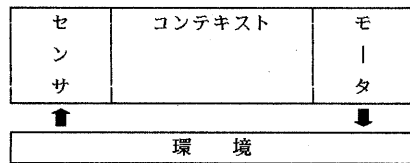


図1 システム構成

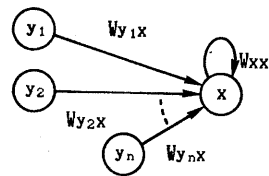


図2 仮説と証拠

3. ノードの基本動作

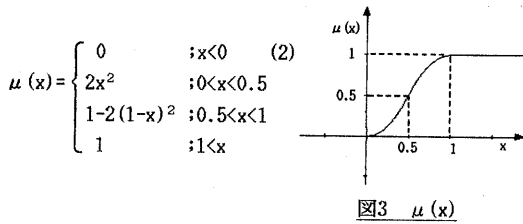
ネットワーク内のノードが他のノードから影響を受けてどのようにその活性度を変化させるかについて述べる。ネットワークのノードの活性度は、正負の実数の範囲で変化する。図2のノードxの活性度は他のノード y_1, y_2, \dots, y_n から影響を受けて変化する。 $W_{y_i x}$ は、ノード y_i からxへのリンクの重みで、正負の実数値を持つ。 W_{xx} はxからx自身への重みである。各ノードの活性度を x, y_1, y_2, \dots, y_n とすると(以後区別が必要な場合を除いてノードxの活性度のことを単にxと書く) xは次の微分方程式に従って変化する。

$$\tau \frac{dx}{dt} = \sum W_{y_i x} \mu(y_i) ; \text{他のノードからの影響} \quad (1)$$

$$+ W_{xx} \mu(x) ; \text{自分自身からの影響}$$

$$- x ; \text{減衰項}$$

(1)式を以後、動作方程式と呼ぶ。 τ は時定数で以後 τ を時間の単位とする。 μ は(2)式、図3に示す様なリミター特性を持つ関数である。



動作方程式の他のノードからの影響の項(右辺第1項)をAと書き、これをノードxへの投票と呼ぶ。Aを使うと動作方程式は次の様に書ける。

$$\tau \frac{dx}{dt} = A - W_{xx} \mu(x) - x \quad (3)$$

自分自身も含めて他からの影響がない場合は、ノードの活性度は指数関数的に減衰する。

図2の x と y_1, y_2, \dots, y_n を仮説とその証拠と考えると μ の特性は、次の様に解釈できる。 $W_{y_1x}, W_{y_2x}, \dots, W_{y_nx} > 0$ とすると、証拠 y_1, y_2, \dots, y_n の活性度が大きいと仮説 x の活性度が大きくなる。しかし、活性度が0.5より小さな証拠は x の活性度に与える影響は小さい。活性度が負の証拠は、仮説 x の活性度には影響を与えない。例えば『プロペラを持つ』ことは『飛行機である』ことの証拠となる。しかし『プロペラを持つ』は『飛行機である』の必要条件ではないので、『プロペラを持つ』が否定的でも『飛行機である』を否定する証拠にはならない。動作方程式(1)より導かれるノードの特性を分析する。他のノードからの投票Aを0としたときの $\tau \frac{dx}{dt}$ を図4に示す。 $W_{xx} > (2 + \sqrt{2})/4$ の曲線はx軸との交点を3つもつ。このうち原点と右端の交点は安定な点で x の活性度はA=0のときは2つの安定な値をとり得ることを示す。例えば $W_{xx} = 1$ であれば、 x の活性度は0と1で安定である。すなわち $W_{xx} > (2 + \sqrt{2})/4$ で自分自身に影響を与えるノードは、双安定のメモリとしての機能を持つ。図4の曲線は、他のノードからの投票Aの値によって上下にシフトする。 $|A|$ が十分に大きく十分に長い時間与えられれば双安定なノードは、一方の状態から他の状態になる。例えば $W_{xx} = 1$ のノード x が $x=0$ の状態にあるとき、 $A=1$ なら 0.546τ 、 $A=0.2$ なら 4.708τ 以上与えられれば状態を変え、それ以後 $A=0$ となっても $x=1$ になる。 $A < 1/8$ であればいくら長い時間与えられても状態を変えることはない。ノード x のインパルス応答とステップ応答を図5,6に示す。図5のインパルス応答では、 $W_{xx} < (2 + \sqrt{2})/4$ の場合は、 $t \rightarrow \infty$ で $x \rightarrow 0$ となるが、 $W_{xx} > (2 + \sqrt{2})/4$ の場合は、0以外の正の値に収束する。図6のステップ応答は、 $W_{xx} = 1$ のノードが他からの投票Aをステップ状に与えられたときの応答で $A=1$ なら 0.546τ で $x=0.5$ となる。 $x > 0.5$ となれば以後は $A=0$ となっても $x \rightarrow 1$ になる。

