分散記憶方式とその性質*

大阪府立大学 工学部　福 永 邦 雄
同　上　笠 井 保

1. まえがき

最近の計算機技術の発展に伴ない、計算機の処理能力は飛躍的に増大し、その応用範囲も益々拡がってきている。しかし、一方では新しい応用分野が現われるに従って、従来のアーキテクチャでは処理能力を本質的に向上できない分野があることも指摘され、いわゆる非ノイマン計算機について検討され始めている。[(1)] 本稿ではこのうち、ノイマン式計算機の基本をなす番地式記憶方式に代る一つの試みとして無番地方式の分散記憶方式を提案し、この方式の有する基本的な性質を述べている。従来の計算機の記憶においては記憶事項をアドレスを介してある特定の領域に書込み、又、読出するため、記憶情報を取扱う際にはその内容よりもアドレスが重要な役割を演ずる。これに対し、最近アドレス操作に制限されることなく、記憶情報を取扱いやすい形で記憶装置全体に分散させて記憶する方法が考えられており、記憶形態と書込み、読出し操作を適当に選べば、種々の機能をもった記憶装置が構成できることが示されている。アソシアトロン[(1)]及びそれに関連する連想記憶装置[(2)〜(6)]はその代表的なものである。本論文では連想機能にとらわれることなく、分散記憶方式を広く一般的な形で定義し、その性質を明らかにしている。その結果、このような記憶方法はアナログ情報を記憶するのに適しているなど、従来のアドレス形式に基づく離散形記憶にない柔軟な性質をもつ記憶方法であることを明らかにしている。まず、2.で情報の書込み、読出し方法を明らかにし、分散記憶方式を定義している。3.では記憶容量と精度について述べ、4.では記憶破壊と読出し誤差について明らかにしている。最後に5.では本方式の特徴をまとめている。

2. 分散記憶方式

2.1 諸定義

n 個の n 次ベクトル系 a_1, a_2, ……, a_n は，各 a_j ($j=1, 2, \cdots\cdots, n$) が

$$a_j = (a_{1j}, a_{2j}, \cdots\cdots, a_{nj})^t \qquad (1)$$

なる列ベクトルで表されるとき，

* 本稿は、電子通信学会論文誌に発表した論文を一部加筆修正して投稿したものです。

$$\boldsymbol{a}_i^t \cdot \boldsymbol{a}_j = \sum_{s=1}^{n} a_{si} \cdot a_{sj} = \begin{cases} 0\ ; & i \neq j \\ n\ ; & i = j \end{cases} \tag{2}$$

の直交性の条件を満足しているとする。行列 A_n はこのベクトル系を用いて

$$A_n = (\boldsymbol{a}_1 ,\ \boldsymbol{a}_2 ,\ \cdots\cdots,\ \boldsymbol{a}_n) \tag{3}$$

で表されている。

次に，j 行 j 列要素が k_j 及び d_j で表される n 次対角行列をそれぞれ K_n 及び D_n とし，j 番目の成分が y_j，r_{sj} 及び b_j で表される列ベクトルをそれぞれ $\boldsymbol{y}$，$\boldsymbol{r}_s$ 及び $\boldsymbol{b}$ とする。

いま変数 x_j に対して

$$\boldsymbol{r}_j = d_j K_n \boldsymbol{a}_j x_j \tag{4}$$

なる演算を行い，ベクトル $\boldsymbol{r}_j$ に変換することを x_j について W_j（K_n，$\boldsymbol{a}_j$，d_j）変換と呼ぶことにする。又，ベクトル $\boldsymbol{y}$ に対して

$$z = u \boldsymbol{b}^t K_n^{-1} \boldsymbol{y} \tag{5}$$

なる演算を行い，スカラ値 z を得ることをベクトル $\boldsymbol{y}$ についての R（u，$\boldsymbol{b}$，K_n）変換を呼ぶことにする。

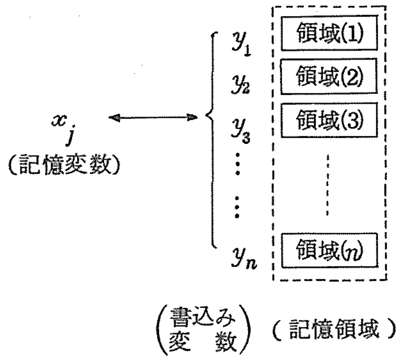

図1　分散記憶方式のモデル

Fig.1－A model of the distributed memory system.

