法政大学学術機関リポジトリ
HOSEI UNIVERSITY REPOSITORY

PDF issue: 2025-05-09

連想記憶モデルBAMの記憶パターンに関する検討

HIROTA, Kaoru / SUZUKI, Kazuya / 鈴木, 一也 / 廣田, 薫

(出版者 / Publisher)
法政大学工学部
(雑誌名 / Journal or Publication Title)
法政大学工学部研究集報 / 法政大学工学部研究集報
(巻 / Volume)
27
(開始ページ / Start Page)
13
(終了ページ / End Page)
23
(発行年 / Year)
1991-02
(URL)
https://doi.org/10.15002/00003906

連想記憶モデルBAMの
記憶パターンに関する検討

廣田　薫*・鈴木一也*

On The Memory Pattern for Bidirectional Associative Memories BAM

Kaoru HIROTA* and Kazuya SUZUKI*

Abstract

Bidirectional Associative Memories BAM proposed by B.Kosko have been investigated from viewpoints of memory pattern conversion, two layer, and pattern division. It become spossible to increase the number of memory patterns, and the condition for memory pattern selection is improved. Finally one to many correspondence of BAM is discussed toward new applicational directions.

§1.　はじめに

（従来の）コンピュータに使用されているメモリー回路では、二進法により直列的に記憶させる方式がとられており、機械的な学習方法でのみ情報を理解し、二つの値を比べる場合でもそれが完全にマッチしているか否かを判断するのみであった。いわゆる"類似"という状態は認識することができない。そのため、コンピュータは高速な直列演算は得意であるが、リアルタイムパターン識別等は苦手としている。しかし、人間はその逆であり、リアルタイムパターン識別は得意であるが、高速での直列演算は苦手である。このような性質をもつ人間の頭脳が行うパターン解析は、現在、連想メモリーをもとにして行われており、それをシミュレートするのがB.Kosko による BAM (Bidirectional Associative Memories) 技術[1] である。

BAM の性質として、記憶パターンの総数が過剰になった場合に分類の精度は落ちることがわかっている。パターンを連想する速度、または連想が可能であるか否かは、パターンの総数と常に深いかかわりがあり、パターンの総数が増えるに従って連想の能力、速度は低下してしまう。

ここでは様々な角度から記憶パターン総数の増加、パターン選定における条件の改善策について検討した結果を報告すると共に、BAMを連想記憶以外に応用する可能性についても報告する。

＊工学部電気工学科計測制御専攻

§2.　BAMの概要

2.1　BAMの関係式

BAMは以下のように定義される[1]。

記憶させるパターンとして式（2－1）を与える。ここで各X, Yは式（2－2），（2－3）のような制約をもつ。この記憶パターンを用いて相関行列Mを式（2－4）により求め、このMによって連想を行う。各X, Y の各要素x_i, y_jを式（2－5），（2－6）のように計算する。

一般に、Fig.1 に示すように双方向の想起を繰り返すことにより連想を行う。このとき、BAMはエネルギー関数式（2－7）を極小にする挙動を示す事がわかっている。

$$(X_1, Y_1),\ (X_2, Y_2), \cdots,\ (X_m, Y_m) \tag{2-1}$$

$$X \in \{0, 1\}^n \quad Y \in \{0, 1\}^p \tag{2-2}$$

$$m < \min\ (n, p) \tag{2-3}$$

$$M = X^T Y \tag{2-4}$$

$$XM_j = \sum_i X_i m_{ij}\ ,\ YM_i^T = \sum_j y_i m_{ij} \tag{2-5}$$

$$X_i = \begin{cases} 1, \text{ if } YM_i^T \geq 0 \\ 0, \text{ if } YM_i^T < 0 \end{cases}$$

$$Y_i = \begin{cases} 1, \text{ if } XM_i \geq 0 \\ 0, \text{ if } XM_i < 0 \end{cases} \tag{2-6}$$

$$E = -XMY^T \tag{2-7}$$

2.2　BAMの実際の挙動

実際には、Fig.2 に示すように、ノイズのある入力に対しても記憶したパターンを連想する事ができる。

§3.　方向性検出への適用の試み

BAMを方向性検出に用いるために、Fig .3 に示すように、X側が8×8の12パターン、それに対してY側が4×6の12パターンとし、データの縮約を目的としたパターンをBAMに記憶させた。重み行列Mに対してX側の12パターンを1つずつ入力したところ、全てFig.4に示すようなパターンに収束してしまい、うまく連想することが出来なかった。

