# Coded Hopfield Networks

Claude Berrou and Vincent Gripon
Telecom Bretagne, Electronics Department
UMR CNRS Lab-STICC
Brest, France
name.surname@telecom-bretagne.eu

*Abstract*— **Error-correcting coding is introduced in associative memories based on Hopfield networks in order to increase the learning diversity as well as the recall robustness in presence of erasures and errors. To achieve this, the graph associated with the classical Hopfield network is transformed into a bipartite graph in which incoming information is linked to orthogonal or quasi-orthogonal codes. Whereas learning is similar to that of classical (i.e. Hebbian) Hopfield networks, memory retrieval relies on error correction decoding which offers strong discrimination properties between the memorized patterns.**

***Keywords*- Hopfield network, error-correction coding, assiociative memory, spread spectrum**

## I. INTRODUCTION

Since the celebrated Dartmouth conference in 1956, organized on John McCarthy's initiative, artificial intelligence and its potential applications have continuously aroused the interest of many scientists. One of the participants of this conference was Claude Shannon, the father of information theory, open to many and various aspects of science. Surprisingly, they are rare today the theoreticians or practitioners of information, in the sense of Shannon's approach, who are interested in artificial intelligence or in its ramifications. And yet information is the fundamental substance of living or artificial systems that learn, communicate and decide. This low level of involvement is all the more astonishing since biologists are still largely unable to explain how information is represented, stored and processed in the neocortex. Despite all the efforts carried out these last twenty years in the exploration of the brain, thanks to more and more sophisticated tools (EEG, MRI,…), the neocortex, from the point of view of information, remains *terra incognita*.

Regarding the information theory, whose developments have long been solicited and captured by the needs of telecommunications, in perpetual demand for improvements, considerable progress has been achieved in the representation, protection, transportation and interpretation of information. In particular, recent years have seen the emergence of new methods that rely on probabilistic message passing within multicellular machines. Each cell is designed so as to process an elementary problem in an optimal way and the exchange of information between all cells leads to an optimal global outcome. Turbo decoding [1] has paved the way for this kind of distributed approach. Turbo decoding was then recognized as an instance of the very general principle of belief propagation [2], which was later found another important application in the decoding of Low Density Parity Check (LDPC) codes [3,4]. The turbo principle and belief propagation are more general than mere error correction techniques and their applications have been extended to demodulation, detection or equalization, for instance. From this point of view, information processing in a receiver is moving closer to how the neocortex performs its mental operations, that is, through a multidirectional exchange of locally produced messages.

The functional similarities between a modern error correcting decoder and the neocortex are many: distributed structure, message passing processing, high level of separability between pieces of information, resistance to noise, resilience, etc. In particular, the realities of a unique fixed point of decoding in the distributed decoder and of a unique thought in the biological neural network, both among an astronomic number of possibilities, invite us to revisit the neural computation field with the help of error correction concepts.

This article presents a concrete example of formal neural networks combined with error correcting codes. This example is a simple one as it is based only on Hopfield networks and is not intended to be fully representative of the role that coding can play in complex neural networks. However, the results that are presented here seem significant enough to justify and arouse further studies.

## II. BIPARTITE HOPFIELD NEURAL NETWORKS

A Hopfield neural network (HNN), an example of which is drawn in figure 1, is carried by a complete, undirected, loopless and weighted graph [5,6]. This graph has $n$ nodes (neurons) and $\frac{n(n-1)}{2}$ links. The bidirectional link between node $i$ and node $j$ is characterized by the (synaptic) weight $w_{ij}$. This weight results from the learning of $M$ messages of $n$ binary antipodal ($\pm$ 1) values: $\{d_i\}$, $i$ = 1,…,$n$, the particular values $d_i$ and $d_j$ being assigned to the $i^{\text{th}}$ and $j^{\text{th}}$ nodes:

$$w_{ij} = \sum_{m=1}^{M} d_i^m d_j^m \qquad (1)$$

This weight may take $P = M + 1$ values. The recalling of a particular message, from a part of its content, is performed through the iterative process described by the following equations, where $v_i^p$ is the outcome value of $i^{\text{th}}$ neuron after $p$ iterations:

$$v_i^p = +1 \quad \text{if} \quad \sum_{\substack{j=1 \\ j\neq i}}^{n} w_{ij} v_j^{p-1} \geq 0$$

$$v_i^p = -1 \quad \text{if} \quad \sum_{\substack{j=1 \\ j\neq i}}^{n} w_{ij} v_j^{p-1} < 0 \qquad (2)$$

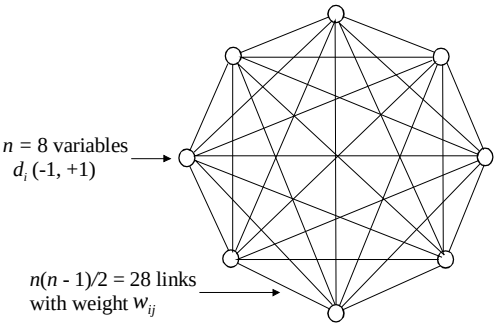


Figure 1. Classical Hopfield Network with $n = 8$ neurons.