2.2 記憶方法

これらの定義をもとに，分散記憶方式について述べよう。記憶しようとする m 個の情報 $\{ x_j \mid j = 1, 2, \cdots\cdots, m \}$ は各々 α ビットで表されているとする。各 x_j は非数値情報でもよいが，α 個のビット構成パターンを整数と対応させれば，数値として取扱うことができるので，特に断らない限り，x_j は

$$| x_j | \leqq 2^{\alpha - 1},\ (j = 1, 2,\ \cdots\cdots,\ m) \tag{6}$$

の範囲の整数であるとして取扱う。

図1は分散記憶方式のモデルである。記憶装置はn個(但し$n \geqq m$)の記憶領域に分かれており，各領域にはβビットの情報を記憶することができ，s番目の領域にはy_sの情報が記憶されているとする。このy_sはx_jの場合と同様，整数であるとし

$$|y_s| \leqq 2^{\beta-1}, \quad (s = 1, 2, \cdots\cdots, n) \tag{7}$$

の範囲の値であるとする。

〔記憶の書込み〕

いま数値x_j($1 \leqq j \leqq m$)を記憶するとき，x_jに対してW_j(K_n, a_j, d_j)変換を行いベクトルr_jを求め，このr_jを記憶数値yにつけ加えた値

$$y \Leftarrow y + r_j \tag{8}$$

を新たなyとして記憶する。このようにyにr_jをつけ加えることをx_jの書込みと呼ぶ。

〔記憶の消去〕

記憶系にx_jの情報が書込まれているとき，x_jに対してW_j(K_n, a_j, d_j)変換を行ってr_jを求め，記憶数値yからr_jを引いた値

$$y \Leftarrow y - r_j \tag{9}$$

を新たな記憶数値とする。このようにyからr_jを減じることをx_jの消去と呼ぶ。

〔記憶の読出し〕

記憶されている数値yに対してR(u, b, K_n)変換を行い数値zを得ることをR(u, b, K_n)変換による読出しと呼ぶ。読出す内容により，パラメータu及びbをを適当に選ぶ。

これらの操作に基づいて次の定義をしよう。

〔定義1〕　変数x_j($j=1, 2, \cdots\cdots, m$)をW_j(K_n, a_j, d_j)変換($n \geqq m$)により書込み変換を行い$\{y_s \mid s = 1, 2, \cdots\cdots, n\}$の形で記憶したのち，$R$($u$, b, K_n)変換により読出す方式をn次の一般形分散記憶方式(general distributed memory system, gdm方式と略記)と呼ぶ。特に読出し変換を$\{R(1/nd_j, a_j, K_n) \mid j=1, 2, \cdots\cdots, n\}$に選んだとき，$n$次の分散記憶方式(dm方式)と呼ぶことにする。これらの方式において用いられる変数x_j($j=1, 2, \cdots\cdots, m$)を記憶変数，y_s($s=1, 2, \cdots\cdots, n$)を書込み変数と呼び，W_j(K_n, a_j, d_j)変換を書込み演算，R(u, b, K_n)を読出し演算と呼ぶことにする。

〔定義2〕　分散記憶において各記憶領域の容量がβビットに制限されているときのgdm，dm方式をそれぞれ容量βビットのgdm，dm方式と呼ぶ。

〔定理1〕　dm 方式において，記憶容量が十分大きく，記憶情報が失われないとき，x_j ($j=1, 2, \cdots\cdots, m$)が既に記憶されていると，読出し演算をR ($1/nd_j$, $\boldsymbol{a}_j$, K_n)に選べば読出された値はx_j となる。

（証明）　$\boldsymbol{y}$ にはm個の情報 $\{ x_j \mid j=1, 2, \cdots\cdots, m \}$ が既に書込まれているとすると，式(5)から読出された値zは

$$z = \frac{1}{nd_j} \boldsymbol{a}_j^t \circ K_n^{-1} \cdot \left(\boldsymbol{r}_j + \sum_{\substack{i=1 \\ i \neq j}}^{m} \boldsymbol{r}_i \right)$$

$$= x_j + \frac{1}{nd_j} \boldsymbol{a}_j^t \cdot \left(\sum_{\substack{i=1 \\ i \neq j}}^{m} d_i \boldsymbol{a}_i x_i \right) \tag{10}$$