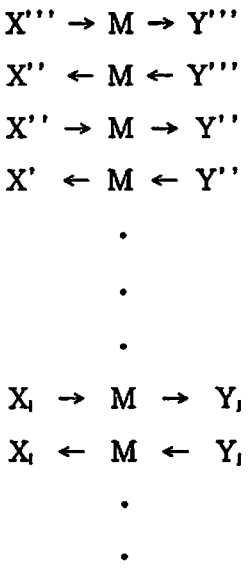

Fig.1　BAM の双方向想起

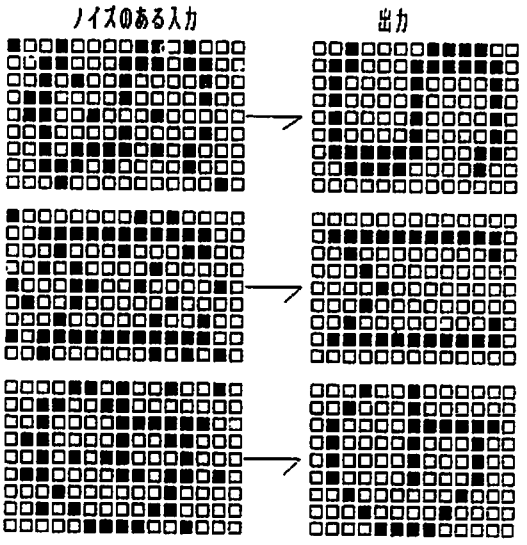

Fig.2　BAMの挙動の一例

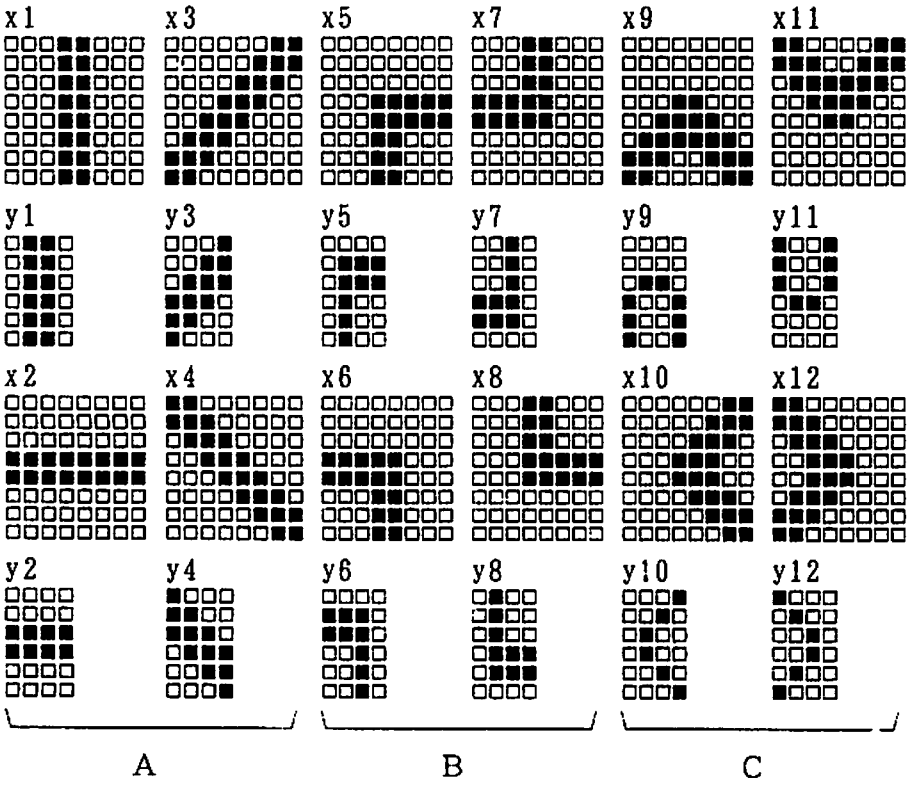

Fig.3　12パターン

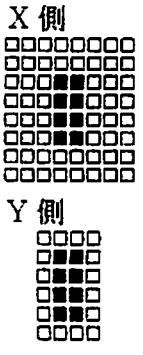

Fig.4　誤動作の出力

§4.　記憶パターンの変換

連想できない原因について考えてみると、Y側の（4×6＝24）bitに対して12パターンの12という数は式（2－3）の制約を満たしているが、実際問題として連想できていないので（4×6＝24）bit を増やすことを考えた。そこでまず記憶パターンを変換することにした。例えば 010101　→ 001100110011 のように変換すれば（8×6＝48）bit となる。しかしこのような変換をする場合は、入力にノイズの入ったパターンも同様の変換をしなければならないため結果として4×6のままで連想させた時の結果にこの変換を施したものは同じになってしまい、うまく行かなかった。記憶パターンに変換を施すということは、入力パターンにも変換を施さなければならないので、この方法ではうまく行かない。

§5.　ハミング距離の概念

BAMの特性を考えた場合にハミング距離という考え方が重要になってくる。
このハミング距離について説明する。
いま、パターンを式（5－1）のように定義したとき

$$A1 = (1,1,0,0,1,1,0,0)$$
