General Mathematics Vol. 13, No. 2 (2005), 35–50

On the Hopfield algorithm. Foundations and examples

Nicolae Popoviciu and Mioara Boncuţ

Dedicated to Professor Dumitru Acu on his 60th birthday

Abstract

The work deals with the Hopfield networks and uses the vector description of the theory, rather then element by element one. The theoretical central part of the work is related with the energy theorem and a Hopfield algorithm based on vector form is elaborated (all the corresponding dimensions are given). This algorithm solves the store-recall problem. The algorithm is used to solve several numerical examples.

2000 Mathematics Subject Classification: 68T05, 82C32

1 Notations and Foundations

A. The earliest recurrent neural network has independently begun with Anderson (1977), Kohonen (1977), but Hopfield (1982) presented a complete mathematical analysis of such a subject [4], page 50. That is why this network is generally referred to as the Hopfield network. The Hopfield network

consists of a set of n interconnected neurons. All neurons are both input and output neurons. Hence, by using the bidirectional associative memory (BAM) notations [7], the input layer Sx is the same with the output layer Sy. This shows that we can consider the Hopfield network as a particular case of the BAM, although there is a doubt [2], page 141 that was the way the Hopfield memory originated.

B. The set of input vectors (input data, learning data, training vectors, input patterns) is a set of vectors of N the form

$$I(x) = \{x(1); x(2); ...; x(N)\},\ x(k) \in H^n,\ k \in 1, N$$

$$x(k) = (x_1(k)x_2(k) \cdots x_n(k))^T$$

where H is the Hamming space having only the elements -1 and +1 (bipolar elements) and T means the transposition.

Originally, Hopfield chose for each $x(k)$ the binary activation values 1 and 0, but using bipolar values +1 and -1 presents some advantages [4], page 50, [1], page 49.

For an input vector x we denote by $x^c \in H^n$ the complement, where the value 1 from x becomes -1 in x^c and vice versa. So, if we encode $I(x)$, we also encode its complement $I(x^c)$.

Definition. The **Hamming distance** between two vectors having the same type $u, v \in H^n$ is a function $DH[u, v]$ representing the number of bits that are different between u and v.

The Hamming distance DH is related to the Euclidian distance DE by the equation [Free, page 129] $DE = 2\sqrt{DH}$.

C. Now we discuss some main characteristics of a Hopfield network.

1. The first aim of the Hopfield network is **to store** the input data $I(x)$ (store phase).

2. Then, the second aim is **to recall** one input vector from $I(x)$ if one knows a test vector xr, with $xr \in H^n$ (retrieval phase). Generally, a test vector is a noisy vector of $I(x)$. This characteristic makes the Hopfield network useful for restoring degraded images [5], page 373. Hence the Hopfield network is used to solve a store-recall problem.

3. In the Hopfield network there are weights associated with the connections between all the neurons of layer Sx. We organize the weights in a square and symmetric matrix

$$W = W_{nxn},\ W = (w_{ij}),\ w_{ij} = w_{ji}.$$

All connections inside the layer Sx are bidirectional and the units may, or may not, have feedback connections to themselves. We can take $w_{ii} = 0$ (see below the proposition 2).

We denote **the columns** of W by vectors $w_j \in R^n$, $j = 1, n$ and **the lines** of W by vectors

$$line_i \in R^n,\ i = 1, n.$$

4. The Hopfield network is a fully interconnected network.

5. The matrix W can be determined in advance if all the training vectors $I(x)$ are known.

Method 1. We use the formula

$$W = \sum_{k=1}^{N} x(k)x(k)^T,\ w_{ii} = 0 \tag{1}$$

Here it is possible to know the input vectors $x(k)$ one after one.

Method 2. We use a dot product

$$w_{ij} = \sum_{k=1}^{N} x_i(k)x_j(k),\ i \neq j,\ w_{ii} = 0. \tag{2}$$

Here it is necessary to know from the beginning all the input vectors $x(k)$. This matrix does not change in time. After the matrix W was computed, we say the input data $I(x)$ was stored. The W matrix is also named the correlation matrix.

Remark 1. The first method is more useful in applications.