となるが，式(2)の $\boldsymbol{a}_j$ ($j=1, 2, \cdots\cdots, n$) の直交性の条件から，上式の右辺第2項は0となる。（証明終）

dm 方式はパラメータ K_n, D_n 及びA_nの選び方により種々の方式が考えられる。このうち，最も簡単な方式から考えてみよう。

〔定義3〕　dm方式のうち，特にK_n, D_nを単位行列に選び，すなわち $k_j = d_j = 1$ ($j=1, 2, \cdots\cdots, n$)とし，行列A_nの i 行 j 列要素 a_{ij} を

$$a_{ij} = \begin{cases} 0 \ ; & i \neq j \\ \sqrt{n} \ ; & i = j \end{cases} \tag{11}$$

に選んだとき，この方式を単純記憶方式（ s − dm 方式）と呼ぶ。

s − dm 方式は図1のモデルにおいて，x_j の記憶情報を j 番目の領域にまとめて記憶する方法であり，j 番目以外の領域には x_j の情報は含まれていない。これは離散形記憶方式の番地選択操作をベクトル$\boldsymbol{a}_j$ に置換えた方法に相当している。

次に，もう少し一般化した記憶方式を示す。

〔定義4〕　dm方式のうち，行列A_n の要素 a_{ij} を

$$| a_{ij} | = 1, \quad (i, j = 1, 2, \cdots\cdots, n) \tag{12}$$

に選んだとき，この方式を一様分散記憶方式（ u − dm 方式と略記）と呼ぶ。特に，u − dm方式のうち，K_n及びD_nを単位行列に選んだとき，この方式を正規一様分散記憶方式（ u_0 − dm 方式

と略記）と呼ぶことにする。

ここに定義した$u-\mathrm{dm}$方式は，いつでも実現可能ではなく一定の条件の下でのみ構成できる。この条件について考えてみよう。

〔定理２〕[7]　$u-\mathrm{dm}$ 方式はnが

$$n=2^{p}\quad(\ p\text{は正整数}\ )\tag{13}$$

と表されるとき，構成可能* であり，そのとき行列 A_n は

$$A_n=\begin{bmatrix} A_{\frac{n}{2}} & A_{\frac{n}{2}} \\ A_{\frac{n}{2}} & -A_{\frac{n}{2}} \end{bmatrix}\tag{14}$$

なるアダマール行列で表される。但し$A_1=1$である。

〔例１〕　ここで４次の$u_0-\mathrm{dm}$ 方式（$m=3$）を例に dm 方式を説明しよう。

まず，記憶変数x_1，x_2，x_3 をそれぞれ次の値とする。

$$x_1=123,\quad x_2=252,\quad x_3=-325$$

いま x_1 を記憶系に書込む場合には，W_1（K_4，a_1，d_1）変換を行ってr_1 を求める。

$$r_1=d_1K_4a_1x_1=1\cdot\begin{bmatrix}1&&&0\\&1&&\\&&1&\\0&&&1\end{bmatrix}\cdot\begin{bmatrix}1\\1\\1\\1\end{bmatrix}\cdot123=\begin{bmatrix}123\\123\\123\\123\end{bmatrix}$$

これより書込み変数yは

$$y\Leftarrow y+r_1=\begin{bmatrix}0\\0\\0\\0\end{bmatrix}+\begin{bmatrix}123\\123\\123\\123\end{bmatrix}=\begin{bmatrix}123\\123\\123\\123\end{bmatrix}$$

と書き改められる。同様にして x_2，x_3 を書込むと，変数yは

$$y=r_1+r_2+r_3=\begin{bmatrix}123\\123\\123\\123\end{bmatrix}+\begin{bmatrix}252\\-252\\252\\-252\end{bmatrix}+\begin{bmatrix}-325\\-325\\325\\325\end{bmatrix}=\begin{bmatrix}50\\-454\\700\\196\end{bmatrix}$$

*　この条件以外でも，nが４の倍数のとき構成できるときがある。但し構成法は式(14)の方法と異なる。

となる。

次に記憶された情報を読出してみよう。例えば x_3 を読出す場合には$R(1/4\cdot d_3, \boldsymbol{a}_3, K_4)$変換を用いる。

このとき，

$$x_3 \Leftarrow \frac{1}{4\cdot d_3}\,\boldsymbol{a}_3^{t}\cdot K_4^{-1}\circ \boldsymbol{y} = \frac{1}{4}\quad\begin{bmatrix}1&1&1&1\end{bmatrix}\cdot\begin{bmatrix}1&&&0\\&1&&\\&&1&\\0&&&1\end{bmatrix}\cdot\begin{bmatrix}50\\-454\\700\\196\end{bmatrix}$$

$$= -325$$

となり，元の値が読出される。

このような記憶方式では読出し演算を工夫することにより，種々の情報を取出すことができる。これらのうち基本的なものについて調べる。

〔定理３〕　記憶変数x_j の定数倍 $c x_j$ を読出すときには，読出し演算を$R(c/n\cdot d_j, \boldsymbol{a}_j, K_n)$に選べばよい。又，特に u_0 − dm 方式において $c_1 x_i \pm c_2 x_j$ の値を読出すには$R(1/n, c_1\boldsymbol{a}_j \pm c_2\boldsymbol{a}_j, K_n)$ の読出し演算を用いればよい。但し，記憶装置の容量は十分大きいとする。　（証明略）