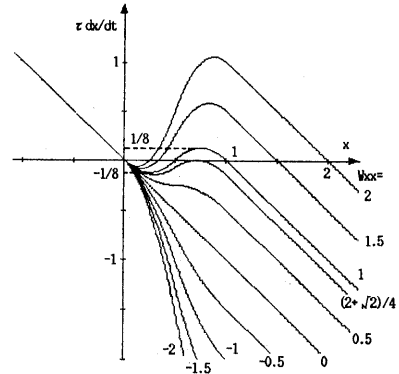


図4 $\tau \frac{dx}{dt} = W_{xx} \mu(x) - x$

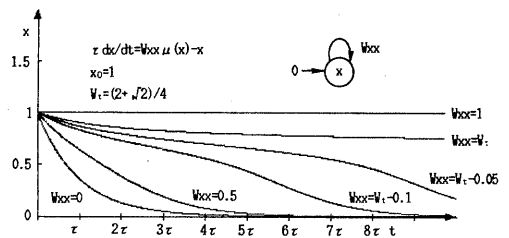


図5 インパルス応答

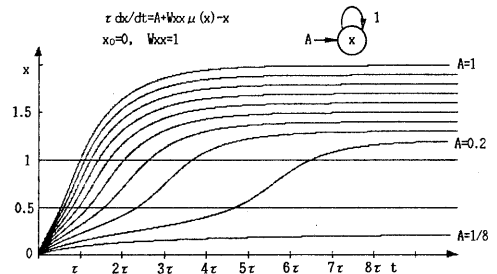


図6 ステップ応答

4. ネットワークのプログラム

今までもニューラルネットワークの研究は行なわれているが、ネットワークを積極的にプログラムしようという研究は、あまりない。本研究の目的の一つは、ニューラルネットワークを積極的にプログラムして、ニューラルネットワークでどんな機能が実現できるか調べることである。この節では、動作方程式(1)で定義される特性を持つノードで構成されるネットワークのプログラミング法について述べる。

4.1 センサ、モータ、環境の定義

ネットワークが、どのような環境で動き、どのようなセンサ、モータを持っているか、そしてセンサ、モータの活性度と、環境の内部状態との関係を定義する必要がある。これは関数def-environmentで行なう。図7にdef-environmentのシンタックスを示す。

```

(def-environment NET      ;ネットワーク名
 :sensors (S1 S2 ... Si) ;センサ名のリスト
 :motors (M1 M2 ... Mj) ;モータ名のリスト
 :states (ST1 ST2 ... STk) ;環境の内部状態名のリスト
 :initialize INIT       ;環境の初期化のときに実行されるLISPプログラム
 :advance ENV           ;環境をシミュレートするLISPプログラム

