Solving Problems of Maximum Likelihood Decoding of Graph Theoretic Codes via A Hopfield Neural Network

Hsiu-Hui Lin , Ja-Ling Wu , and Li-Chen Fu

Department of Computer Science and Information Engineering
National Taiwan University Taipei, 10764, Taiwan, R.O.C.

Abstract

This paper present a neural network approach to solving problem of decoding graph theoretic codes (GTC's). The equivalence relation has first been established between the problem of Maximum likelihood decoding (MLD) of graph theoretic codes and that of minimizing an energy function of the Hopfield networks associated with those graphs [1]. This, in turn, allows us to construct a Hopfield neural network which performs a function of MLD in a natural way. However, the existence of the local minima problem [2], though considerably relaxed in these nets, prevents the completeness of the new decoding approach. Therefore, we modify the traditional Hopfield model by adding a detection mechanism to overcome the problem. Statistical analysis and simulations are provided to show the effectiveness of the new model.

1 Introduction

There are two main aspects of the coding problem : 1) how to find a practical method to encode and 2) how to find a practical method to decode. For graph-theoretic codes, either cut-set codes or circuit codes, the generator matrices can be easily obtained via a spanning tree of its corresponding graph [3]. Therefore, what we have to consider for GTC's is how to decode them more efficiently. Besides, since the cut-set code and the circuit code of a given graph are mutually dual because of the orthogonal property between the cut set and the circuit subspaces [4], we need only to consider the decoding problem of one kind of GTC's , the cut-set code.

Bruck has established a connection between the Hopfield model and the cut-set codes[1] in two important lemmas. They first prove that finding a global maximum of the energy function of a network is equivalent to finding a minimum cut in a graph. Next, they prove that MLD of a cut-set code is equivalent to finding the minimum cut in a corresponding graph. That equivalent relationship between finding a global maximum of the energy function of a Hopfield network and the MLD of a cut-set code allows us to develop a neural network system to decode graph theoretic codes.

However, neural networks, though obtaining high computation speed through massive parallelism, has some serious problems, such as problems of spurious stable states and problems of slow convergence [5,6]. These defects will deteriorate the performance of the so constructed decoding system. Especially, the local minima problem will cause received words to be incorrectly decoded. To remedy this, a decoding system based on a modified Hopfield network is proposed. The key idea lies in that the states transition of the network is deterministic once the the architecture of the network is formed, given the graph and the received word and, hence, the convergence property will heavily depend on the choice of initial states. Furthermore, experimental results show that the local minima problem arising from decoding GTC's is not so severe as those from other Hopfield applications, such as Traveling Salesman problem (TSP) [6]. Therefore, we append a detection mechanism to the back of the original Hopfield network, which will cause the network to remain active by randomly selecting another initial states whenever the received word is incorrectly decoded.

Organization of this paper is as follows: In section 2, we briefly review some theoretical issues on graph theory, error-control coding and neural networks. Section 3 clearly describes how Hopfield neural network is applied to solve the MLD problem of GTC's, and later proposes a modified system to remedy the local minima problem. In section 4, experimental results are shown to illustrate the effectiveness of the proposed approach, and some discussions are given. Finally, conclusions are reached and future researches are discussed in section 5.

2 Preliminary

2.1 Background on Graph Theoretic Code

Let $\hat{G}$ be a connected m-vertex graph with n edges $e_1, e_2, \ldots, e_n$. A cut set of $\hat{G}$ is a set of edges such that removal of these edges will cause the resulting graph disconnected. We can represent a cut set in a vector form , $g = (g_1, g_2, \ldots, g_n)$, where $g_i = 1$ if e_i is an edge of the cut set and $g_i = 0$ otherwise, $1 \leq i \leq n$. It is well known that the vectors that correspond to the cut sets in a connected graph $\hat{G}$ form a linear vector space over GF(2) of dimension $m-1$, called cut-set space [6]. Hence, the cut-set space of a connected graph $\hat{G}$ is a $[n, m-1]$ linear block code, denoted by $C_{\hat{G}}$.