3. 分散記憶と記憶精度

分散記憶方式で情報を保持する場合の記憶容量と記憶精度の関係について調べてみよう。いま，αビットで表されている情報を記憶しようとするとき，従来の離散形記憶方法ではαビットの記憶容量を用意すればよかった。しかし，本方法では記憶の書込み，読出しにおいて演算を行い，情報を分散させたり，又，その情報を合成することにより，情報が変形されている。このような方法においては，記憶した情報を誤りなく取出すために必要な記憶容量はどのようになるかについて考える。まず，本方式のうち，最も基本的なu_0 − dm方式の場合について調べる。

〔定理４〕　αビットで表されているn個の情報をn次のu_0 − dm方式で記憶し，誤りなく読出すためには，各記憶領域当り $\beta = (\alpha + \log_2 n)$ ビットの記憶容量が必要であり，全体では$\gamma = n(\alpha + \log_2 n)$ビット必要になる。

（証明）　最初に述べたように，記憶情報x_j $(j=1, 2, \cdots\cdots, n)$は$\alpha$ビットで表される整数であり

$$|x_j| \leqq 2^{\alpha-1} \tag{15}$$

の範囲の値をとる。いま，すべての x_j ($j=1,2,\cdots\cdots,n$) について W_j (K_n, a_j, d_j) 変換を行い，書込みを行うと書込み変数 y_j は

$$y_j=\sum_{j=1}^{n} k_i a_i d_j x_j \qquad (i=1,2,\cdots\cdots,n) \tag{16}$$

と表される。ここで，x_j が式(15) の範囲の値をとるとき，y_i の絶対値の最大値 $|y_i|_{\max}$ を求める。いま，この値は

$$|y_i|_{\max}=|k_i|\sum_{j=1}^{n}|a_{ij}|\cdot|d_j|\cdot|x_j|_{\max} \tag{17}$$

となるが，u_0 − dm 方式においては

$$\left.\begin{array}{l} |a_{ij}|=1 \\ |d_j|=1 \\ |k_i|=1 \end{array}\right\} \quad (i,j=1,2,\cdots\cdots,n) \tag{18}$$

であり，又，式(15)から

$$|x_j|_{\max}=2^{\alpha-1} \tag{19}$$

であることから

$$|y_i|_{\max}=n\cdot 2^{\alpha-1} \tag{20}$$

となる。従って，y_i は $-|y_i|_{\max}$ から $|y_i|_{\max}$ の範囲の整数値をとることから，この変数を記憶するためには

$$\beta=\log_2(2\cdot 2^{\alpha-1}\cdot n)=\alpha+\log_2 n \tag{21}$$

ビット必要になり，全体として n 個あることから，$r=n(\alpha+\log_2 n)$ ビット必要である。読出し演算は書込み演算と全く逆のことから，上述のビット数だけ用意すれば誤りなく読出せる。

（証明終）

この定理は α ビットの情報を記憶するとき，離散形では α ビットの記憶容量を用意すればよいのに対して，分散形の記憶においては ($\alpha+\log_2 n$) ビットの容量が必要であることを示している。

そこで，記憶変数が α ビットの整数で表されるとしたとき，書込み変数の整数部分が ($\alpha+\log_2 n$) ビットで表される分散記憶方式を等容量形分散記憶方式と呼ぶことにする。この記憶方式は興味ある性質を有する。まず次の方式から考えよう。