```

図7 def-environmentのシンタックス

ネットワークは、動作方程式(1)に従って動く。本システムでは、動作方程式は、 Δt ごとにオイラー法によって解いている。def-environmentの中のENVは、 Δt ごとに実行され、モータ M_1, M_2, \dots, M_j の活性度に応じて環境の状態 ST_1, ST_2, \dots, ST_k を変化させる。そしてその状態に応じてセンサ S_1, S_2, \dots, S_i の活性度をセットする。ネットワークが実世界の中で動く場合はENVは単なる仮想的な環境のシミュレータではなく、実世界と、センサ、モータとのインタフェースを行なう(図8)。例えば、キーボードとディスプレイを使って人と対話するネットワークでは、アルファベット26文字に対応してそれぞれ26個のセンサと26個のモータを持つ。ENVは Δt ごとにキーボードをスキャンし押されているキーに対応するセンサの活性度を1にする。また、26個のモータのうち、活性度が最大でその活性度が閾値を越えているものに対応する文字をディスプレイに出力する。

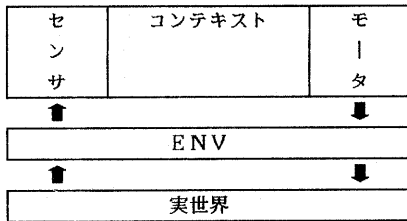


図8 実世界と、センサ、モータとのインタフェース

4.2 ネットワークの定義

ネットワークの定義は、ネットワーク内のセンサノードとモータノードを介在するコンテキストノードにどの様なノードが存在し、それらがどの様に影響を及ぼしあうかを定めることである。ネットワークを構成するノードとリンクは、それぞれある概念と概念間の関係と見ることができる。即ち、ネットワークの定義は、いろいろな概念間の関係を定義することである。以下ではノード(センサノード、モータノード、コンテキストノード)のことを概念と呼ぶことがある。ネットワークを定義するとき、よく現れる概念間の関係をシステムはテンプレートとしていくつか用意している。ユーザは、それらを使ってより複雑な関係を定義することができる。最も基本的なテンプレートとしてifc(IF Coincide then activate)を用意している。ifcのシンタックスと意味を図9に示す。

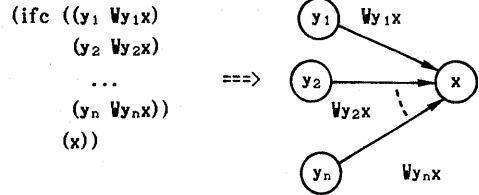


図9 ifc(IF Coincide then activate)

y_1, y_2, \dots, y_n, x は概念名、 $Wy_{ix}, Wy_{2x}, \dots, Wy_{nx}$ は数である。概念名は、各概念に固有の名前である。この名前によってネットワーク内の概念を指定することができる。ここで強調すべきことは、ネットワークを定義するときには、ある特定の概念を指定するために、この概念名が使われるが、ネットワークの動作時には概念名はまったく意味を持たないということである。図9の関係を実際にネットワークとして展開するためには関数teachを使う。

```

(teach NET (ifc ((y1 Wy1x) (y2 Wy2x) ... (yn Wynx))
 (x)))

```

y_1, y_2, \dots, y_n, x の中ですでにネットワークNETに存在する概念名があるときには、同一の概念を指定したことになる。また、まだ存在しない概念名があれば、新たにネットワークに概念がつけ加えられる。基本的には、ifcがあればどんなネットワークでも定義できるが、これだけでは不便である。ユーザが概念間の関係を定義し、さらにその関係を使ってより複雑な関係を定義できる機能が必要である。関係を定義するには関数def-relationを使う。2つの概念 x, y の活性度がほぼ同じであれば、活性化する概念 $x=y$ は、図10の様に構成できる。→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10の関係は、def-relationを使って(5)式の様定義できる。

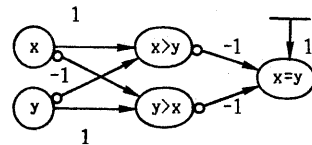


図10 equally-active

```

(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)))