6. Having the matrix W and a test vector xr we look for the corresponding training vector. In order to find the corresponding vector of $I(x)$ we use the Hopfield Algorithm (see below).

7. The Hopfield network (and BAM network) has a major limitation [2], page 133, [3], page 42, [4], page 52 : the number of input patterns that can be stored and accurately recalled is limited by the relation $N < 0.15n$.

8. The store-recall problem can be improved [1], page 50, [5], page 373, by modifying the input set to contain orthogonal vectors , with dot product zero i.e. $< u, v >= 0,\ u, v \in I(x)$.

2 The Hopfield Algorithm

From the known input set $I(x)$, the Hopfield Algorithm retrieves a input vector with the help of a test vector xr (noisy vector). At the beginning we denote the test vector by $xr = xr(1)$.

At the time t (natural number), the algorithm propagates the information inside the Sx layer as follows:

a) compute activation

$$net_j(t) =< w_j, xr(t) >,\ j = \overline{1, n} \tag{3}$$

where w_j is the column j of the matrix W.

b) update the values on Sx layer, namely the vector $xr(t)$ becomes $xr(t+1)$ with the components [4], page 51, [2], page 143

$$xr_j(t+1) = \begin{cases} +1, & net_j(t) > 0 \\ -1, & net_j(t) < 0 \\ xr_j(t), & net_j(t) = 0 \end{cases}, \ j = \overline{1,n}. \tag{4}$$

We notice that [3] page 41 updates the above values only with the first two positions, as for sign function.

The Hopfield has the following steps.

Step 1. One knows the input set $I(x)$ and store it by computing the matrix $W = W_{nxn}$ with formula (1) or (2). One or more test vectors xr are given. Put $t = 1$.

Step 2. (optional) Compute the complement set $I(x^c)$ and the Hamming distance between the vectors of $I(x)$ and the test vectors xr. Check if the input set $I(x)$ is orthogonal or not.

Step 3. At the time t, propagate the information from the Sx layer to the Sx layer by the formulas (3) and (4). So we find the vector $xr(t+1) \in H^n$.

Step 4. Put $t+1$ instead of t and repeat the step 3 until there are no further changes in the components of $xr(t+1)$.

Remark 2. [2], page 133. If all goes well, the final stable state will recall one of the vectors $I(x)$ used to construct the store matrix W.

Remark 3. We hope that the final output of the Hopfield Algorithm is a vector who is closest in Hamming distance to the original vector xr.

3 The Hopfield Energy Function

Definition. Using [2], pages 137 and 139 we call **the energy function** associated with the input vector $x \in H^n$ and the matrix W the real number

$$E = E(x),\ E(x) = -x^T W x \tag{5}$$

$$E(x) = -\sum_{i=1}^{n}\sum_{j=1}^{n} x_i w_{ij} x_j \tag{6}$$

The energy function is called also Lyapunow function in the theory of dynamical systems. This is a quadratic form. In [3], page 42, the energy function is defined with a multiplying coefficient 1/2 at the beginning.

Energy Theorem. In a Hopfield neural network, any change by (4) in the components of the vector x results in a decrease in $E(x)$ function.

The proof has the same steps as the energy theorem for BAM networks [2], page 139 and [7].

Proposition 1. The correlation matrix W has the form

$$W = NI_n + S \tag{7}$$

where I_n is the identity matrix and S is a symmetric matrix having zeros on the main diagonal.

Proof. We compute W by (1) and obtain a sum of matrices having on the main diagonal the non-zero elements

$$x_1^2(k), x_2^2(k), ..., x_n^2(k), \quad k = 1, N.$$

Because $x(k) \in H^n$, then $x_i(k) \in \{-1, 1\}$, $x_i^2(k) = 1$ and we can construct the identity matrix I_n. In this way the matrix W becomes of the form (7). (End).

Proposition 2. In a Hopfield network, the energy function has the form

$$E(x) = -N^2 - x^T S x \tag{8}$$

Proof. By using the definition (5) and the above proposition 1 one obtains successively

$$E(x) = -x^T(NI_n + S)x = -Nx^t x - x^T S x = -N \cdot N - x^T S x$$

[End].

