

離散時系列の暗記学習メカニズムと 試験例題への応用

畝見 達夫

長岡技術科学大学 工学部 計画・経営系
unemi@voscc.nagaokaut.ac.jp

比較的粒度の小さな離散時系列を入出力データとする一種の学習メカニズムを提案し、簡単な試験例題への応用を通して解説する。学習システムの目的は入力データに関する評価関数の形で与えられ、経験を参考にしながら評価の高い入力を得られるよう各ステップの出力を決定する。個々の入出力データはその時間順序に従った連想リンクで結合され記憶される。また、過去の記憶と現在の入力を比較し、類似度の高い記憶を想起表と呼ばれる記憶領域に蓄える。この想起表中のデータを元に出力選択に伴う予想を行ない、予想される入力に対する評価から出力を決定する。想起および予想の過程では、リハーサルに基づく記憶の強化が行なわれ、重要な記憶が生き残る。

A Rote Learning Mechanism for Discrete Time-Sequence and Its Application for a Testbed

Tatsuo Unemi

Department of Planning and Management Sciences, Nagaoka University of Technology
unemi@voscc.nagaokaut.ac.jp

1603-1 Kamitomioka-machi, Nagaoka, Niigata 940-21, JAPAN

This paper proposes a learning mechanism for which input/output data are formed in discrete time-sequence of relatively small grain size. We show an application to a simple testbed to illustrate its properties. The objective of learner is given as a form of evaluating function of input data. The learner refers its own experience to make a decision in each step so that it will get valuable input in future. Each of input/output data is memorized in a node connected by association links in the order of time. The learner compares current input/output data with its experience, and stores the old data which have a relatively high correlation into a working memory named *recalling table*. Referring this table, it makes prediction to select output data for the next step, and decides from the value of predicted input data. During the process of recalling and prediction, a *rehearsal-based strengthening* of memory is done to let important data survive in the memory.

1 はじめに

筆者は、人間の学習モデルを目指して行なった研究 [1] をもとに、そこから得られた記憶に基づく学習メカニズムの一般化を試みた [2]。ここでは、初期の研究で用いたものよりもいくぶん複雑な例題への応用を通し、一般化された学習メカニズムの精緻化を試みた結果について報告する。

ここで提案される学習メカニズムは「暗記学習」の枠組みを基本とする。暗記学習は初期の人工知能において研究された手法であり、Samuel のチェッカ・プレイヤ [3] がその代表として知られている。暗記学習では、経験をそのまま記憶し、同じ状況が再び現われたときにその記憶を参考に行動を決定するものであり、帰納学習 [4] やチャンキング [5] における経験の一般化のような記憶の加工は行なわない。

学習の形態を特徴付ける要素として、暗記、帰納、類推といったメカニズムと共に、学習システムによって扱われる入出力データの構造が挙げられる。すなわち、どのような構文構造の入力データからどのような構文構造の出力データを学習するかということである。人工知能の分野では問題解決におけるルールや戦略の獲得あるいはパターン認識が応用分野として取り上げられてきたが、ここで提示する学習システムはそれらとは異なり、比較的粒度の小さな離散時系列を入出力データとする。この設定は現実環境の中に生きる生体の状況に由来する。すなわち、適当な時間間隔でサンプリングされたデータの列を入力とし、同様の間隔で出力信号を出す。際立った特質の1つは、入出力データの単位が不明確なことである。各時点でサンプリングされるデータは単独では情報が少なすぎるため、適当な長さのデータ列をひとまとまりとして扱う必要がある。しかし、どの範囲をひとまとまりとするかという問題に対しては外界からは何の明示的情報も与えられない。本モデルでは、敢えて特定の範囲をひとまとまりの単位として区切ることをせず、暗記された入出力データ列をそのまま利用する。

2 システムの概観

2.1 試験例題の概観

初期の研究 [1] で用いた「2次元平面の世界を餌を求めて歩き回る虫」という設定を踏襲する。ただし、世界(環境)の構造を、将棋盤のように格子状に粗く区切られたものから、連続な2次元ユークリッド平面にを拡張し、そのシミュレータを試験例題として用いる。世界には一定の大きさをもつ円形の「餌」と「糞」が適当な秩序の基に配置されている。学習システムの主体である「虫」は、「視覚」、「嗅覚」、「触覚(味覚)」の3つの感覚入力と、「歩幅」、「方向変換量」の2つの運動出力をもつ。虫の目的は、「糞に触らず餌を食べること」である。すなわち、触覚(味覚)以外のデータには何の先験的価値も与えられず、虫は経験を通して目的を達成する方法を学習しなければならない。図1に環境世界の表示例を示す。