According to [7], an upper bound on the learning diversity (*i.e.* the number of messages), conditionally to error-free recalling, is:

$$M_{\max} = \frac{n}{\log(n)} \qquad (3)$$

where log(*n*) is the natural logarithm. Since the messages have length *n*, an upper bound on the binary capacity $C_b$ is:

$$C_{b,\max} = nM_{\max} = \frac{n^2}{\log(n)} \qquad (4)$$

The quantity of binary information $Q_b$ that the HNN requires in order to store *M* messages is:

$$Q_b = \frac{n(n-1)\log_2(M+1)}{2} \approx \frac{n^2 \log_2(M+1)}{2} \qquad (5)$$

The storage efficiency $\eta = \frac{C_b}{Q_b}$ is then upper-bounded by:

$$\eta_{\max} = \frac{C_{b,\max}}{Q_b} = \frac{2n^2}{n(n-1)\log(n)\log_2(M_{\max}+1)} \approx \frac{2}{\log(n)\log_2(\frac{n}{\log(n)}+1)} \qquad (6)$$

The storage efficiency of HNN is fairly poor (*e.g.* $\eta_{\max} \approx 5\ 10^{-2}$ for $n = 100$) and tends to 0 when *n* tends to infinity.

Let us propose a simple way to combine error correcting coding and HNN principles through a bipartite graph, as represented in figure 2. This graph links the message and codeword contents through weights $w_{ij}$ resulting from learning, like in HNN. Such a scheme has the great advantage of making independent the size of the network and the length of the messages. This gives us an additional degree of freedom to improve the learning diversity of the scheme.

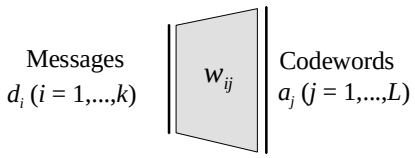


Figure 2. Bipartite coded Hopfield Network.

The left part of the graph is associated with real messages $\{d_i\}$, $i = 1,\ldots, k$ and the right part with orthogonal or quasi-orthogonal codewords $\{a_j\}$, $j = 1,\ldots, L$ where $a_j = \pm 1$. Each new incoming message $\{d_i\}$ is associated with a new codeword $\{a_j\}$, different from the previous ones. The equations of the network are the following, in which *M* is the number of learnt messages, μ is anyone of these, *i* and *j* are the left and right indices respectively. *s* is a particular value of *i* or *j*. Finally, *R* is the coding rate, which is quite low for practical values of *M* and *L*.

$$w_{ij} = \sum_{m=1}^{M} d_i^m a_j^m \qquad (7)$$

$$\sum_{i=1}^{k} w_{is} d_i^\mu = \sum_{\substack{m=1 \\ m\neq\mu}}^{M} a_s^m \sum_{i=1}^{k} d_i^m d_i^\mu + k a_s^\mu \qquad (8)$$

$$\sum_{j=1}^{L} w_{sj} a_j^\mu = \sum_{\substack{m=1 \\ m\neq\mu}}^{M} d_s^m \sum_{j=1}^{L} a_j^m a_j^\mu + L d_s^\mu \qquad (9)$$

$$R = \frac{\log_2(M)}{L} \qquad (10)$$

Equations (7) and (9) are reciprocal of Code Division Multiple Access (CDMA) equations for *M* users and spreading sequences of length *L*, if we liken weights $w_{ij}$ to the sum, at time *i*, of the *M* users signals in the chip of index *j*. If the codewords are perfectly orthogonal (Walsh-Hadamard sequences) or slightly non orthogonal (pseudo noise sequences with normalized cross correlation equal to 1/*L*) and as far as $M \leq L$, the content of any message μ can be recovered with a high probability in presence of erasures, to a certain amount.

The selection of message μ is performed by Maximum Likelihood (ML) decoding from the estimates of $a_s^\mu$, $s = 1,\ldots,L$ provided by relation (8). Strictly speaking, the scheme of figure 2 can no longer be called a Hopfield network, as the decoding does not rely on equations (2), but learning rules share the same principle of superposition. Note that relations (7) to (9) authorize any type of value (real, integer, binary) for $\{d_i\}$. In the sequel, we will consider only binary values.