Similarly, there is a circuit code associated with a connected graph. These two codes are the so-called graph theoretic codes. However, since they are dual codes, we consider the cut-set code only for illustration in this paper [4].

Before the encoding/decoding problem and the error-correcting ability of GTC is discussed, some definitions are given below :

Definition 1 *The incidence matrix $A_{\hat{G}}$ of a graph $\hat{G} = (\hat{V}, \hat{E})$ is a $| \hat{V} | \times | \hat{E} |$ matrix in which row i is the vector of the set of edges incident upon vertex $i \in \hat{V}$.*

Definition 2 *The minimum distance d^* of a code is defined as the smallest Hamming distance of any apir of codewords, where the Hamming distance $d(X,Y)$ between codewords X and Y is the number of places in which they differ.*

Definition 3 *Given a graph $\hat{G}$, the MLD of the cut-set code $C_{\hat{G}}$ is to decode a received word Y as a codeword in $C_{\hat{G}}$ closest to Y in Hamming distance.*

The minimum distance is an important parameter for error-correcting ability [4] for it determines the number of errors that the code can correct. Let t be the largest integer satisfying $d^* \geq 2t+1$. Each codeword can be enclosed by a sphere of radius t and none of the spheres will be intersecting. A received word in a sphere is decoded as the codeword which is the center of that sphere. If t or fewer errors occur, then the received word is always in the proper sphere and the decoding will be correct. Therefore, we call t the error-correcting ability , which is determined by the minimum dis-

CH 3006-4/91/0000 – 1200 $1.00 © IEEE

tance of a code.

For a cut-set code associated with the graph $\hat{G} = (\hat{V}, \hat{E})$, the minimum distance is the number of edges in the cut set with the fewest edges. In order to have distance d, every vertex in the graph must have degree $\geq d$. Since an edge is incident on two vertices, a limit on the minimum distance of a GTC can be thus derived as: $d \leq 2|\hat{E}|/|\hat{V}|$.

The encoding problem of a cut-set code $C_{\hat{G}}$ is to find the generator matrix G. From the fact that every row in $A_{\hat{G}}$ is a cut set and any $\mid \hat{V} \mid -1$ rows of $A_{\hat{G}}$ form a basis of the cut set space of $\hat{G}$, we can esaily get the generator matrix G.

On the other hand, the MLD problem is known to be NP-hard [8], and the hardness remains even with preprocessing or other heurestics [9]. However, the following two propositions give us another point of view about the MLD of cut-set codes. The proofs can be found in [1] and, hence, will be omitted here.

Proposition 1 *Let $\hat{G} = (\hat{V}, \hat{E})$ be a unweighted connected graph where $\hat{V}$ and $\hat{E}$ respectively denote the set of vertices and the set of edges. Let $C_{\hat{G}}$ be the cut-set code associated with $\hat{G}$, and let Y be a vector (over $\{0,1\}^{|\hat{E}|}$) to be decoded . Based on the same graph, assign weights to the edges of $\hat{G}$ as follows:*

$$W_i = -1^{Y_i}. \tag{1}$$

and denote the weighted graph $\hat{G}_Y$, where Y=[$y_1, y_2, \cdots, y_{|\hat{E}|}$] and W_i is the weight of the ith edge in G. Then the MLD of Y with respect to $C_{\hat{G}}$ is equivalent to finding a minimum cut in $\hat{G}_Y$.

Example 1: The incidence matrix of the 5-vertex , 8-edge graph $\hat{G}$ shown in Fig.1 is $A_{\hat{G}}$, and the cut-set code $C_{\hat{G}}$ is a [8,5,3] code. Let the vector Y to be [00100001], then the MLD of Y is equivalent to finding the minimum cut of the weighted graph $\hat{G}_Y$ shown in Fig. 2.

$$A_{\hat{G}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

2.2 Background on Neural networks

A discrete-time Hopfield neural network N with n neurons is uniquely specified by a pair (W, T), where W is the connection matrix which is symmetric and zero-diagonal with each entry w_{ij} denoting the connection strength between neuron i and neuron j, and T is a vector with T_i standing for threshold value associated with the neuron i.