視覚は非負整数を要素とする固定長ベクトルである。要素の個数は視細胞の数と考えてよい。要素の値は視覚に捉えられた物体から虫までの距離である。視野は図2のように虫を要とする扇形の領域を角度方向と半径方向に分割したもの

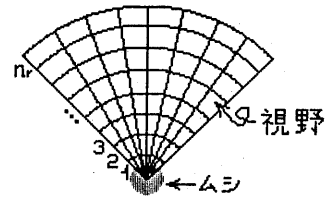


図2: 虫の視野

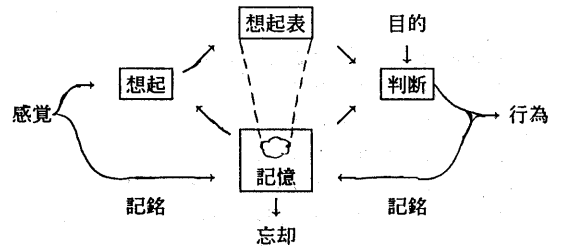


図3: 時系列に基づく学習モデル

である。

嗅覚は、餌の匂いを1、糞の匂いを-1とし、虫から一定距離の範囲内にある全ての物体についてそれぞれの匂いを距離で割った値の総和である。

触覚(味覚)は餌を食べたときは1、糞に触ったときは-1、何も無ければ0とする。

2.2 学習モデルの概観

図3に本モデルの構造を示す。想起、記憶、判断という3つの機能要素から構成される。それぞれの詳細については次の節で述べる。複数の結合されたヒューリスティクスに基づいて記憶要素と連想の強さを調整することにより、自らの経験を通して世界の法則を獲得し、結果として目的にそった行動決定を行なうようになる。

3 学習メカニズム

以下に学習モデルの形式的表現、および試験例題における具体化の例を示す。

3.1 記憶の基本構造

過去の経験の記憶 M は印象度 p を伴うノードの集合 N とリンクの集合 L から構成されるネットワークで表現する。このネットワークの中に入出力データの時系列を次々と記憶していく。また、思考の文脈(注視点)を表現するために想起表 R を利用する。 R は要素の個数の最大値 N_R があらかじめ制限された集合である。

$$M = \langle N, L \rangle \quad (1)$$

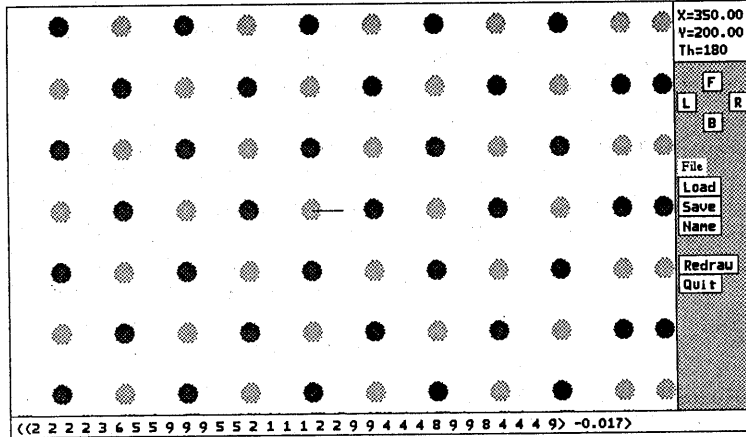


図 1: 試験例題の環境世界

$$N \subset \{(d, p) \mid d \in D, p \in [0, 1]\} \quad (2)$$

$$L \subset \{(n_1, n_2, p) \mid n_1, n_2 \in N, n_1 \neq n_2, p \in [0, 1]\} \quad (3)$$

$$R \subset \{(n, r) \mid n \in N, r \in [0, 1]\} \wedge |R| \leq N_R \quad (4)$$

ここで、 D は可能な感覚および運動命令の集合である。以下、各機能要素における具体的計算メカニズムについて解説する。

3.2 感覚データの入力

外界からの信号を適当な時間間隔で適当な型のデータとして入力する。このとき、主にデータの変化に依存した初期印象度を各データに割り付ける。データ変化が大きい程、印象度が大きくなるように、時刻 t における 1 つの入力データ d_t に対する初期印象度の値 p_t は次のような式で計算する。

$$p_t = \alpha \cdot p_{t-1} + (1 - \alpha) \cdot (1 - \Phi(d_{et}, d_t)) \quad (5)$$