〔定義5〕　$u-\mathrm{dm}$ 方式において，行列 K_n が単位行列であり，d_j（$j=1,2,\cdots\cdots,n$）が

$$n=\sum_{j=1}^{n}|d_j| \tag{22}$$

の関係を満たすとき，等容量形一様分散記憶方式（$cu-\mathrm{dm}$ 方式）と呼ぶことにする。

$cu-\mathrm{dm}$ 方式は式(17)から明らかなように，各 y_i（$i=1,2,\cdots\cdots,n$）の絶対値の最大値 $|y_i|_{\max}$ は

$$|y_i|_{\max}=\sum_{j=1}^{n}|d_j|\cdot|x_j|_{\max} \tag{23}$$

となり，式(19)及び(22)より

$$|y_i|_{\max}=n\cdot|x_j|_{\max}\leqq n\cdot 2^{\alpha-1} \tag{24}$$

である。従って，y_i は（$\alpha+\log_2 n$）ビットで表せることになり等容量形の分散記憶方式になっている。

このように定義した $cu-\mathrm{dm}$ 方式は $u_0-\mathrm{dm}$ 方式と異なり，次のような性質を有している。

〔定理5〕　α ビットで表されている記憶変数 x_j（$j=1,2,\cdots\cdots,n$）を容量（$\alpha+\log_2 n$）ビットの n 次の $cu-\mathrm{dm}$ 方式で記憶したとき，d_j が $|d_j|<1$ であれば，x_j の α ビットの情報のうち，$\rho=-\log_2|d_j|$ ビットの情報が失われる。但し $\rho>\alpha$ のとき，x_j のすべての情報が失われる。

（証明）　記憶変数 x_j（$j=1,2,\cdots\cdots,n$）は式(7)のように

$$|x_j|\leqq 2^{\alpha-1}$$

の範囲の整数であるので，式(24)から明らかなように，書込み変数 y_i の整数部分は（$\alpha+\log_2 n$）ビットで表される。しかるに，記憶方法としては容量（$\alpha+\log_2 n$）ビットの $cu-\mathrm{dm}$ 方式であるため，各 y_i の小数部分の情報は記憶されない。式(16)から書込み変数は

$$y_i=\sum_{j=1}^{n}a_{ij}\,d_j\,x_j \tag{25}$$

と表され，一様性の性質から

$$|a_{ij}|=1\quad(i,\ j=1,2,\cdots\cdots,n) \tag{26}$$

であるため，$d_j x_j$（$j=1,2,\cdots\cdots,n$）の小数部分は $-\log_2|d_j|$ ビットとなる。
従って，x_j が α ビットで表されていると，そのうちの $-\log_2|d_j|$ ビットの情報が失われる。

（証明終）

ここで，容量（$\alpha+\log_2 n$）ビットの cu-dm 方式についてパラメータ d_j と x_j の精度の関係を調べよう。式(16)で与えられるように，各 y_i は

$$y_i = \sum_{j=1}^{n} a_{ij}\, d_j\, x_j$$

と表される。cu-dm方式ではパラメータ a_{ij} は $|a_{ij}|=1$ であり，d_j の絶対値の総和は n に正規化されていることを考えると，y_i は各記憶変数 x_j に重み $|d_j|$ をつけ加え合せて値を得ている。従って各 y_i は $|d_j|/n$ の割合で x_j の情報を含んでいることになる。これらのことから $|d_j|$ を大きくしていくと y_i には x_j の情報が多く含まれることになり，極端に $|d_j|$ が n になると

$$y_i = n\, a_{ij}\, x_j \quad (i=1, 2, \cdots\cdots, n) \tag{27}$$

となり，すべての y_i には x_j の情報のみが含まれ，ほかの記憶変数の情報は全く含まれなくなる。逆にパラメータ d_j を小さくしていくと，$d_j\, x_j$ の値が小さくなり，y_i における $d_j x_j$ の占める割合は小さくなる。特に，書込み変数 y_i の小数点以下の数値が無視されることを考えると，$|d_j|$ が1より小さくなれば，上述の定理のとおり x_j の情報の一部が失われることになる。このように，パラメータ d_j は記憶変数 x_j の情報を各 y_i に,どの程度大きく反映させるかを決める係数になっていることが分かる。

次に，等容量形分散記憶方式を一般的な形で定義しておこう。

〔定義6〕　dm方式において，d_j（$j=1, 2, \cdots\cdots, n$）を任意に選んだとき，行列 K_n の各要素 k_i（$i=1, 2, \cdots\cdots, n$）が

$$k_i = \frac{n}{\sum_{j=1}^{n} |a_{ij}| \cdot |d_j|} \tag{28}$$

の関係を満たすとき，この方式を等容量形分散記憶方式（c-dm方式）と呼ぶことにする。

c-dm方式も式(17)から明らかなように，書込み変数の整数部分は記憶変数のそれより $\log_2 n$ ビット多くなっている。表1にこれまでに定義した分散記憶方式を分類している。

表1　分散記憶方式の分類

- 分散記憶方式（dm方式）
  - 一様分散形（u-dm方式）
    - 等容量形（cu-dm方式）
      - 正規一様分散形（u_0-dm方式）
    - 非等容量形
  - 非一様分散形
    - 等容量形（c-dm方式）
    - 非等容量形
      - 単純記憶形（s-dm方式）