$$A2 = (1,1,1,1,1,1,1,1) \qquad (5-1)$$

A1, A2のハミング距離は次のようなハミング関数式（5－2）,（5－3）で計算される。

$$H\ (A1,\ A2) = \sum_{i=0}^{n} f\ (a1_i,\ a2_i) = 4 \qquad (5-2)$$

$$f\ (a1_i,\ a2_i) = \begin{cases} 1\ ,\ \text{if}\ a1_i \neq a2_i \\ 0\ ,\ \text{if}\ a1_i = a2_i \end{cases} \qquad (5-3)$$

§6.　BAMの2層化

連想がうまく行われる時には、入力パターンにハミング距離が最も近い記憶パターンが想起されるので、記憶させるパターンを選定する場合にはハミング距離が遠くはなれている方が、より連想しやすいのではないかと予測できる。そこで、うまく連想することの出来ない12パターンを2層に分けてみることにした。2層に分けるということは、Fig.5のように、いままでX側とY側しかなかったものの間に、中間層を入れることである。この中間層にできるだけハミング距離の離れたパターンを記憶させることによって、X側, Y側のパターンには変換などの操作を加えることなく、記憶させたいパターンをそのままにしておくことが出来る。この2層化を連想できなか

った12パターンに対して行ったところ、2層化しない場合に比べて良い結果が出たが、12パターン全ての入力に対応させることは出来なかった。パターン数が多いことが原因であると考えられる。