ここで、 α は $\alpha \in [0, 1]$ なる定数、 p_{t-1} は前ステップにおける初期印象度、 d_{et} は感覚データの予測値、 $\Phi(x, y)$ は $\Phi(x, y) \in [0, 1]$ なる x と y との類似度である。 α が小さいほど、変化に対して敏感になる。 d_{et} および $\Phi(x, y)$ の計算法はデータ型に依存して定義されるべきものである。次にデータがスカラ量の場合における d_{et} および $\Phi(x, y)$ の計算式の例を示す。

$$d_{et} = \beta \cdot d_{et-1} + (1 - \beta) \cdot d_{t-1} \quad (6)$$

$$\Phi(x, y) = 1 - \frac{|x - y|}{d_{max} - d_{min}} \quad (7)$$

ここで、 β は $\beta \in [0, 1]$ なる定数であり、データ値が取り得る区間 $[d_{min}, d_{max}]$ が既知であるものとする。 $\beta = 0$ のときは $d_{et} = d_{t-1}$ つまり直前のデータ値が予測値となる。 β が 1 に近づくに従って予測値に対するデータ値の影響が小さくなり予測値の時間変化は緩やかなものとなる。 $\Phi(x, y)$ の値は x と y が最も似ているつまり $x = y$ のとき 1、 x と y が最も似ていないとき 0 となる適当な関数であればよい。試験例題において、嗅覚および触覚については (7) 式を用いた。

データがスカラ量を要素とする固定長ベクトルの場合、ベクトルの各要素について上の式を応用できる。 d_{et} をベク

トルとし、その n 番目の要素を d_{et}^i とすると、各要素の予測値および類似度は次のようになる。

$$d_{et}^i = \beta \cdot d_{et-1}^i + (1 - \beta) \cdot d_{t-1}^i \quad (8)$$

$$\Phi(x, y) = \frac{1}{n} \sum_{i=1}^n \left(1 - \frac{|x_i - y_i|}{d_{max} - d_{min}} \right) \quad (9)$$

ここで、 $x = (x_1, x_2, \dots, x_n)$ 、 $y = (y_1, y_2, \dots, y_n)$ である。また、ベクトルの各要素にもそれぞれに対応する印象度を与える。時刻 t における i 番目の要素 d_t^i に対する印象度 p_t^i は空間方向の変化も反映させて次のように計算する。

$$p_t^i = \lambda \cdot \{\alpha \cdot p_{t-1}^i + (1 - \alpha) \cdot (1 - \Phi(d_{et}^i, d_t^i))\} + (1 - \lambda) \cdot \{\alpha \cdot p_{t-1}^i + (1 - \alpha) \cdot (1 - \Phi_{st}^i)\} \quad (10)$$

ここで、

$$\Phi_{st}^i = \frac{\Phi(d_t^{i-1}, d_t^i) + \Phi(d_t^{i+1}, d_t^i)}{2}$$

λ は $\lambda \in [0, 1]$ なる定数である。また、 $d_t^0 = d_t^1$ 、 $d_t^{n+1} = d_t^n$ として計算する。試験例題における視覚については、これらベクトルに関する式を用いている。各入力データはそれぞれの初期印象度と組にされ、想起および記憶過程に送られる。

3.3 記憶と忘却

現時点の各入出力データのすべてを、直前のすべてデータとリンクで連結する。新たに記憶に追加されるノード集合 ΔN およびリンク集合 ΔL は次のように表現できる。但し、 P はノードからその印象度を求める関数、 T はノードからその生成時刻を求める関数であり、時刻は経過とともに増加する自然数で表現するものとする。

$$\Delta N = \{(d_c, p_c) \mid d_c \in C\} \quad (11)$$

$$\Delta L = \{(n_o, n_c, p_l) \mid n_c \in \Delta N, n_o \in N, T(n_c) = T(n_o) + 1, p_l = \delta \cdot P(n_o) + (1 - \delta) \cdot P(n_c)\} \quad (12)$$

ここで、 C は現時点の入出力データの集合、 p_c は d_c に対する初期印象度、 δ は $\delta \in [0, 1]$ なる定数である。

暗記学習では経験を経る毎に新たな記憶領域を次々と消費し続けることになる。生体の場合は生れつき一生分の記憶領域が用意されている可能性もあるが、現状で利用可能な計算機上で実現する場合には、メモリおよび計算コストの点から、このような無制限な記憶の消費は避けるべきである。1つの対処法は、 R と同様に N および L についても要素数を制限し、印象度の低い方から制限を上回った分の個数だけ削除するというものである。これにより、印象度の低い部分は忘却されることとなる。