```

$x, y, x=y$ は仮引数である。 $x>y, y>x$ はローカルな概念として宣言され、他からはこの概念を参照することはできない。仮引数とローカルな概念以外、(5)式では $*true*$ はグローバルな概念である。グローバルな概念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の概念である。(5)式の(declare ...)以降の3つの(ifc ...)は、それぞれ幾つかの概念の関係を定義している。この3つの式の順序はequally-activeの機能には、まったく影響を与えないことに注意されたい。equally-activeは、次の様に使う。

```
(teach NET (6)
  (equally-active left-eye right-eye the-front))
```

これで、図10の $x, y, x=y$ をそれぞれleft-eye, right-eye, the-frontに、 $x>y, y>x$ を他から参照できない名前を持つ2つの概念に変えた関係が指定したネットワークNETにつけ加えられる。def-relationで定義した関係は、より複雑な関係を定義するときの部品として使用できる。(6)式は、equally-activeの内部構造を無視して図11の様に書くことができる。

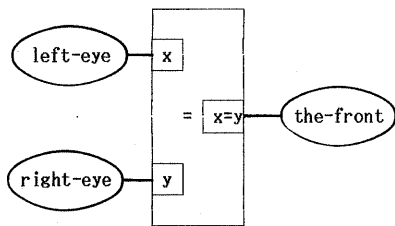


図11 部品としてのequally-active

引数の数が不定個の関係は、現在のところdef-relationでは定義できないので、システムの基本的な関係としてLISPで定義している。

ネットワークの定義をしていると、概念yから概念xへのリンクが重複して定義されることがある。そのときには、それぞれの定義の平均を W_{yx} の値とする。すなわち、ある関係1の中で概念yから概念xへのリンクが $W_{yx}=W_1$ と定義され、関係2の中で $W_{yx}=W_2$ と定義されると $W_{yx}=(W_1+W_2)/2$ となる。これによって、ネットワーク全体としては、関係1と関係2で定義される動作を平均した様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やすくする原因となるかもしれないが、今までに行なった小規模な実験では、問題は生じてない。

5. 実験

この節では、システムが用意する基本的な関係について述べる。また、応用例としてP.H. Winston¹⁾の動物当てシステム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

する関係の構成は、これが最良というものではなく、もっと良い構成があるかもしれないということである。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。

5.1 wta(Winner Take All) ネットワーク

いくつかの競合する概念は、図12の様なネットワークで構成できる。各ノードa, b, cが競合する概念を表す。Aa, Ab, Acは、それぞれ概念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Aa, Ab, Acの投票をう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概念だけが活性化し他は抑制される。ある一つ概念、例えばa、が活性化した後には $Aa=Ab=Ac=0$ となってもaは活性を保持する。即ち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。Aa, Ab, Acがほぼ同じ値であれば、どの概念も活性化できない。また、最大の投票が1/8より小さいときにもどの概念も活性化できない。

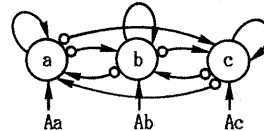


図12 wtaネットワーク

図12で $a=1, b=c=-1$ 即ちaが活性化している状態から $b=1, a=c=-1$ 即ちbが活性化している状態に変化させるためには $Aa < -1/8, Ab > 1/8, Ac = 0$ としてしばらくこの投票を保持すれば良い。図4の $\tau dx/dt$ の特性を見ると $W_{xx}=1$ の概念は、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。 $Aa < -1/8$ 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 $Ab > 1/8$ 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。