Let $V(t) = [v_1(t), \ldots, v_n(t)]^T$ represent the state of the neural network at time t, where $v_i(t)$ is the state of the ith enuron, with the value either +1 or -1. The state evolution is according to the following :

$$v_i(t+1) = \begin{cases} 1, & \text{if } \sum_{j=1}^{n} w_{ji}v_j(t) - T_i \geq 0 \\ -1, & \text{otherwise} \end{cases} \tag{2}$$

Usually, there will be an energy function $E(t)$ defined as follows:

$$E(t) = -V^T(t)WV(t) + 2V^T(t)T, \tag{3}$$

to illustrate the possible situations of the neuron state after the state convergence of the network. These possible situations generally correspond to local minima of the energy function E [2].

The following result explores the relation between a minimum cut in the graph associated with the linear code as well as the code vector and the energy (local) minima of a suitably constructed Hopfield neural network. Together with Proposition 1, we now get the full picture of how the equivalence relation between the problem of MLD of GTC's and that of searching for a neuron state of a Hopfield neural network at which the energy reaches the minimum.

Proposition 2 *Given a weighted graph G_Y associated with some code vector Y, construct a Hopfield neural network such that its structure is a counterpart of G_Y and, in addition, the threshold vector T is identically zero, i.e. , $T \equiv 0$. Then the problem of searching at a neuron state V for which the energy function E reaches the minimum is equivalent to that of finding a minimum cut of the graph G_Y.*

Example 2 : The connection matrix W of the Hopfield neural net N corresponding to the weighted graph $\hat{G_Y}$ in Example 1 is :

$$W = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 & -1 \\ 1 & 0 & -1 & 0 & 1 \\ 1 & 1 & -1 & 1 & 0 \end{bmatrix}$$

The global minima of N are state $V = (1,1,-1,1,1)$ and state $\bar{V}$ = (-1,-1,1,-1,-1). On the other hand, the minimum cut of $\hat{G}_Y$ is $\{e_2, e_3, e_8\}$, or [01100001] in a vector form. The equivalence relation is easily observed if we divide the vertex set of $\hat{G}$ according to the value of the neuron state.

3 Decode Graph-Theoretic Codes

3.1 Decoding Algorithm 1

Generally, to apply the Hopfield model to any optimization problem, we need to map the original problem to a Hopfield netowrk. In short, we first have to derive an energy function E such that the desired solution occurs at a minimum of E, and then derive the connection matrix from the energy function. Next, randomly choose a reasonable initial state, and let the network iterate until it converges to a stable state. At last, interpret the stable state, through energy function mapping relation, as a solution of the original problem.

The propositions shown in the previous section have provided a mapping between the energy function of a Hopfield net to the MLD problem of a GTC. What now left for us is how to interpret the stable state into a cut-set codeword. Let V_1 be the set of neurons with state +1, and V_2 be the set of neurons with state -1.Obviously, whether edge e_i is included in the cut set can be determined by the inner product between the final state V and the column vector of incidence matrix $A_{\hat{G}}$ representing edge e_i. However, we need to transform the domain of neuron states from $\{1, -1\}$ to $\{1, 0\}$, since the vector space of graph is over GF(2).

The following procedure shows how to map MLD problems of GTC onto Hopfield neural networks.

Procedure :

Goal Construct a Hopfield neural network to solve an MLD problem.

Input : A Graph $\hat{G}$ and a code vector Y.