3.4 類似性に基づく想起

各入出力データについて現時点の現実のものとの記憶データ中のものとの類似度を測り、類似性が高く印象度の高い記憶データを想起表に載せる。但し、既に想起表に載っているデータについては、その想起表中の印象度を増加させるに留める。 C 中のデータ d_c とその初期印象度 p_c および記憶中のノード (d_n, p_n) に対して、 R 上の印象度 r_t は次のように計算する。

$$r_t = \begin{cases} r + (1-r) \cdot p_c \cdot p_n \cdot \Phi(d_c, d_n) & \text{if } ((d_n, p_n), r) \in R \\ p_c \cdot p_n \cdot \Phi(d_c, d_n) & \text{otherwise} \end{cases} \quad (13)$$

ベクトルの場合には、類似度を各要素の印象度が反映するよう(9)式を次のように拡張する。

$$\Phi(x, y) = \frac{1}{n} \sum_{i=1}^n \left(1 - \sqrt{p_x^i \cdot p_y^i} \cdot \frac{|d_x^i - d_y^i|}{d_{max} - d_{min}} \right) \quad (14)$$

ここで、 (d_x^i, p_x^i) は、 x の i 番目の要素、 (d_y^i, p_y^i) は、 y の i 番目の要素である。(13)式は想起されたデータ1つ1つについて逐次的に処理するときの r_t の計算手続きを示しているが、複数の相異なる根拠が同一の要素を支持するときには、結果的にそれらの複数の印象度を合成することになる。2つの印象度 x, y を合成する関数 F_2 は(13)式から次のように定義される。

$$F_2(x, y) \stackrel{\text{def}}{=} x + y - xy$$

これを任意個の印象度からなる集合 $X = \{x_1, x_2, \dots, x_n\}$ を引数にとる関数に拡張すると、次のような X の要素に關する基本対称式の整式となる。

$$\begin{aligned} F(X) &\stackrel{\text{def}}{=} F_2(x_1, F_2(x_2, \dots, F_2(x_{n-1}, x_n) \dots)) \\ &= x_1 + x_2 + \dots + x_n \\ &\quad - x_1 x_2 - x_1 x_3 - \dots - x_1 x_n - x_2 x_3 - \dots \\ &\quad \dots - x_{n-1} x_n \\ &\quad + x_1 x_2 x_3 + \dots + x_{n-2} x_{n-1} x_n \\ &\quad \vdots \\ &\quad + (-1)^{n-1} x_1 x_2 \dots x_n \\ &= \sum_{i=1}^n \left\{ (-1)^{i-1} \sum_{S \subseteq X \setminus \{i\}} \left(\prod_{x \in S} x \right) \right\} \quad (15) \end{aligned}$$

この関数は、引数の要素について対称であるため、その値は計算順序によらない。また、引数の全ての要素の値が(0,1)

の範囲にあれば、値も(0,1)に入り、かつ値はどの引数要素よりも大きくなる。

(4)式の定義により R に含まれる要素の個数が N_R 個に制限されているため、上の式で計算した r_t を含む2項組 $((d_n, p_n), r_t)$ および変更を受けなかった R の要素の中から印象度の高い方から N_R 個が新たな R の要素となる。

逐次実行型の計算機では、全ての記憶データについて類似度を計算していたのではノード数に比例した計算時間を要することになる。小さな計算コストで類似性の高いものを優先的に検索する手続きが用意できれば問題はないが、一般にはそのような手続きは存在しない。但し、データ間に全順序関係が成り立つ場合は、順序に従った何段階かの分類木を作っておく方法がある[2]。

3.5 連想に基づく想起表の更新

想起表中のデータから更に記憶データ中のリンクを辿って連想されるデータで想起表の内容を置き換える。このとき R 中の新たな印象度は元の値とリンクの印象度との相乗平均を取る。想起の場合と同様に、あるノードに対応する R 中の要素は1つのみとし、同一のノードに対して複数の連想がある場合は、(15)式で定義された関数を用いて合成した印象度を与える。これらの要素の内、印象度の高い方から N_R 個の要素だけを残し、それ以外は想起表から削除する。新たな想起表の要素の候補となるノードの集合 Q は次のように表現できる。

$$Q = \{ \langle n_q, r_q \rangle \mid r_q = F(\{ \sqrt{r \cdot p_l} \mid (n, r) \in R, (n, n_q, p_l) \in L \}) \} \quad (16)$$

3.6 記憶の強化と減衰

M および R 中の全ての印象度を各ステップ毎に減衰させる。ただし、ノード N については R の要素に含まれるノードの印象度を強化する。具体的には次のように計算する。

$$p_t = \begin{cases} p_{t-1} + \eta \cdot (1 - p_{t-1}) & \text{if } ((d, p_{t-1}), r) \in R \\ \eta \cdot p_{t-1} & \text{otherwise} \end{cases} \quad (17)$$