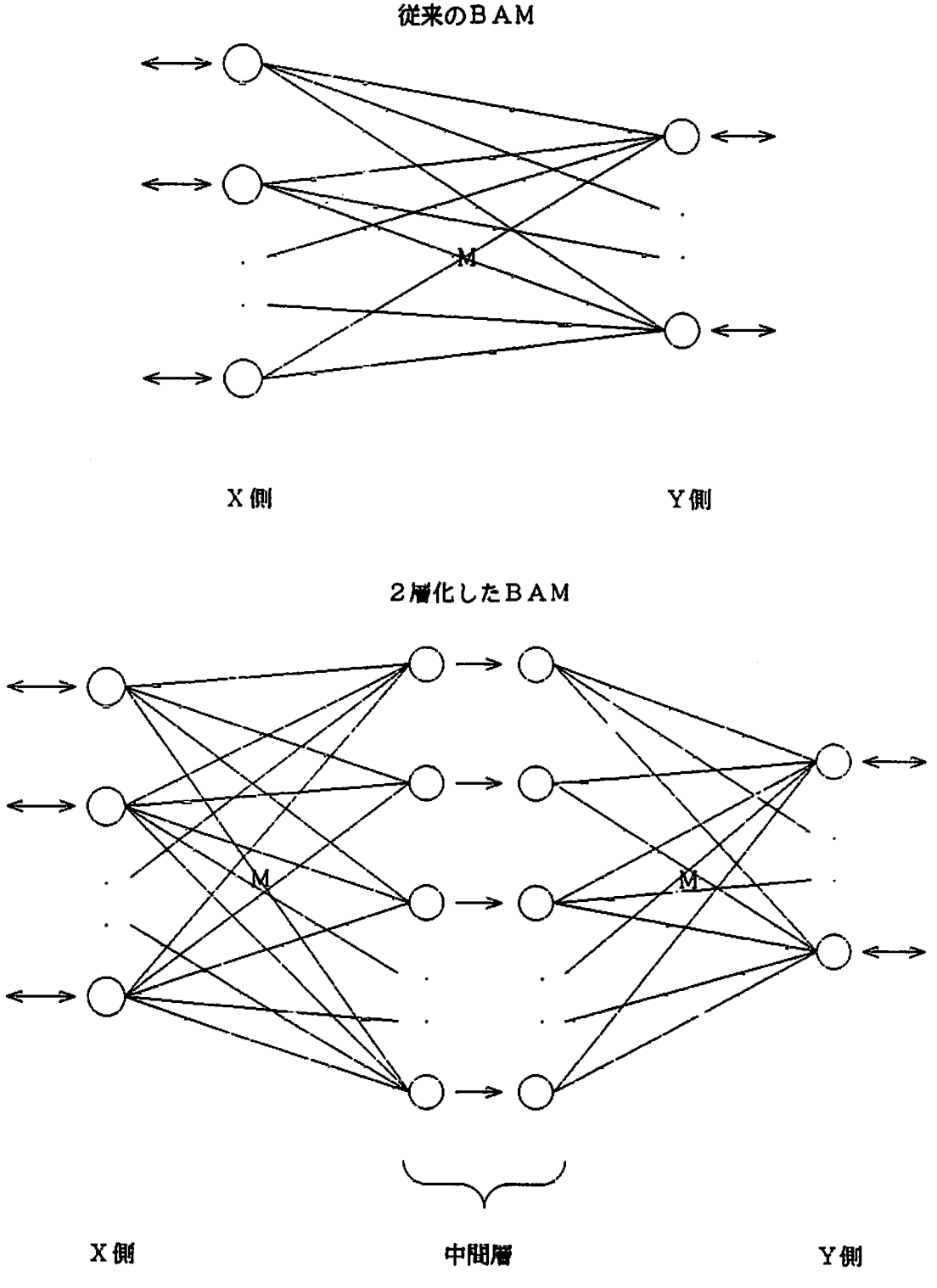

Fig.5　BAMの2層化

§7.　記憶パターンの分割

12パターンで連想がうまく行かないのであれば12パターンを分割することでその連想を可能に出来るのではないかと考えテストを行った。はじめはFig.3のA, B, Cのように分割したパターンについて記憶させ連想させた。その結果はFig.6のようにy側の5, 6, 7, 8を除いてうまく連想されている。