1. Construction Stage :

- Construct the weighted graph $\hat{G_Y}$ according to equation (2).
- Construct a Hopfield neural network as a counter part of the graph $\hat{G_Y}$ and is specified by the pair (W, T) where $T \equiv 0$ and $W = [w_{ij}]$ is defined as :

$$w_{ij} = \begin{cases} 0 & \text{if no edge connects vertices } i \text{ and } j \text{ corresponding the neuron } i \text{ and } j \text{ respectively} \\ -1 & \text{if } e_k \text{ connects vertices } i \text{ and } j \text{ corresponding the neuron } i \text{ and } j \text{ respectively and } y_k = 1 \\ 1 & \text{if } e_k \text{ connects vertices } i \text{ and } j \text{ corresponding the neuron } i \text{ and } j \text{ respectively and } y_k = 0 \end{cases} \tag{4}$$

2. **Convergence Stage :**
   - Randomly select an initial state and wait for the network to converge to a stable state, say V.

3. **Interpretation Stage :**
   - Map the stable state into a cut-set vector of $\hat{G_Y}$ which is then the decoded codeword of Y.
     1. Transform the stable state $V \in \{1,-1\}^{(k+1)}$ into a new state $V' \in \{1,0\}^{(k+1)}$ according to the following :
     $$\hat{v}_i = \begin{cases} 0 & \text{if } v_i = 1 \\ -1 & \text{if } v_i = -1 \end{cases} \tag{5}$$
     2. Use the incidence matrix $A_{\hat{G}}$ of graph $\hat{G}$ to obtain the cut-set vector M associated with V' by :
     $$M = V' \cdot A_{\hat{G}} \tag{6}$$
     which is also the decoded codeword.

4. **End.**

Obviously, the performance of this Hopfield net based decoder depends on how many percentages the net converges to its global minima. The following proposition is a loose estimation on the convergence behavior.

Proposition 3 *For an n-neuron Hopfield net N operating in discrete time and in asynchronous mode, the probability that the net will converge to a state whose energy is an global minimum is at least* $\frac{1}{2^{n-2}}$.

Proof : Let $V = [v_1, \ldots, v_n]$ be a global minimum state of an n-neuron Hopfield net, and denote the n neighbor state of V as $V^i = [v_1, \ldots, \bar{v_i}, \ldots, v_n]$, where $\bar{v_i} = 1$ if $v_i = -1$ and $\bar{v_i} = -1$ otherwise, $(i = 1, \ldots, n)$. Assume the probability distribution over the selection of initial state and next state is *linear*, then the probability for any state to be chosen as an initial state is $\frac{1}{2^n}$, and every V^i has the probability at least $\frac{1}{n}$ to evolve to state V, since $E(V)$ is global minimum and

$$E(V^i) - E(V) = (v_i - \bar{v_i})(\sum_{j=1}^{n} w_{ij} v_j) > 0$$

. Hence, the probability to converge to a global minimum state V is at least $\frac{1}{2^n} + n \times \frac{1}{2^n} \times \frac{1}{n} = \frac{1}{2^{n-1}}$. Moreover, the symmetry of the connection matrix leads to the symmetry over the distribution of the energy, that is, state $\bar{V} = [\bar{v_1}, \ldots, \bar{v_n}]$ is also a global minimum of N. Therefore, the probability that an n-neuron Hopfield net will converge to a global minimum is at least $2 \times \frac{1}{2^{n-1}} = \frac{1}{2^{n-2}}$.

3.2 Decoding Algorithm 2

Local minima is a fatal problem for all the applications of Hopfield Neural Networks. For decoding a linear block code, incorrect decoding caused by local minima is not accpetable. To put a neural network based decoder into practice, we have to circumvent this defect. Since the landscape of the energy function is determined during the construction stage, and, hence, the convergence behavior totally depends on the selection of initial states, we can attach a detection stage to the original decoding system which is responsible for detecting whether a received word is correctly decoded or not. Moreover, a feedback flow is connected from dectection stage to convergence stage to allow another trial, if detected as fail(incorrect). A parameter m must be set to decide the maximum number of the trails.