ここで、 η は $\eta \in (0, 1)$ なる定数である。さらにリンクについては3.7節で述べる“有用な予測”に使用されたものについても印象度を強化する。この減衰と強化によって、より有用と思われるデータを記憶に残すことができる。

R に含まれるノードとは、すなわち過去の経験の中で、現在の状況に似ていると思われるものに対応する。このようなノードに対する強化は、後述する予測に伴う強化と共に、謂わば“リハーサルに基づく記憶の強化”と名付けるべきものである。

3.7 予想に基づく判断

想起表に載っているすべての運動命令から連想を繰り返して、目的を満足する感覚を連想できれば正の評価を、できなければ負の評価を与え、相対的に評価の高い運動命令を実行する。目的は感覚データを領域とし、実数を値域とする関数 $f: D \rightarrow R$ の形で与える。

3.3節で述べたとおり、一般に1つのノードから多数のノードへのリンクが存在するため、盲目的な連想の繰り返しは組み合わせの爆発を起こすことになり、適当な時間内に十分な時間間隔の予想を行なうことが困難となる。これを避ける方法としてアジェンダ制御によるヒューリスティック探索の手法[6]を応用する。アジェンダ制御とは、複数の実行可能なタスクを要素とするアジェンダ・リストと呼ばれるリストを用意し、優先順位の高いタスクに多くの計算資源を割り当てながら疑似並列的に実行する方式である。問題解決における探索とは異なり目標状態は存在せず、適当な時間あるいは走査ノード数を経たところで探索を打ち切る。

想起表中の全ての運動命令ノード $n_i = \langle a_i, p_i \rangle$ についてタスクを生成する。各タスクは状態として (i) 探索開始時点の運動命令 a , (ii) 現在注目しているノード n , (iii) 優先度 p , (iv) 評価点 v の4項目の属性をもつ。初期状態での値はそれぞれ, (i) $a = a_i$, (ii) $n = n_i$, (iii) $p = 1$ とする。 (iv) v の初期値は評価の戦略に依存する。 a 以外の値は探索の進行にともなって変化する。

各ノードの評価点はノードに記録されたデータの評価と印象度の積とする。すなわち、評価関数 f の領域を拡張し、ノード集合 N を含むものとする、

$$f(n) = f(\langle d_n, p_n \rangle) = p_n \cdot f(d_n) \quad (18)$$

となる。評価点 v は連想によってノードを訪れる毎に更新されるが、その具体的方法は最終的に行為を選択するための評価戦略に依存する。探索の結果、個々の運動命令について複数のタスクから評価点を得られるが、これらを総合し、その運動命令の評価を決める要素として次のようなものが挙げられる。(i) 評価点の総和 $\sum f(n_i)$ 。(ii) 評価点の最大値 $\max f(n_i)$ 。冒険的、リスクが大きい。(iii) 評価点の最小値 $\min f(n_i)$ 。保守的。大きな目的関数値を与える感覚にたどり着けない。(iv) 評価点の分散 σ_f^2 。値が小さい方が確実な予想とである。(i) および (ii) は値が大きいほど、(iii) および (iv) は値が小さいほど評価が高くなる。モデルの一般性を保つため、これらの評価点の加重和を取り総合的判断を行なうこととする。この重みの選択によりシステムの性格が与えられる。これら4つの評価要素に基づく総合評価点 v_d の計算式は次のように書ける。

$$v_d = w_1 \{\sum f(n_i)\} + w_2 \{\max f(n_i)\} - w_3 \{\min f(n_i)\} - w_4 \sigma_f \quad (19)$$

ここで w_1, w_2, w_3, w_4 は非負実数の定数である。

タスクの優先度 p には1段の連想を行なう毎に、連想に使われたリンクの印象度との相乗平均 $\sqrt{p \cdot p_i}$ を値として与える。また、複数のノードから1つのノードへ繋がるリンクも存在するため探索が重複する可能性があるが、同じノードから出発し途中で分れた連想経路が再び合流した場合は優先度および評価点を(15)式の関数 F で合成したタスクを1つにまとめる。

判断過程の最終段階では、排反する運動命令同士の総合評価を比較し、最高点のデータを最終的に出力すべき運動命令として決定する。決定された命令は出力されると同時に3.3節で述べた方法により記録されるが、その初期印象度 p_a は、それと排反するデータについての評価点の差に関するモー

メントの総和に反比例した値を与える。モーメントの総和が大きい程、排反する命令の間の優劣が明確であるということになる。具体的には(2)式の印象度の値に対する制約 $p \in [0, 1]$ を満足させるために次のような計算式を用いる。

$$p_a = \frac{1}{v_{max}} \sum_{i=1}^n (v_{da} - v_{di}) \cdot (1 - \Phi(d_a, d_i)) \quad (20)$$