4. 記憶破壊と精度

情報を分散して記憶したとき，部分的に記憶破壊が起ると，もとの情報はどのような影響を受けるかを，これまでに提案した幾つかの方式を例に定量的に調べてみよう。ここで，議論の準備のため，記憶モデルについてもう少し述べておこう。記憶モデルは図1のように，n個の記憶領域に分かれており，第i番目の領域には（$\alpha+\log_2 n$）ビットで表される整数の書込み変数y_iが記憶されているとする。一方，記憶変数$\{x_j \mid j=1, 2, \cdots\cdots, m\}$は$\alpha$ビットで表される整数であり，各$x_j$は次の範囲の値をとる確率変数とし，

$$|x_j| \leqq 2^{\alpha-1} \quad (x_j \fallingdotseq 0)$$

各々独立に，一様な確率分布に従っているとする。いまx_jの平均的な値を調べるため，x_jの期待値μ_x，x_j^2の期待値σ_x^2を求めると

$$\mu_x = \overline{x_j} = 0 \tag{29}$$

$$\sigma_x^2 = \overline{x_j^2} = \frac{1}{6}(2^{\alpha}+1)(2^{\alpha-1}) \tag{30}$$

となる。

ここで，分散記憶の記憶破壊の一般性を失うことなく議論を簡単にするため，図2のように記憶領域の最初のp個が壊され，それらの書込み変数の値が失われ

$$y_s = 0 \quad (s=1, 2, \cdots\cdots, p) \tag{31}$$

になったとしよう。ここで,

$$v = p/n \tag{32}$$

を記憶の損失率と呼ぶことにする。

このように記憶が壊れたとき，記憶された情報がどのように影響を受けるかについて調べよう。まず，dm方式の最も基本的な方法の一つである u_0 − dm 方式について調べる。

〔定理6〕　n 次の u_0 − dm方式において，記憶損失率が v のとき，記憶変数 x_j （ $j=1$，2，……，n ）が記憶破壊に伴って受ける平均2乗誤差 e_j^2 は $v\,\sigma_x^2$ である。

（証明）　損失率 v のときの x_j の読出し値を x_j' とすると

$$x_j' = \frac{1}{n} a_j^t\ K_n^{-1} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ y_{p+1} \\ \vdots \\ y_n \end{bmatrix} = \frac{1}{n} a_j^t\ K_n^{-1} \left(Y - \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right)$$

$$= x_j - \frac{1}{n}\ a_j^t \begin{pmatrix} 1 & & & & 0 \\ & \ddots_{p} & & & \\ & & 1 & & \\ & & & 0 & \\ 0 & & & & \ddots\ 0 \end{pmatrix} A_n\ X \tag{33}$$

となる。従って,

$$x_j - x_j' = \frac{1}{n} \sum_{s=1}^{n} \sum_{t=1}^{p} a_{tj}\ a_{ts}\ x_s \tag{34}$$

が得られる。これより x_j の受ける誤差の平均2乗値 e_j^2 は

$$e_j^2 = E\left[\ (x_j - x_j')^2\ \right] = \frac{1}{n^2} \sum_{s=1}^{n} \sum_{v=1}^{n} \sum_{t=1}^{p} \sum_{w=1}^{p}$$

$$\cdot\ a_{tj}\ a_{wj}\ a_{ts}\ a_{wv}\ E\ (\ x_s\ x_v\ ) \tag{35}$$

となる。但し E は期待値を表す。

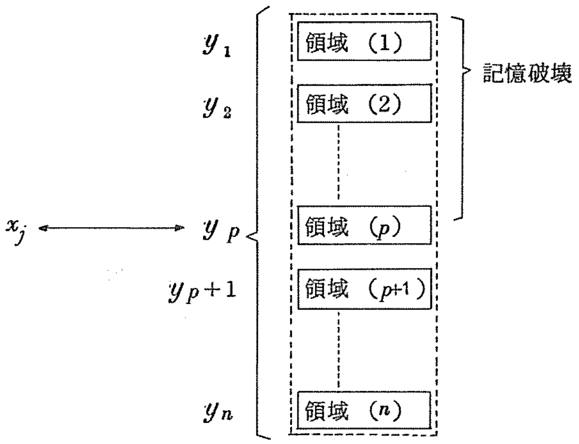

図２　記憶破壊のモデル

Fig. 2-A model of vanishment of the stored information.