Now the problem turns to how to detect the correctness of decoding. We first introduce the concept of complete decoders and incomplete decoders : A complete decoder decodes every received word as a closed codeword, while an imcomplete decoder decodes only those received words lying in a decoding sphere about a codeword. When more than t errors occur, the complete decoder will often decode incorrectly but occasionally will find the correct codeword. . Therefore, we turn to adapting an incomplete decoder for our modification. If the Hamming distance $d(M,Y) > t$, detection stage treats it as failure, for the received word Y is not lying in the sphere of codeword M. There are two possible causes : either the errors are too many to be corrected, or the decoder is stuck at some local minimum. The latter case may eventually finds a successful solution for sufficient trials cohereas the former case , however, will never succeed in the detection stage. The detailed description of the detection stage is shown below, and the system flowchart of the modified neural network based decoder is shown in Fig. 3 : **Detection Stage :** Detect wether the decoded codeword is correct or not

- If $d(M,Y) \leq t$: succeed and go to an end.
- If $d(M,Y) > t$ and trial number exceeds m : fail and go to an end.
- If $d(M,Y) > t$ and trial number does not exceed m : retry and go to convergence stage.

Proposition 4 *The modified decoding system will correctly decode those received words with no more t errors in possibility one.*

Proof : Let p be the possibility that a Hopfield net will converge to a global minimum. The possibility that the net will not converge to a global minimum within m trials is $1 - (p + (1-p)p + (1-p)^2 p + \cdots + (1-p)^{m-1} p) = (1-p)^m$. As m approaches to infinite, $(1-p)^m$ approaches to zero, since $p \geq \frac{1}{2^{n-2}}$. Therefore, we conclude that the net is guaranteed to converge to global minima in possibility one.

4 Experiments And Discussions

4.1 Decoding Algorithm 1

Before the experimental results are shown, let us discuss why the convergence probability in Proposition 5 is under-estimated in two ways :

1. The probability is under-estimated because only those states which are neighbors to global minima are counted. The fact is, however, that any state which may evolve to any neighboring state of a global minimum also has the probability to converge to global minimum.
2. Let $Next(V)$ represent the set of those neighboring states of V with energy not greater than that of state V, that is, $Next(V)$ is the set of all possible next states of V. Obviously, the probability to select any next state is inversely proportional to the size of the set $Next(V)$, which may be extremely greater than $\frac{1}{n}$ for many states.

To sum up, the convergence behavior of this neural network based decoder is, in fact, quite promising.

Experiments were repeated for every graph to test how many percentages this neural-decoder can correctly decode all the received words within the correcting ability of its corresponding graph theoretic code. Experimental results are shown in Table 1, where, 10 out of 14 GTCs are free from local minima problems, that is, every received words within the error-correcting ability are all decoded correctly, while the other 4 GTC's has at least 72.3 % successful decoding rate. In comparison with those satisfying convergence behaviors of many Hopfield applications, the result shown here appears to be much better which encourages the further modification with a detection and a feedback mechanism.

Of course, these graphs are not the most general ones. We have been trying to find the reasons why this kind of Hopfield network application has such satisfactory behavior, but failed to find any theoretic evidence. However, by our experience, we believe that it is because the connection strength is limited to 1,-1 and 0 so that the overall convergence behavior of any graph is far better than any other Hopfield application whcih have appeared.

4.2 Decoding Algorithm 2

Parameter m, the maximum number of trials allowed, plays the same important role to our modified decoding system as what a *timer* plays to a distributed network system. We have to choose m so appropriately as to guarantee that any received word within the error-correcting ability of a GTC will converge to its corresponding global minimum with no more m trials. Then those received words which will not converge within m trials is treated as the ones which out of the error-correcting ability and thus should be rejected by the decoding system. If the value of m is not large enough, received word might be rejected just because insufficient trials allowed, even though the received words is within the correcting ability. On the other hand, reliability will be promoted as m getting larger at the expense of slower performance of the overall system . Table 2 lists the theoretic values of m on different probability p to get a 99 % successful decoding rate. We experimented on those graphs in Table 1 with correct decoding rate (CDR) which is less than one, and found that with at most 6 trials can the received words within the correcting ability can be decoded correctly.