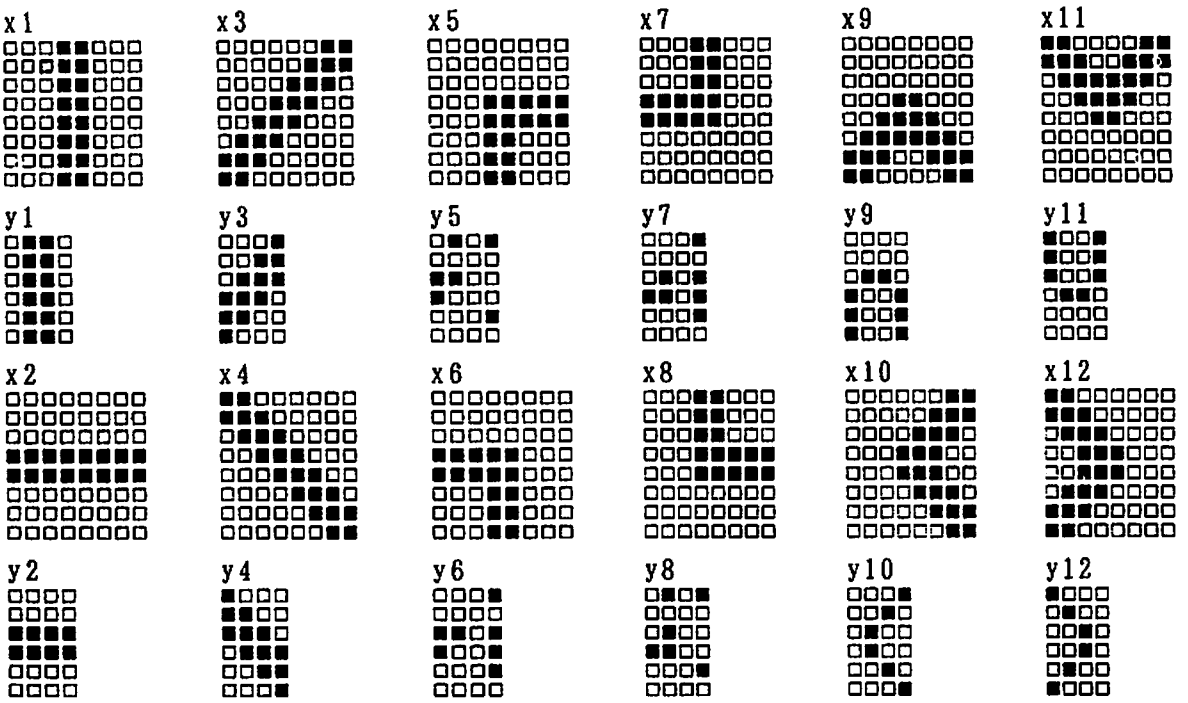

Fig.6　3分割での出力

このような3分割は、どの様に分割してもうまくいく訳ではない。そこで、ハミング距離を用いた効果的なパターン分割方法を新たに提案する。まずハミング距離を、12パターンで考えられる全ての組合せについて、Table1.1,1.2 に示すように計算する。そしてこれを用いてハミング距離の近いパターンが別のカテゴリになるようにクラスタリングを行う。今回の場合bit数の少ないTable1. 2に示した y側のハミング距離を用いて分割を行った。

一番近いハミング距離を探すとy1に対するy5, y6, y7, y8であるので、まず分割する。次に近いのは、y2に対するy3, y4, y9, y10であるのでそれを分割する。ここで、y1とy2を比べると比較的ハミング距離は遠いので同じグループにする。y10, y12に関しても同様に同じグループにする。このようにして分割するとFig.7に示すようになり、これをそれぞれBAMに記憶させるとうまく連想させることが出来た。しかし同様の方法を2分割に対しても適用して実験を行ったが、これでは良い結果を得られなかった。

Table1.1 X側パターンハミング距離

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
X1	0	24	26	26	12	12	12	12	26	26	26	26
X2		0	26	26	12	12	12	12	26	26	26	26
X3			0	36	26	26	26	26	18	18	18	18
X4				0	26	26	26	26	18	18	18	18
X5					0	12	24	12	24	24	28	28
X6						0	12	24	24	28	28	24
X7							0	12	28	28	24	24
X8								0	28	24	24	28
X9									0	18	36	18
X10										0	18	36
X11											0	18
X12												0

Table1.2 Y側パターンハミング距離

	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12
Y1	0	12	12	12	7	7	7	7	16	10	16	10
Y2		0	8	8	9	9	9	9	8	10	8	10
Y3			0	16	7	13	7	13	10	10	10	10
Y4				0	13	7	13	7	10	10	10	10
Y5					0	10	10	10	13	9	11	9
Y6						0	10	10	13	9	11	9
Y7							0	10	11	9	13	9
Y8								0	11	9	13	9
Y9									0	10	16	10
Y10										0	10	12
Y11											0	10
Y12												0

この方法を用いると、パターンを分割する場合に1つの入力に対して3つのBAM（重み行列）に入力しなければならないことや、それに伴って出力も3つ出てきてしまうことになるが、入力、記憶、出力のそれぞれのパターンを比べれば、この3つ出力のどれを選択すれば良いのかは自ずとわかるはずである。