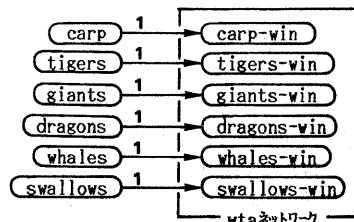
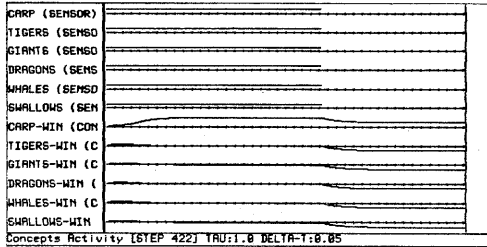


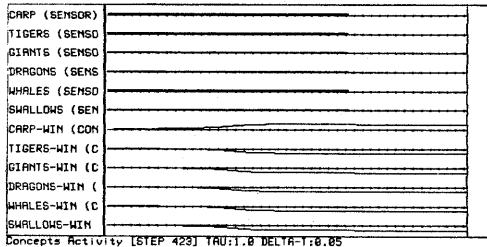
図13 セントラルリーグ

図13は、wtaネットワークの実験に使ったネットワークである。carp, tigers, giants, dragons, whales, swallowsはセンサで、環境をシミュレートするプログラムによりユーザの指定した活性度に固定される。carp-win, tigers-win, giants-win, dragons-win, whales-win, swallows-win

は、コンテキストでwtaネットワークを構成している。実験結果を図14に示す。図14はネットワークのデバッグ用に作ったモニタの画面の一部である。グラフの縦軸が活性度(軸と軸の間隔が4)、横軸が時間で目盛は 0.5τ である。図14(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が唯一活性化される。その後センサの活性度を0にしてもcarp-winの活性度は保持される。図14(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, carp以外は0.8



(b) carp=0.3, carp以外は0.2

図14 セントラルリーグの結果

5.2 ifs (IF Sequence then activate)

一連の概念 a_1, a_2, \dots, a_n が順に活性化されたときに活性化される様な概念 x を構成したいことがある。例えば h, e, l, l, o, w という概念がこの順に活性化したときに $helloworld$ という概念が活性化する様にしたい。ifs (IF Sequence then activate) は、このような関係を定義するのに使う。

$$(ifs\ h\ e\ l\ l\ o\ w\ helloworld) \quad (7)$$

で上の関係が定義できる。ifsに要求される特性としては以下の事項がある。

- a) シーケンス a_1, \dots, a_n によって x が正しく活性化する。
- b) シーケンスの一部が次の $b1), b2), b3)$ の様に不完全でもその程度に応じて x が活性化する。
 - b1) シーケンスの一部が欠ける。
 - b2) シーケンスの一部に余分なものが入る (すなわち a_i の直後に a_{i+1} が活性化しない)。
 - b3) 各 a_i の活性化の時間間隔にバラつきがある。

しかしシーケンスがどの程度正しく生じると、どの程度 x を活性化させれば良いかは一意に決められるものでは

なく、用途に応じていろいろな特性が必要になる。現在はifsは1種だけ用意している。その構成法を以下に述べる。ifsを構成するために2種の部品ifs-aux1, ifs-aux2を用意する(図15)。ifs-aux1とifs-aux2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 t_1 が t_2 を抑制しトリガ t が活性化しても y は活性化しない。 x が活性化すると t_2 への抑制がなくなる。そこでトリガ t が活性化すると、 t_2 を通して y が活性化する。ifx-aux2の場合は y によって x は抑制される。このとき t_2 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つの部品を使う