ここで、 d_a は選択されたデータ、 d_i ($i = 1, 2, \dots, n$) は d_a と排反するデータ、 v_d は d に対する総合評価点、 v_{max} はその時点までのモーメント和の最大値である。 p_a が閾値 ν より小さい場合にはランダムにデータを選択することで、一種の冒険を試みる。

予想のための連想に用いられたリンクのうち、運動出力の評価に役立ったものについては、評価の絶対値に応じてその印象度を強化する。

4 試験例題への応用

4.1 設定項目の仕様

試験例題への応用に当って設定された項目を整理すると次のようになる。

世界の構造 物体の配置および大きさは、おおよそ図1のとおりである。表示画面における上下の境界線および左右の境界線はそれぞれ接続しており、世界はドーナツの表面のような有限の面積をもつ循環的な構造とし、その広さは縦方向400単位、横方向640単位とした。餌および糞の半径は10単位である。

入出力データ 3種の感覚入力と2種の運動出力を採用する。

(i) 視覚：データは32個の整数を要素とするベクトルで、要素の値の範囲は $[0, 9]$ である。視野は角度 180° 、半径150単位。

(ii) 嗅覚：虫からの距離が130単位以内にある物体について計算する。 $(-1, 1)$ の範囲にある実数値を取る。

(iii) 触覚(味覚)：1, 0, -1のいずれかを値とする。

入力データ値の計算方法は2.1節のとおりである。(iv) 歩幅：1, 2, 3のいずれかを値とする。

(v) 方向変換量：-2, -1, 0, 1, 2のいずれかを値とする。負の場合は左回転、正の場合は右回転、0は変化なしである。単位回転角は 15° とした。類似度 $\Phi(x, y)$ の計算には(7)式を用いた。

判断過程 判断に関連する項目は次の3点である。

(i) 目的となる評価関数 f ：触覚(味覚)についてだけ定義し、その値は引数となるデータそのものとした。すなわち、餌に当れば評価は1、糞に当れば評価は-1、何もなければ0となる。

(ii) 予想探索の終了条件：探索ノード数が500を越えたところで打ち切る。

(iii) 判断における各評価要素の重み：評価点の総和のみを用いた。すなわち、 $w_1 = 1, w_2 = w_3 = w_4 = 0$ とした。

記憶容量 ノード集合 N : データの種類毎に容量を設定した。リンク集合 L , 想起表 R の容量を合せ以下の表のとおりである。

視覚	嗅覚	触覚(味覚)	歩幅	方向変換量
300	150	50	300	300

リンク	想起表
1000	25

いくつかの係数 関連する係数の値は次のとおりである。

$$\begin{array}{cccccc} \alpha & \beta & \lambda & \delta & \zeta & \eta & \nu \\ 0.2 & 0.2 & 0.33 & 0.5 & 0.4 & 0.95 & 0.05 \end{array}$$

4.2 実験

図4, 5, 6 に1つの実験結果を示す。それぞれ, 200ステップ, 400ステップ, 600ステップの実行後の足跡を実線で示したものである。線の太さは決定された運動命令の強さを表わしている。

図7, 8, 9 はそれぞれの時点での記憶の状態を表示したものである。数字はステップの番号で, 小さい方が古い。数字の下の5つのノードは上から視覚, 嗅覚, 触覚(味覚), 歩幅, 方向変換量の記憶ノードであり, 大きさを強さを表現している。それらのノードを結ぶ実線はリンクであり, 同様に太さで強さを表現してある。200ステップまででは, 記憶の中で際だった系列は現れてきていないが, 400ステップでは, いくつかの特徴的と思われる系列が獲得されていることがわかる。これらの系列のほとんどは600ステップの後にも生き残っている。

4.3 実験を通してのモデルの問題点

実験自体の困難さ, およびモデル設計の困難さの原因は, ほとんど本メカニズムの複雑性に由来するものである。

ここで提案したメカニズムは強さの調整を基本とするものであるが, メカニズムの複雑さのために, ニューラルネットの研究[7]で行なわれているような, システムの挙動に関する数学的解析はほとんど困難である。元来, 完結した個体のモデルとして考案したものであるため, このような複雑性は不可避であり, 今のところ実験による検証以外にシステムの性質を調査する術がない。想起, 記憶, 判断といったサブシステムについて何らかの解析を試みるということから出発すべきであろう。

モデル自身のパラメータの多さも, 実験を困難なものにしている。ちょっとしたパラメータの変更はその他の要因, 特に環境の特性と複雑に絡み合って, システムの挙動に微妙な影響を与える。1つ1つのパラメータについて適応的, つまりシステム自身が値を調整するように変更できないか検討する必要がある。