Remark 4. Because the N^2 is a constant value and finally we try to minimize the function $E(x)$, we can consider that

$$E(x) = -x^T S x \quad \text{or} \quad E(x) = -x^T W x \tag{9}$$

where the matrix W is computed by (1) and has the zeros values on the main diagonal. Hence, the change in energy , during a state change is independent of the diagonal elements on the weight matrix W [2], page 141.

4 Numerical Examples

A. We compute the weight matrix for several input vectors given by $I(x)$. First of all we change the zeros to negative ones in each input vector [5], page 373 and obtain the modified $I(x)$. If the initial input set $I(x)$ is orthogonal, the modified $I(x)$ could be not.

All the time we use (1) from method 1.

Example 1. $N = 3, n = 4$.

$$I(x) = \{x(1) = (-11-1-1)^T, x(2) = (1-11-1)^T, x(3) = (-1-1-11)^T\},$$

$$W = x(1)x(1)^T + x(2)x(2)^T + x(3)x(3)^T$$

Assign $w_{ii} = 1, i = 1, 4$ and the matrix W has the lines

$$0\ -1\ 3\ -1;\ -1\ 0\ -1\ -1;\ 3\ -1\ 0\ -1;\ -1\ -1\ -1\ 0.$$

Example 2. $N = 3,\ n = 5.$

$$I(x) = \begin{cases} x(1) = & (-1 & 1 & -1 & -1 & 1)^T \\ x(2) = & (1 & -1 & -1 & 1 & -1)^T \\ x(3) = & (-1 & -1 & 1 & -1 & -1)^T \end{cases}.$$

$$W = x(1)x(1)^T + x(2)x(2)^T + x(3)x(3)^T$$

Assign $w_{ii} = 0,\ i = 1, 5$ and the matrix W has the lines

$$0\ -1\ -1\ 3\ -1;\ -1\ 0\ -1\ -1\ 3;\ -1\ -1\ 0\ -1\ -1;\ 3\ -1\ -1\ 0\ -1;\ -1\ 3\ -1\ -1\ 0.$$

Example 3. $N = 4,\ n = 5.$

$$I(x) = \begin{cases} x(1) = & (1 & -1 & 1 & -1 & -1)^T \\ x(2) = & (-1 & 1 & -1 & 1 & -1)^T \\ x(3) = & (-1 & 1 & -1 & 1 & 1)^T \\ x(4) = & (1 & -1 & 1 & -1 & 1)^T \end{cases}.$$

The matrix W has the lines:

0 - 4 4 - 4 0; - 4 0 - 4 4 0; 4 - 4 0 - 4 0; - 4 4 - 4 0 0; 0 0 0 0 0 .

B. Now we apply the Hopfield algorithm.

Example 1. Using a Hopfield network, store and recall information for the input data:$n = 4, N = 2,$

$$I(x) = \begin{cases} x(1) = & (1 & -1 & 1 & 1)^T \\ x(2) = & (-1 & 1 & -1 & 1)^T \end{cases}.$$

and the text vectors $xr1 = (1\ 1\ -1\ 1)^T, xr2 = (-1\ -1\ 1\ 1)^T.$

Solution. The Hamming distances are

$$DH[x(1), xr1] = 2, DH[x(2), xr1] = 1$$

$$DH[x(1), xr2] = 1, DH[x(2), xr2] = 2.$$

The weight matrix $W = W_{4x4}$ is $W = \begin{pmatrix} 0 & -2 & 2 & 0 \\ -2 & 0 & -2 & 0 \\ 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$.

For test vector $xr1$, the results of the Hopfield algorithm are contained in the **Table 1.**

					xr1		
	1	2	3	4	1	2	3
1	0	-2	2	0	1	-1	-1
2	-2	0	-2	0	1	1	1
3	2	-2	0	0	-1	-1	-1
4	0	0	0	0	1	1	1
EF					4	-12	-12

We see that

$$xr1(1) \neq xr1(2) = xr1(3); STOP;$$

$$xr1(3) = x(2).$$

Hence, by the test vector $xr1$ we have found the input vector $x(2)$. EF means the Energy Function value, computed accordingly with formula (9). This function decreases because

$$EF[xr1(1)] = 4, EF[xr1(2)] = -12,$$

$$EF[xr1(3)] = EF[x(2)] = -12.$$

For test vector $xr2$, the results of the Hopfield algorithm are contained in the **Table 2**.