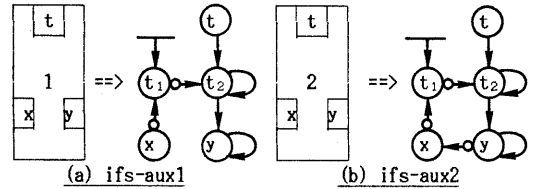
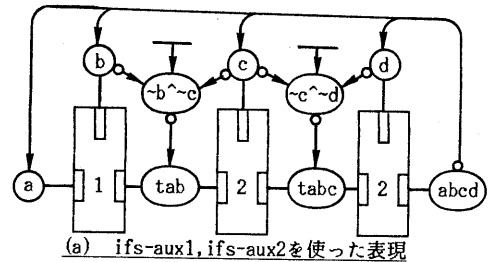
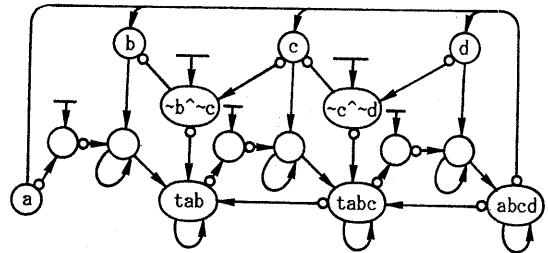


図15 ifsの構成部品



(a) ifs-aux1, ifs-aux2を使った表現



(b) 展開した結果

図16 (ifs a b c d abcd)

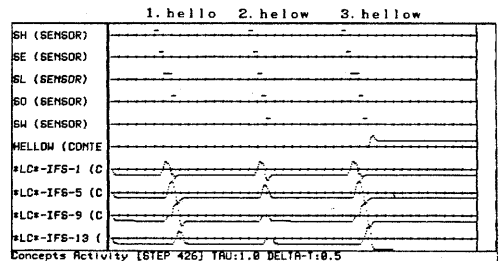


図17 (ifs sh se sl sl so sw hellow)の実験結果

と (ifs a b c d abcd) は、図16(a) の様に構成できる。図16(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、tabc は a, b, c が順に活性化したことを記憶するローカルな概念である。tab は a, b の直後に c が活性化しないと抑制される。実験結果を図17に示す。横軸の目盛は 5τ である。sh, se, sl, so, sw は、それぞれキーボードから h, e, l, o, w のキーが押されたときに 2.5τ だけ活性化度が 1 になるセンサである。hellow は h, e, l, l, o, w とキー入力があったときに活性化するコンテキストでセンサとの関係は (ifs sh se sl sl so sw hellow) で定義される。
 lc-ifs-1,5,9,13 は、それぞれ he, hel, hell, hello とキー入力があったときに活性化されるローカルなコンテキストである。図17の 1, 2, 3 は、それぞれ hello, helow, hellow と入力した場合で不完全な 1 と 2 ではコンテキスト hellow は活性化しない。正しく入力が行なわれた 3 の場合だけ活性化する。

5.3 動物の特徴から動物の種を決定するシステム

P.H.Winston¹⁾ の動物あてシステムのプロダクションルールをネットワークで構成し実験した。このシステム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システムである。図18に推論に必要な関係 zoo の定義を示す。(if...then...) は一つのプロダクションルールに対応し図19の様にネットワークに変換される。is-tiger* 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 つのノードの活性化度がすべて 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 木の AND ノードを構成したことになる。

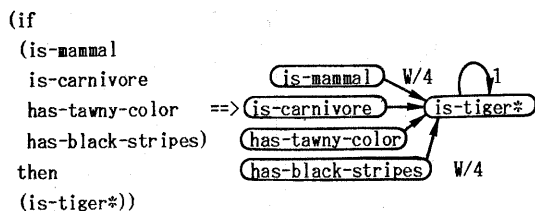


図19 プロダクションルールのネットワーク表現 (W=0.2)