5 おわりに

ここで提案したモデルは, 記憶の構造としてネットワークを基本としながらも, 必ずしも神経回路網の形態, つまり関

素子と活性化伝播にこだわるつもりはない。しかし, モデルを知能のアーキテクチャとして捕え直すならば, 代数的モデル表現に基づく計算機シミュレーションだけではなく, 自律した演算素子をノードとするネットワークによる計算モデルを考える必要がある。

あるいは, 分散協調問題解決の枠組み[8]で捉え直すことも有用と思われる。図3における感覚入力, 想起, 判断, 行為出力といった機能単位, および記憶と想起表中の個々の要素をエージェントと考えることができる。

また, ここでは基本的には時間的な前後関係による連想のみを用いており, データは具体的レベルのみであって記憶の抽象化は一切行なわれていない。暗記学習の枠からはみ出すことにはなるが, より高度な機能を実現するにはデータの抽象化[9]が必要となろう。

参考文献

- [1] 畷見達夫: 学習機能を中心とする基礎的認知行動モデル, 日本認知科学会第1回大会・発表論文集, A-2, (1984)
- [2] 畷見達夫: 離散時系列の暗記学習, 計測自動制御学会第8回知識工学シンポジウム, pp. 47-50, (1988)
- [3] Cohen, P. R. and E. A. Feigenbaum (Eds.): *The Handbook of Artificial Intelligence*, Vol. III, Chapter XIV, B.2, William Kaufmann, (1982)
- [4] Michalski, R. S.: *A Theory and Methodology of Inductive Learning*, in R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds.), *Machine Learning - An Artificial Intelligence Approach*, Springer-Verlag, (1984)
- [5] Laird, J. E, P. S. Rosenbloom and A. Newell: *Chunking in Soar: The Anatomy of a General Learning Mechanism*, *Machine Learning*, Vol. 1, pp. 11-46, (1986)
- [6] 畷見達夫: アジェンダ・コントロールに基づく並行実行 *Prolog*, 情報処理学会研究報告 87-AI-54-3, (1987)
- [7] 麻生英樹: ニューラルネットワーク情報処理-コネクショニズム入門、あるいは柔らかな記号に向けて-, 産業図書, (1988)
- [8] 小野典彦: 分散協調問題解決, 淵一博監修, 古川・溝口共編, 知識プログラミング, 共立出版, (1988)
- [9] 畷見達夫: 抽象化リンクによる信号と記号の融合, 日本認知科学会第5回大会・発表論文集, B-1, (1988)