					xr2		
	1	2	3	4	1	2	3
1	0	-2	2	0	-1	1	1
2	-2	0	2	0	-1	-1	-1
3	2	-2	0	0	1	1	1
4	0	0	0	0	1	1	1
EF					4	-12	-12

We see that

$$xr2(1) \neq xr2(2) = xr2(3); STOP;$$

$$xr2(3) = x(1).$$

Hence, by the test vector $xr2$ we have found the input vector $x(1)$. The Energy Function decreases because

$$EF[xr2(1)] = 4, EF[xr2(2)] = -12,$$

$$EF[xr2(3)] = EF[x(1)] = -12.$$

Example 2. Using a Hopfield network, store and recall information for the input data $N = 6, N = 2$,

$$I(x) = \begin{cases} x(1) = & (1 & -1 & -1 & 1 & -1 & 1)^T \\ x(2) = & (1 & 1 & 1 & -1 & -1 & -1)^T \end{cases},$$

and the test vectors

$$xr1 = (1\ 1\ 1\ 1\ -1\ 1)^T$$

$$xr2 = (-1\ 1\ 1\ -1\ 1\ -1)^T$$

$$xr3 = (1\ 1\ 1\ -1\ 1\ -1)^T$$

$$xr4 = (1\ 1\ -1\ 1\ -1\ 1)^T$$

$$xr5 = (1\ -1\ 1\ 1\ -1\ 1)^T$$

Solution. The weight matrix $W = W_{6x6}$ is the following

$$W = \begin{pmatrix} 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 2 & -2 & 0 & -2 \\ 0 & 2 & 0 & -2 & 0 & -2 \\ 0 & -2 & -2 & 0 & 0 & 2 \\ -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & -2 & 2 & 0 & 0 \end{pmatrix}.$$

We recommend to compute the complement set $I(x^c)$ and all the complement of the test vectors xrk^c. The Hamming distances are

$$DH[x(1), xr1] = 2, DH[x(2), xr1] = 2$$

$$DH[x(1), xr2] = 6, DH[x(2), xr2] = 2$$

$$DH[x(1), xr3] = 5, DH[x(2), xr3] = 1$$

$$DH[x(1), xr4] = 1, DH[x(2), xr4] = 3$$

$$DH[x(1), xr5] = 1, DH[x(2), xr5] = 3.$$

For test vector $xr1$ (or $r1$), the results of the Hopfield algorithm are

contained in the **Table 3**.

							r1				
	1	2	3	4	5	6	1	2	3	4	4c
1	0	0	0	0	-2	0	1	1	1	1	-1
2	0	0	2	-2	0	-2	1	-1	1	-1	1
3	0	2	0	-2	0	-2	1	-1	1	-1	1
4	0	-2	-2	0	0	2	1	-1	1	-1	1
5	-2	0	0	0	0	0	-1	-1	-1	-1	1
6	0	-2	-2	2	0	0	1	-1	1	-1	1
EF							4	4	4	4	4

(In some places of the table, a negative number $-p$ is written on a column as - and p).

We see the cycling repetition

$$xr1(1) \neq xr1(2) \neq xr1(3) \neq xr1(4),$$

$$xr1(1) = xr1(3), xr1(2) = xr1(4); STOP;$$

the algorithm fails ; neither $xr1(4)$ nor $xr1(4)^c$ belong to $I(x)$, $I(x^c)$. The Energy Function has the constant positive value 4. The algorithm is unsuccessfully.

For test vector $xr2$ (or $r2$), the results of the Hopfield algorithm are

contained in the **Table 4**.

							r2		
	1	2	3	4	5	6	1	2	2c
1	0	0	0	0	-2	0	-1	-1	1
2	0	0	2	-2	0	-2	1	1	-1
3	0	2	0	-2	0	-2	1	1	-1
4	0	-2	-2	0	0	2	-1	-1	1
5	-2	0	0	0	0	0	1	1	-1
6	0	-2	-2	2	0	0	-1	-1	1
EF							-28	-28	-28

We see the repetition

$$xr2(1) = xr2(2); STOP;$$

$xr2(2) \notin I(x)$, but $xr2(2) \in I(x^c)$ and so we find the vector $x(1)$ from input set.