ここで，行列 A_n は対称行列であり，各 x_j は確率的に独立であることを考えると，

$$e_j^2 = \frac{1}{n^2}\sum_{w=1}^{p}\sum_{l=1}^{p} a_{jl}\, a_{jw}\left(\sum_{s=1}^{n} a_{sl}\, a_{sw}\, \sigma_x^2\right) \tag{36}$$

となるが，ベクトル $\boldsymbol{a}_j$ （ $j=1, 2, \cdots\cdots, n$ ）は直交しているので

$$e_j^2 = \frac{1}{n}\sum_{l=1}^{p} a_{jl}^2\, \sigma_x^2 = v\sigma_x^2 \tag{37}$$

となる。（証明終）

この定理は記憶損失率が v のとき，損失に伴うもとの情報 x_j が受ける影響は $v\sigma_x^2$ であり，σ_x^2 は x_j の平均２乗値であることを考えると，全体のうち v の割合の情報が失われることを示している。このことは u_0 − dm方式が情報を n 個の領域に等しく分けて保持していることからも理解できる。

更に一般的な分散記憶方式である cu − dm 方式の場合について述べよう。

〔定理７〕 n 次の cu −dm 方式において，一つの記憶領域が壊れたとき，記憶変数 x_j （$j=1, 2, \cdots\cdots, n$ ）が記憶破壊に伴って受ける誤差の平均２乗値 e_j^2 は

$$e_j^2 = \frac{\sigma_x^2}{n^2 d_j^2}\sum_{s=1}^{n} d_s^2 \tag{38}$$

となる。

（証明）　x_j の読出し値を x_j' とすると

$$x_j' = \frac{1}{nd} a_j^t K_n^{-1} \begin{bmatrix} 0 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \frac{1}{nd} a_j^t K_n^{-1} \left(y - \begin{bmatrix} y_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right)$$

$$= x_j - \frac{1}{n d_j} a_{1j} \cdot \sum_{s=1}^{n} a_{1s} d_s x_s \tag{39}$$

となる。これより，

$$e_j^2 = E(x_j - x_j')^2 = \frac{1}{n^2 d_j^2} \sum_{s=1}^{n} \sum_{r=1}^{n}$$

$$\cdot a_{1s} a_{1r} d_s d_r E(x_s x_r) \tag{40}$$

となるが，各 x_s は確率的に独立であることより

$$e_j^2 = \frac{\sigma_x^2}{n^2 d_j^2} \sum_{s=1}^{n} d_s^2 \tag{41}$$

が得られる。　（証明終）

この定理は記憶破壊に伴う記憶誤りと係数 d_j（$j=1, 2, \cdots\cdots n$）の関係を表している。この結果を更に分かりやすくするために，一つの係数 d_q を順次大きくし，ほかの係数を

$$d_j = \frac{n - d_q}{n-1} \quad (j = 1, 2, \cdots\cdots, q-1, q+1, \cdots\cdots, n) \tag{42}$$

としたときの記憶変数 x_q の誤差 e_q^2 の変化を図3に表している。まず，$d_q = 1$ のとき，e_q^2 / σ_x^2 は $1/n$ となり定理6の結果と一致する。これより d_q を順次大きくしていくと，e_q^2 は急速に小さくなり，$d_q = n$ になると e_q^2 / σ_x^2 は $1/n^2$ になる。前にも述べたように，$d_q = n$ のとき y_i（$i=1, 2, \cdots\cdots, n$）はすべて x_q のみの情報を記憶しているため，n 重に x_q の情報を保持していることになる。従って，破壊に伴う誤差 e_q^2 は $d_q = 1$ の場合の $1/n$ になると考えられる。これらのことから d_q を大きく選ぶと，記憶破壊が起ってもより正確に x_q を読出せることを示している。定理5及びここに述べたことから，係数 d_q を小さくすると記憶情

報 x_q があいまいになり，且つ記憶破壊に対して大きく影響を受ける。一方，d_q を大きく選んでいくと，正確に記憶できるばかりではなく，記憶破壊に対しても影響を受けにくくなる。

ここで，我々人間の記憶破壊との関係について述べてみよう。人間は数多くのことを記憶することができるが，これら記憶情報は時間とともに失われていき，最後には忘れてしまう。このことは，記憶内部で一定の割合でランダムに記憶破壊が生じていると考えて差支えないであろう。しかし，同一時期に覚えたことでも，すぐ忘れる情報もあれば，かなり長時間経過しても鮮明に記憶している情報と色々である。このことをdm方式と対比させれば，記憶変数情報 x_j を正確に，且つ長時間忘れることなく記憶するためには係数 d_j を大きく選ぶことに対応し，逆に変数情報 x_j は概数でよく，又，それほど長時間記憶しておく必要もない場合には d_j を小さく選ぶことに対応している。これらのことから，次のような定義を行なう。

