アソシアトロンの原理[*]

わっしー教授[†]

平成 31 年 2 月 11 日

1　基礎

原著[1]に沿って、ネットにある情報[2]を参考に、中野馨氏の創造したアソシアトロンの原理をメモがてらに書いておく。

第 p パターンを $1, 0, -1$ のいずれかの値からなる n 次元ベクトル x^p として表現する。一つの要素が励起している状態を 1、していない状態を -1、0 は無効な状態ということになる。すなわち、

$$x^p = \begin{pmatrix} x_1^p \\ x_2^p \\ \vdots \\ x_n^p \end{pmatrix}.$$

である。

上記のベクトルに左から、それをただ転置しただけのベクトルをかける。すなわち、$x^p x^{pT}$ とすると、結果は行列になる。この行列を M^p とすると

$$M^p = \begin{pmatrix} x_1^p x_1^p & x_1^p x_2^p & \cdots & x_1^p x_n^p \\ x_2^p x_1^p & x_2^p x_2^p & \cdots & x_2^p x_n^p \\ \vdots & \vdots & \ddots & \vdots \\ x_n^p x_1^p & x_n^p x_2^p & \cdots & x_n^p x_n^p \end{pmatrix}$$

M^p の第 i 行、第 j 列要素を m_{ij}^p とすると、$m_{ij}^p = x_i^p x_j^p$ である。x_i^p は、$0, -1, 1$ のいずれかであるから、この m_{ij}^p もまた、そのいずれかの値となる。ここが、単純なメカニズムになる一つの工夫である。

次に、この元のパターンを想起させる n 次元ベクトルというものを考えよう。これを y としよう。いま、この想起ベクトルとして、第 t 要素が、元のパターンベクトル x^p の第 t 番目の要素 x_t^p

[*] http://www.ibot.co.jp/wpibot/?p=2456 に掲載されたもの
[†] 松竹芸能所属芸人 ツイッター @wassiisg
[1] 中野馨,『アソシアトロン—連想記憶のモデルと知的情報処理』昭晃堂, 1979
[2] http://www.sist.ac.jp/ kanakubo/research/neuro/associatron.html 等

と同じ（それは必ず、-1 か 1 である !!）で、他が全てゼロだったとしよう。すなわち、

$$y = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ x_t^p \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

である。

この想起ベクトル y を M^p 行列に、左からかけ $M^p y$ としてみよう。すると、それは M^p の第 t 番目の列ベクトルに、x_t^p をかけただけの n 次元ベクトルとなる。すなわち、

$$M^p y = \begin{pmatrix} x_1^p x_t^p x_t^p \\ x_2^p x_t^p x_t^p \\ \vdots \\ x_n^p x_t^p x_t^p \end{pmatrix}.$$

である。ここで、上にも書いたように x_t^p は、必ず -1 か 1 なので、$x_t^p x_t^p$ は、必ず 1 である。結局、

$$M^p y = \begin{pmatrix} x_1^p \\ x_2^p \\ \vdots \\ x_n^p \end{pmatrix}.$$

となり、元のパターベクトルが再現されるのである。y は、元のベクトルの 1 つの要素だけが等しいだけなのに、元のベクトル全体が再現されるということになる。

私にとっては、ここまでの点でも素晴らしいアイデアと思えたが、原著によれば、これは「当然のこと」なのだそうだ[3]。

2　展開

原著によれば、この後の展開が決定的な飛躍なのだそうだ。特に以下の（2）に記載した点である。ただ、私には、あとの展開は、この原理から簡単に推測できる。でも、確かに、飛躍的な発想であることは確かだ。

（1）もし、想起ベクトル y が元のパターンの 1 要素ではなく、色々あった場合。つまり、上では、y の一つの要素が、元のパターンと同じで、他は 0、すなわち無効という場合だったが、y の要素が、元のパターンと同じものもあれば、違う場合もあるという状況である。y ベクトルを一般的に、次のように表してみよう。

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}.$$

[3] 原著、p.19.

$y_i, i=1,2,\cdots,n$ は、$-1,1$ のいずれかである。そして、$M^p y$ するとどうなるか。それは多数決原理である。もちろん、作られたベクトルは、必ずしも $1,0,-1$ ではなく、正、負ないしはゼロである。しかし、単純に、正であれば 1、負であれば -1 でなければ 0 と置き換えれば良い。ただ、そういうことなのだが、論文などでは量子化関数というのを用いて表現されるので、それも書いておく。量子化関数とは、引数の要素を正、ゼロ、負に対応して $1,0,-1$ 変換する関数であり、次のように表される。任意のベクトル（行列でも良い）V について、

$$\phi(V) = \begin{cases} 1 & (v_i > 0) \\ 0 & (v_i = 0) \\ -1 & (v_i < 0) \end{cases} \quad \forall v_i \in V$$

として、$z = \phi(M^p y)$ を考えるということである。このベクトル z は、想起ベクトルの要素 y_i に、元のパターンのものと一致するものが多ければ、元のパターンに近くなるし、そうでなければ外れてくる。

（２）一番、重要なのは、複数のパターンを記憶させる場合である。いま、同じ次元で別な内容のパターンが k 種類あったとしよう（$x^1, x^2, \cdots, x^k,$)。このとき上記で示したパターン行列 M が k 個作られる事になる。これらを $M^1, M^2, \cdots, M^k$ としよう。いま、このパターン行列を全て足して、$1,0,-1$ に変換したものを M とする。すなわち、

$$M = \phi\left(\sum_{p=1}^{k} M^p\right)$$

である。加えた結果に量子化関数を適応して、その要素が正であれば 1、負であれば -1 でなければ 0 と置き換たものにしている。この段階では、量子化関数を適応しないほうがいいのかもしれないと思ったが、原著には、上記と同じように、このマトリクス自体も量子化している[4]。こうすると、ある想起ベクトル y があって My を作ると、（１）と同じような多数決原理が効いてきて、元のベクトルに最も近いパターンを抽出することができるわけである。y が変われば、最も近いパターンも変わってくる。

すなわち、

$$z = \phi\left(\phi\left(\sum_{p=1}^{k} M^p\right) y\right)$$

で、この z から、もっとも近いパターンを計算できるというわけである。

これでおしまい。

3 まとめ

何しろシンプルである。応用の見通しもすっきりわかる。一つの細胞が $1,0,-1$ の三つの状態を取るので、最低 2 ビット必要になる。今日、メモリはとても安くて早い。現代的なコンピュータの状況での可能性を探ると面白そう。

[4]中野馨、『人間情報工学：バイオニクスからロボットまで 』、コロナ社 , 1996 の中で、マトリクス自体は量子化しない場合について記述されている。p.140.