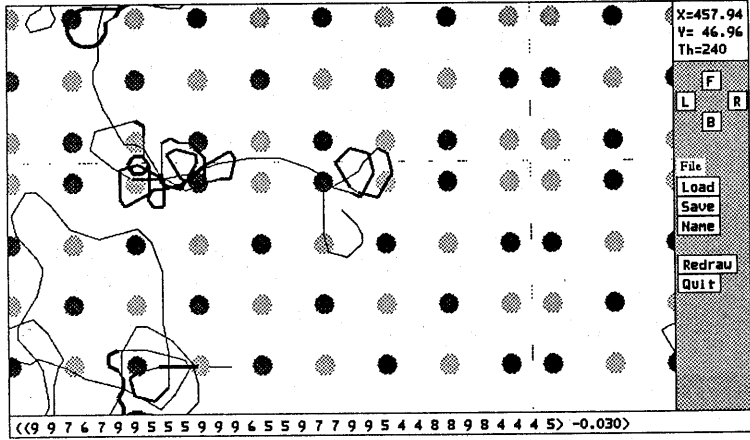


图 4: 实验结果: 足跡-I

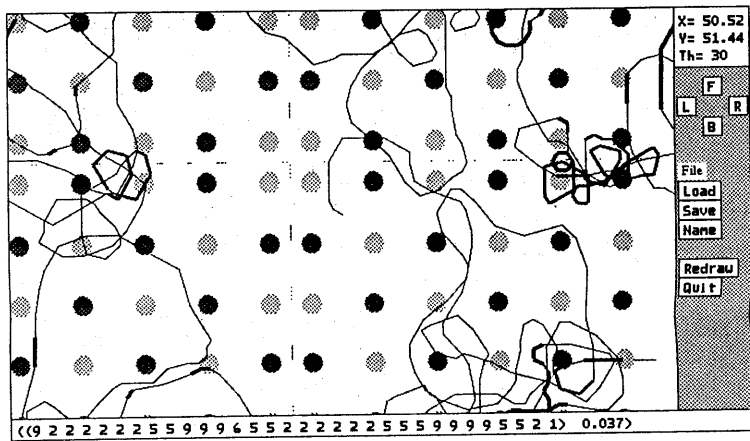


图 5: 实验结果: 足跡-II

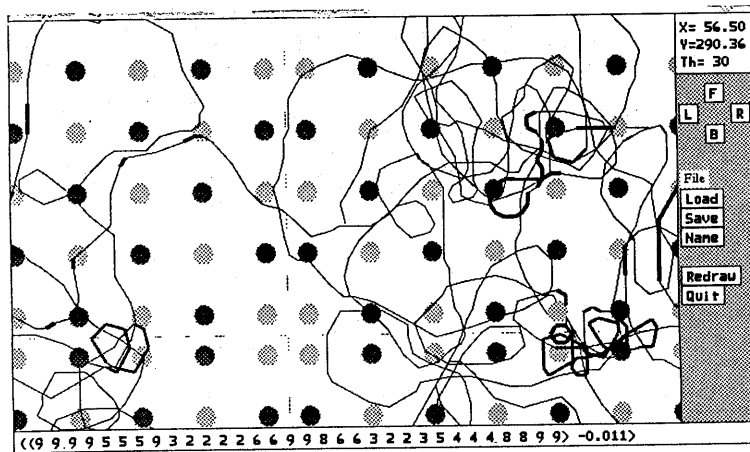


图 6: 实验结果: 足跡-III

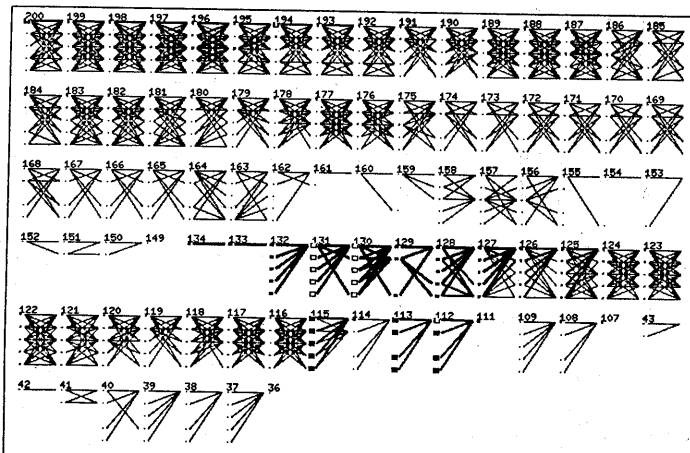


図 7: 実験結果: 記憶-I

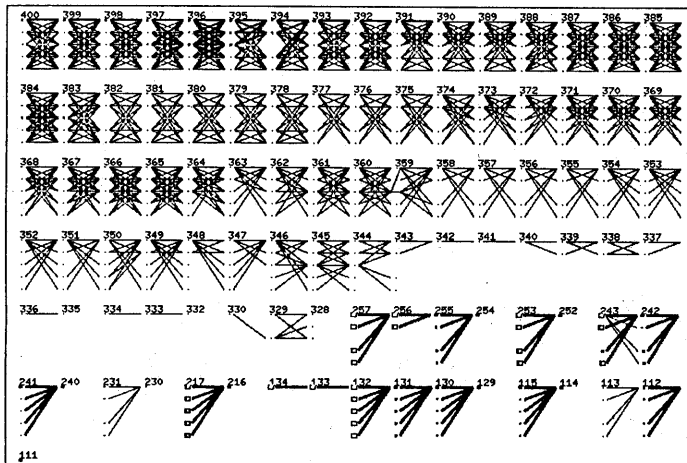


図 8: 実験結果: 記憶-II

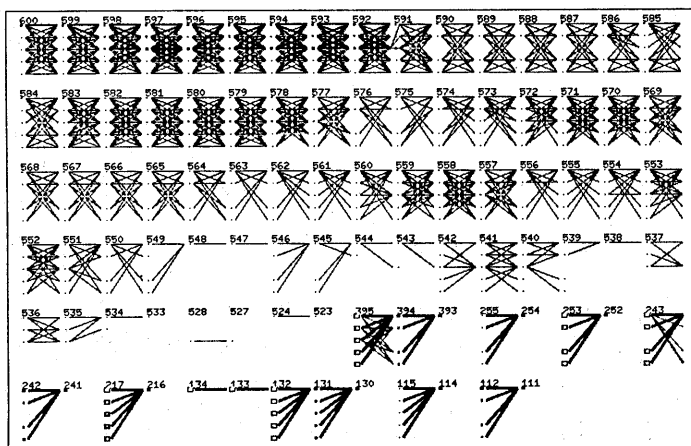


図 9: 実験結果: 記憶-III