zoo をネットワークに展開すると図20の様になる。図20のネットワークは、図18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ネットワークと、ほとんど同型のネットワークである。このことから関係 zoo の定義をネットワークに展開した時点ですですに必要な推論は半分済んでいるともいえる。この展開によって 40 個のノードと 102 本のリンクが作られた。図20のノードのうち他からのリンクを持たない物がセンサノードである。動物の一つの特徴 (21 個ある) に対して一つのセンサが対応する。21 個のセンサのうちユーザの選んだいくつかの特徴に対応するセンサは、環境をシミュレートするプログラムにより活性化度が 1 に固定される。動物の種に対応して 7 つのモータ

```

(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 Take 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-cheetah*) (is-cheetah))
  (ifc (is-tiger*) (is-tiger))
  (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18 動物あての知識 zoo

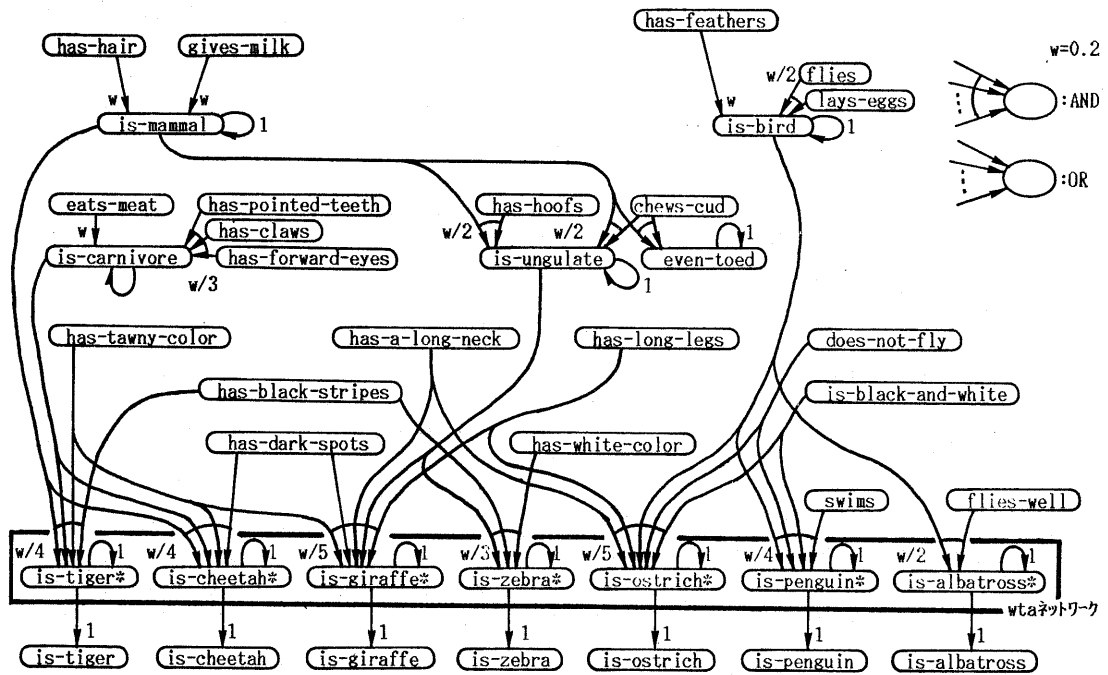


図20 動物当てシステムのネットワーク

表1 動物当てシステムの実験結果
T:Tiger C:Cheetah G:Giraffe Z:Zebra O:Ostrich P:Penguin A:Albatross

場合	動物当てシステムの実験結果																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
	T	C	G	Z	O	P	A	*	C+4	A+3	O+0	T-1	T-2	T-2	T-2	T-3	Z-1	Z-1	Z-1	Z	Z	Z-1	Z+T
Has Hair	0	0	0	0	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gives Milk	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Flies	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Does Not Fly	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Flies Well	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Lays Eggs	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Eats Meat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Has Pointed Teeth	0	0	0	0	0	0	0	0	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	0	0	0	0	0	0	0	0	0	0
Has Forward Eyes	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Has Hoofs	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chews Cud	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Has Tawny Color	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Has White Color	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Is Black and White	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Has Dark Spots	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Has Black Stripes	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Has Long Legs	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Has A Long Neck	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Swims	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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(%)	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