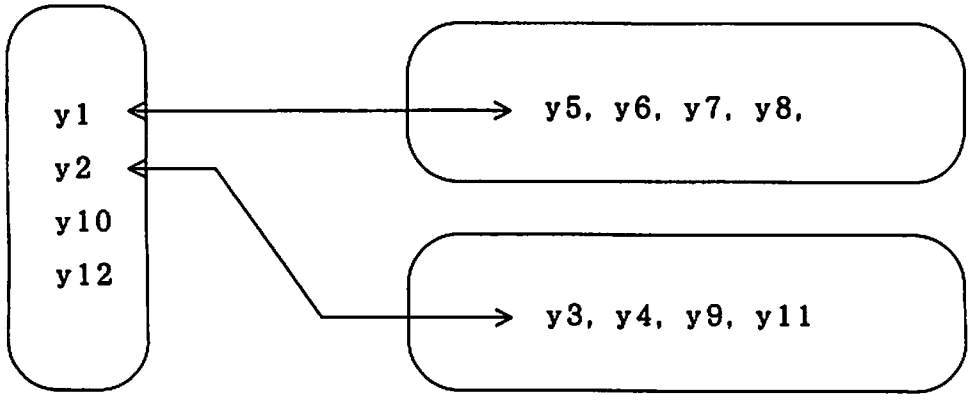

Fig.7　ハミング距離を用いた分割

§8.　分割　プラス　2層化

§7のパターン分割の話では3分割の場合には良好な結果を得たが、2分割ではうまくいかなかった。また、2層化の話では12パターンの連想は完全にはうまくいかなかった。そこでこの2つを重ね合わせて実験をすることにした。

つまり、2分割したものを2層化するのである。この実験で、はじめは中間層に用いる記憶パターンのbit数をX側の（8×8＝64）と同じ数にして行ったが結果は良くなかった、そこで自由に設定できる中間層のbit数を増やしてみることにした。（24×24＝576）のbit数で実験したところ良い結果が得られた。この分割＋2層化の方法では、取り扱い可能なパターン数の増加とは別にもう1つの利点が挙げられる。それは中間層を用いたためにX側とY側とで別々の分割が出来ることである。例えばTable2 のように、Y側でY1とY2がハミング距離が遠くY1とY3は近いときに、X側ではその逆にX1とX2が近くX1とX3が遠くにある場合を考えよう。

Table2　ハミング距離のX側とY側でのズレ

	X1	X2	X3
X1	0	10	100
X2		0	100
X3			0

	Y1	Y2	Y3
Y1	0	100	10
Y2		0	100
Y3			0

このような場合であっても間に中間層があることで別々のクラスタリングが可能となるのである。ハミング距離によるX側とY側の同一のクラスタイリングでうまくいかない場合には有効であると考えられる。

§9.　一対多への応用

9.1　一対多対応への応用

BAMは連想記憶モデルであるので、通常、記憶パターンは一対一で対応させるのであるが、一対多の対応で記憶パターンを設定した場合にはどの様な挙動を示すかテストを行った。Fig.8 のようなパターンを記憶パターンとして用いX側の記憶"1"、"2" に対して縦縞を"3"、"4" に対して横縞を対応させ記憶させた。それに対して、1、2、3、4をそれぞれ入力したところFig.9のような出力が得られた。X側には1と2、3と4が重なったパターンがそれぞれ出力されているが、Y側では良い結果が得られた。