The algorithm is successfully.

For test vector $xr3$ (or $r3$), the results of the Hopfield algorithm are contained in the **Table 5**.

							r3				
	1	2	3	4	5	6	1	2	3	2c	3c
1	0	0	0	0	-2	0	1	-1	1	1	-1
2	0	0	2	-2	0	-2	1	1	1	-1	-1
3	0	2	0	-2	0	-2	1	1	1	-1	-1
4	0	-2	-2	0	0	2	-1	-1	-1	1	1
5	-2	0	0	0	0	0	1	-1	1	1	-1
6	0	-2	-2	2	0	0	-1	-1	-1	1	1
EF							-20	-20	-20	-20	-20

We see the cycling repetition

$$xr3(1) = xr3(3); STOP;$$

neither $xr3(3)$ nor $x3(3)^c$ belong to $I(x)$ and $I(x)^c$. The algorithm is unsuccessfully.

For test vector $xr4$ (or $r4$), the results of the Hopfield algorithm are contained in the **Table6**.

							r4		
	1	2	3	4	5	6	1	2	3
1	0	0	0	0	-2	0	1	1	1
2	0	0	2	-2	0	-2	1	-1	-1
3	0	2	0	-2	0	-2	-1	-1	-1
4	0	-2	-2	0	0	2	1	1	1
5	-2	0	0	0	0	0	-1	-1	-1
6	0	-2	-2	2	0	0	1	1	1
EF							-4	-26	-26

We see the repetition

$$xr4(2) = xr4(3); STOP;$$

$xr4(3) = x(1)$ and so we find the vector x(1). The Energy Function decreases because $EF[xr4(1)] = -4, EF[xr4(2)] = -26$.

The algorithm is successfully.

For test vector $xr5$ (or $r5$), the results of the Hopfield algorithm are

contained in the **Table 7**.

							r5		
	1	2	3	4	5	6	1	2	3
1	0	0	0	0	-2	0	1	1	1
2	0	0	2	-2	0	-2	-1	-1	-1
3	0	2	0	-2	0	-2	1	-1	-1
4	0	-2	-2	0	0	2	1	1	1
5	-2	0	0	0	0	0	-1	-1	-1
6	0	-2	-2	2	0	0	1	1	1
EF							-4	-28	-28

We see the repetition

$$xr5(2) = xr5(3); STOP;$$

$xr5(3) = x(1)$ and so we find the vector $x(1)$. The Energy Function decreases because

$$EF[xr5(1)] = -4, EF[xr5(2)] = -28.$$

The algorithm is successfully.

References

[1] Anderson D., McNeill G., *Artificial Neural Network Technology*, A DACS State-of the-Art Report, August, 1992.

[2] Freeman A. J., Skapura M. D., *Neural Networks: Algorithms, Applications and Programming Techniques*, Addison-Wesley Publishing Company, 1991.

[3] Jain K. A., Mao J., Mohiuddin K. M., *Artificial Neural Networks: A Tutorial*, IEEE, March, 1996.

[4] Krose B., Van der Smagt P., *An Introduction to Neural Networks*, Chapter 6, Eight Edition, University of Amsterdam, 1996.

[5] MacPhee-Cobb L., Chapter 7, *Introduction to Artificilal Intelligence*, 2003.

[6] Popoviciu N., *An Efficient Algorithm Based on Vector Computation*, to appear in General Mathematics, Lucian Blaga University of Sibiu, Romania.

[7] Popoviciu N., *On the Bidirectional Associative Memory Algorithm. Foundations and Examples*, to appear in Hyperion Journal, Hyperion University of Bucharest.

[8] Rao B., *C++ Neural Networks and Fuzzy Logic*, Chapter 10, MTBooks, IDG Books Worldwide, Inc. 1995.

Hyperion University of Bucharest
Mathematics - Informatics Faculty
E-mail: *npopoviciu@abt.ro*

Department of Mathematics / Faculty of Sciences
"Lucian Blaga" University of Sibiu
Str. Dr. I. Raţiu, nr. 5-7
550012 - Sibiu, Romania
E-mail: *mioara.boncut@ulbsibiu.ro*