5 Conclusion

This paper presented a neural network approach for solving maximum likelihood decoding problems of graph theoretic codes. Simulation results showed that the original Hopfield network can decode GTCs with successful rate over 70 %, far better than any other Hopfield applications. Moreover, the modified decoding system with detection mechanism can circumvent the local minima problems, decoding the received words within the error-correcting ability in probability one. To complete this coding system, further investigations are required concerning how to construct graph theoretic codes as desired and efficiently encode a GTC in a systematic form. Besides, how to apply neural networks to decode other linear block codes i< also a topic of our future research.

References

[1] J. Bruck and M. Blaum, "Neural Networks, Error-Correcting Codes, and Polynomials over the Binary n-Cube." *IEEE Trans. Inform. Theory*, vol. 35, pp. 976-987. Sept 1989.

[2] J.J.Hopfield and D.W.Tank, "Neural Computation of Decisions in Optimization Problems," *Biol. Cybernet.* 52,(1985), pp.141-152.

[3] N. Deo, *Graph theory with Applications to Engineering and Computer Science.* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1974.

[4] W.W.Peterson and E.J.Weldon, JR. *Error-Correcting Codes.* 2nd ed. MIT. press. 1972.

[5] R.P.Lippmann,"An introduction to Computing with Neuralnets," *IEEE ASSP Magazine*, Apr 1987.

[6] B.Shirazi and S.Yih, Critical Analysis of Applying Hopfield Neural et Model to Optimization Problems ."

[7] S.L.Hakimi and H.Frank, "cut-set matrices and linear codes," *IEEE Trans. Inform. Theory*, vol, IT-11, pp.457-458, July 1965.

[8] E.R.Berlekamp, R.J. Mcliece, and H.C.A. van Tilborg, " On the inherent intractability of certain coding problems," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 384-386, May 1978.

[9] J.Bruck and M.Naor,The hardness of Decoding Linear Codes with Preprocessing," *IEEE Trans. Inform. Theory*, vol. 36, no. 2, Mar 1990.

[10] J.J.Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci.* (USA) vol. 79, pp. 2554-2558. 1982.

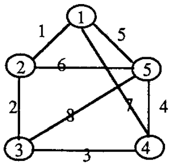


Fig. 1

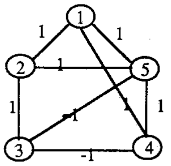
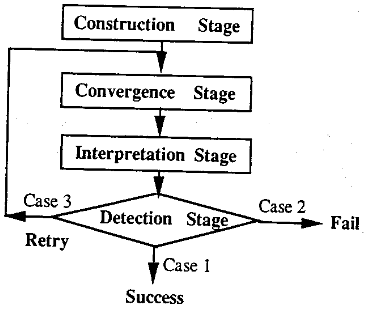


Fig. 2



Case 1 : d(M,Y) <= t
Case 2 : d(M,Y) > t and trial number exceeds m
Case 3 : d(M,Y) > t and trial number does not exceed m

Fig. 3 Decoding flowchart

Graph	[n,k,d]	CDR	Graph	[n,k,d]	CDR
	[5,3,2]	1	K3,6	[18,8,3]	1
	[6,3,3]	1		[14,5,4]	1
	[8,4,3]	1		[9,4,3]	1
K_5	[10,4,4]	1		[12,5,4]	0.844
$K_{3,3}$	[9,5,3]	1		[12,7,3]	0.723
$K_{3,4}$	[12,6,3]	1	peterson	[15,9,3]	0.824
$K_{3,5}$	[15,7,3]	1		[20,7,5]	0.898

n : Codeword length k : Information length d: the minimum distance
CDR : Correct Decoding Rate

Table 1. The convergence behavior of the neural networks for decoding the cut-set codes.

p	m
0.1	44
0.2	21
0.3	13
0.4	9
0.5	7
0.6	6
0.7	4
0.8	3
0.9	2

p : the CDR in a given trial

m : the maximum number of trials needed

Table 2. Theoretical Values of m