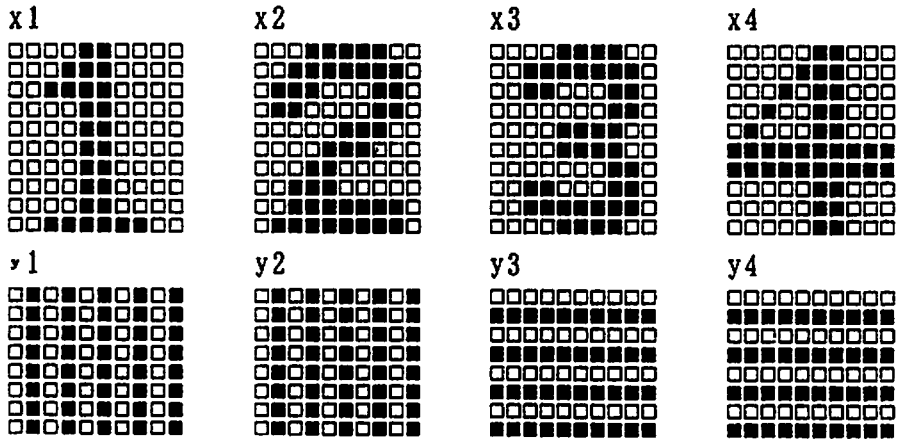

Fig.8　一対多対応の記憶パターン

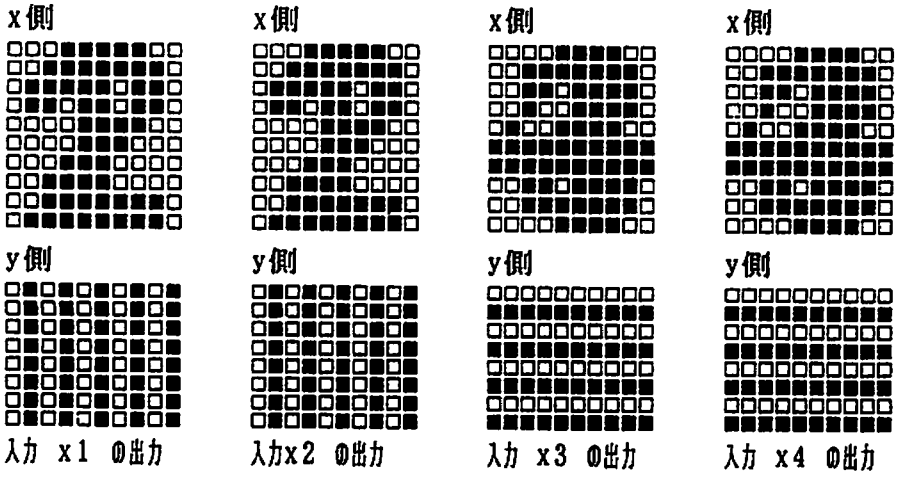

Fig.9　一対多出力パターン

以上の結果から本来のBAMの機能である連想記憶とは別の方向の利用が考えられる。例えばイカとタコに対して軟体類、カニ、エビに対して甲殻類というように記憶させたとき、イカに似た入力に対して軟体類が、またカニに似た入力に対しては甲殻類を出力するといった類の応用が考えられる。

9.2　応用例

実際にはY側の軟体類に対してFig.8のY側のような横縞を、甲殻類に対しては縦縞を対応させ、記憶させる。

X側にはFig.10のような特徴によって式（9－1）のようにパターンを決める。このようにBAMに記憶させ実験したところ良好な結果を得ることが出来た。

	1	0
ハサミ	ある	ない
皮膚	硬い	柔らかい
しっぽ	ある	ない
色	赤い	白い

$$1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad 0 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

Fig.10　動物の特長と実際のパターン

$$\begin{array}{lcccccc} & & \text{ハサミ} & \text{皮膚} & \text{しっぽ} & \text{色} & \\ \text{イカ} = & (& 0 & 0 & 0 & 0 &) \\ \text{タコ} = & (& 0 & 0 & 0 & 1 &) \\ \text{カニ} = & (& 1 & 1 & 0 & 1 &) \\ \text{エビ} = & (& 0 & 1 & 1 & 1 &) \end{array} \qquad (9-1)$$

§10.　まとめ

連想記憶モデルBAMの記憶パターンについて、そのパターン選定の条件を改善するために、パターンの変換、BAMの二層化、パターンの分割を提案し、その有効性について検討を行った。

また、連想記憶の一対多対応への応用について提案し、その実例を紹介した。

参考文献

1) Bart Kosko, Bidirectional Associative Memories, IEEE, Transactions On Systems, Man, And Cybernetics, Vol 18, No.1, PP.49／60, JANUARY／FEBRUARY 1988

2) 今崎直樹・山口亨, ファジィニューラルネットに向けて　－連想記憶とファジィ－ Computer Today, pp.52／58, 1990／1