〔定義7〕　cu－dm 方式におけるパラメータ d_j（$j=1,2,\cdots\cdots,n$）を記憶変数 x_j の印象係数と呼ぶことにする。

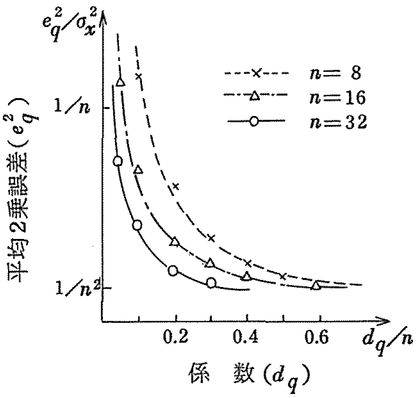

図3　係数 d_q と記憶誤差 e_q^2 の関係

Fig.3- Relations between the memory error e_q^2 and the coefficient d_q

5. 分散記憶方式の特徴

ここではこの記憶方式のもつ特徴をまとめてみよう。

(1)　本方式は情報をディジタル形式で記憶するにもかかわらず，取扱う情報はアナログ情報が適している。これを記憶破壊を例にとって説明しよう。本方式では部分的な記憶破壊でもすべての記憶項目に影響が及ぶ。しかし，その記憶損の影響は記憶項目すべてに分散されるため，一つ一つの項目が受ける影響は項目の数に反比例して小さくなる。一方，離散形の記憶においては失われた部分に対応する記憶項目のみが影響を受け，ほかの部分が全く影響を受けないのと対照的である。ここで，ディジタル情報とアナログ情報のもつ性格を考えてみよう。一般にディジタル情報の取扱いには論理演算的な性格が強く，誤りを極端に嫌う。もし誤りが起りそうな場合には，適当に冗長をつけ誤りを検出し，訂正してでも取扱うとする傾向が強い。他方，アナログ情報は本質的に誤差を含み，いくら正確に取扱ってみても一定以上の精度を得ることができないことが多い。このようにしてみると記憶誤りが分散される本方式はアナログ情報の記憶に適しており，離散形はディジタル情報の記憶に適していることが理解できる。

(2) 印象係数を適当に選ぶことにより，記憶情報の印象の強さまたは弱さなどを実現することが可能である。

(3) 本方式では，概数のみが必要な情報から，多くの有効数字が必要な正確な情報を記憶する場合も，その都度記憶項目に必要なビット割当てすることなく単に印象係数を変化させるだけで，すべての記憶領域を余すところなく使用することが可能である。

(4) この記憶方法は無番地記憶方式であるため，アドレス用信号が不要である。

これらの特徴を要約すると，分散記憶方式は柔軟性に富み，アナログ情報の記憶に適している記憶方法といえる。

6. むすび

情報を記憶領域全体に分散して記憶する分散記憶方式を提案し，その基本的な性質を調べた。その結果，本方式は従来から工学的に広く用いられている離散形記憶方法にない数多くの特徴を有していることが明らかになった。特に，ディジタル形式の記憶装置にアナログ情報を記憶する際に用いれば非常に有効であることが判明した。このことは，現在ディジタル形式の情報処理では非能率である分野，また本質的にアナログ情報を処理している分野へ応用すれば有益であると思われる。更に，我々人間の記憶行為においてみかけられる現象と類似点が多く，人間の情報処理がアナログ的であることの間接的な説明になるように思える。

文　　献

(1) 相磯秀夫他：“新らしい計算機アーキラクチャ——非ノイマン機能——”，情報処理，Vol.19，No12(1968).

(2) B.Parbami：“Associatron Memories and Processors.” Proc.IEEE，Vol.61，No.6 (1973).

(3) K.Nakano：“Association－a model of associative memory”，IEEE Trans. Vol.SMC－2，No.3 (1972).

(4) T.Kohnen：“Associative memory.” Springer－Verlag.(1977).

(5) 上坂，尾関：“連想記憶の二，三の性質”，信学論（D），Vol.59－D，No.8(1976).

(6) K. Fukunaga and T.Kasai：“An associative memory with distributed imformation” IECE Japan，Vol.62－E，No.9(1979).

(7) K.G.Beauchamp：“Walsh function and their applications”，P.24，Academic Press (1975).