0も与えた場合T2(%)	8.9	8.9	9.1	9.2	8.5	8.5	8.4	-	8.9	8.9	9.1	9.2	-	8.5	8.5	8.4	8.9	8.9	9.1	9.2	8.5	8.5	8.4

0:与えた特徴 0: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

(is-tiger, is-cheetah, ..., is-albatross)がある。環境をシミュレートするプログラムは、7つのモータのうち活性度が0.9以上になったモータに対応する動物名を出力する。残りの12個はコンテキストで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個のコンテキストのうち7個(is-tiger*, is-cheetah*, ..., is-albatross*)は7つの動物の種に対応し、これが重み1でモータに影響を与える。そして、この7つのコンテキストは、wtaネットワークを構成しており、同時には唯一つのコンテキストしか活性度が大きくなれない。実験結果を図21, 22, 表1に示す。図21は動物の特徴としてトラの特徴を与えた場合(表1の場合1)である。横軸の目盛は 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11.4 τ 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なわれたことを示す。図18のzooの中のすべてのルールは常に実行されているともいえる。図22はトラの特徴から2つ取り除いたものとシマウマの特徴から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のがis-zebraの活性度が0.9を越えるのは37.8 τ 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れて、動物の種に対応するモータの活性度が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8.4 τ から13.3 τ の間に対応する種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、どの種も活性化しない。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。正しい特徴の組合せに、いくつか誤った特徴をつけ加えたり、又は、必要な特徴をとり除いた場合にも、結論を出すことができ、その結論は人間が見ても納得の行くものである。

5. むすび

ここで示した他にもいくつかの実験を行なった。あるノードが活性化したときに、それがトリガとなって一連のノードを順に活性化するネットワークや、wtaを使って変数の機能(変数名に対応するノードが活性化することによって、その変数の値に対応するノードが活性化する)も実現できる。プロダクションルールの実験では、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得ることを示した。5.3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。今後は後方推論を行なうネットワークや半自動によるネットワークのチューニングについて検討を行ないたい。現在はシステムはLMIのLISPマシンlambda上にインプリメントしているが今までに行な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。例えば、図13のセントラルリグ(ノード数12、リンク数42)では1ステップ(Δt 毎の積分)に要する時間は0.22秒、図20のzoo(ノード数40、リンク数102)では0.28秒である。これは逐次型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ネットワークのハードウェア化が絶対に必要と考える。

6. 参考文献

- 1) P.H. Winston, B.K.P. Horn; 培風館; 情報処理シリーズ4 LISP; pp.237-239
- 2) J.A. Feldman, D.H. Ballard; "Connectionist Models and Their Properties"; COGNITIVE SCIENCE 6 (1982), pp.205-254
- 3) D.S. Touretzky, G.E. Hinton; "Symbols among the Neurons: Details of a Connectionist Inference Architecture"; IJCAI (1985), pp.238-243
- 4) 甘利俊一; 神経回路網の数理; 産業図書

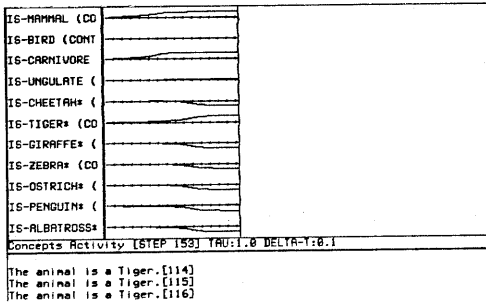


図21 トラの特徴を与えた場合

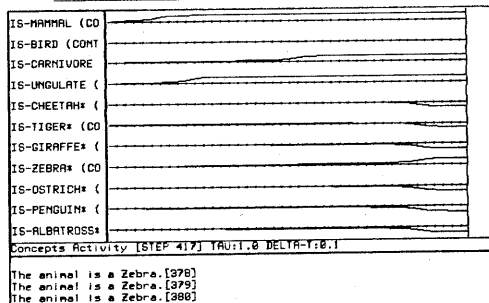


図22 シマウマ-1, トラ-2の特徴を与えた場合