For example, with $k = 40$ and $L = 256$, the coded Hopfield network is able to learn 256 messages of 40 binary values. Figure 3 gives the recall performance of the coded network with such parameters, when random erasures occur in $\{d_i\}$. Up to an erasure rate of 50%, the network is able to recover any message with very high probability. By comparison, the performance of the classical HNN having roughly the same number of informational weights ($\approx 10^4$), that is $n \approx 150$, is

given. We can observe that the network hardly accepts up to 25 messages, even without any erasure. Learning diversity is then considerably increased when error correcting coding is added to the neural network, in the same way as communication systems are considerably improved using channel coding. Of course, this comparison does not take into account the complexity of the decoder (whose structure and connections are established once for all and do not depend on messages). Section III describes a way to implement ML decoding using formal neurons.

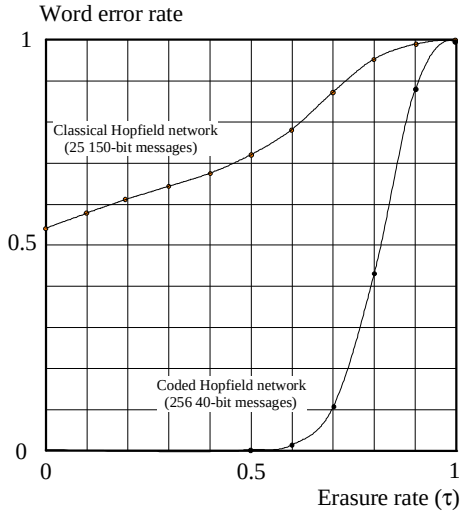


Figure 3. Compared performance of the coded Hopfield network with $k = 40$ and $L = 256$ and the classical HNN with $n = 150$ (that is, roughly with the same number of weights). The erasure rate $\tau$ is the proportion of randomly erased values in the message to recover. The word error rate is the proportion of messages not correctly recovered (at least one value is wrong).

Where does the diversity gain really come from? Relation (5) gives the amount of binary information that a HNN has stored after the learning of $M$ messages. The capacity of the network (the number of messages multiplied by their length) cannot surpass $Q_b$. If the messages have length $n$, like in the HNN, a strict upper bound on $M$ is $\frac{n}{2}\log_2(P)$, independently of any other consideration. If, by an appropriate means, the length of messages is limited to a value $k$ less than $n$, this upper bound becomes $\frac{n^2 \log_2(P)}{2k}$. Thus, the upper bound is linear in $n$ if full length is targeted and quadratic in $n$ if messages have fixed shorter length. As for the capacity, it remains unchanged. As already mentioned, the scheme of figure 2 is a way to make $n$ and $k$ independent, and then to allow and control small values of $k$. This sparsity in data representation, added to the discrimination capacity of orthogonal codes, explains the strong diversity gain of coded Hopfield networks compared to HNN.

## III. Neural Implementation of ML decoding

Generally speaking, ML decoding relies on a bipartite graph linking the content and activity of codewords. With received data $\{x_j\}$, $j = 1,\ldots,J$, which are real in the general case, are associated $J$ neurons with real values $\{y_j\}$. On the other side of the graph, $Q$ neurons called *fanals* with binary values (0, 1) $\{u_q\}$, $q = 1,\ldots,Q$ materialize the $Q$ possible codewords. The edges of the graph have binary weights ($\pm$ 1) $t_{jq}$. An example is given in figure 4 for six 4-bit codewords. The decoding operations are given by the following equations, which take into account a possible message passing procedure at a higher level of processing:

Initialisation:

$$y_j^0 = 0 \quad j = 1,\ldots,J$$

$$y_j^1 = x_j \quad j = 1,\ldots,J \tag{11}$$

At iteration $p$ ($1 \le p \le p_{\max}$):

$$z_q^p = \sum_{j=1}^{J} t_{jq}(y_j^p + \gamma y_j^{p-1}) \quad q = 1,\ldots,Q \tag{12}$$

$$z_{\max}^p = \max\{z_q^p\} \tag{13}$$

$$u_q^p = 1 \text{ if } z_q^p = z_{\max}^p \text{ and if } z_{\max}^p > \sigma$$

$$u_q^p = 0 \text{ otherwise} \tag{14}$$

$$v_j^p = \sum_{q=1}^{Q} t_{jq} u_q^p \tag{15}$$

$$y_j^p = 1 \text{ if } v_j^p > 0$$

$$y_j^p = -1 \text{ if } v_j^p < 0$$

$$y_j^p = 0 \text{ if } v_j^p = 0 \tag{16}$$

$\gamma$ is a memory parameter which allows us to keep, at rank $p$ of the iterative process, a fraction of the results obtained at rank $p - 1$. This memory effect is essential when several codes are combined in a composite network but has not to be exaggerated to avoid the persistence of errors. When just one code is considered, there is no global iterative decoding and the memory effect has obviously no interest.

Note that equations (13) and (14) authorize several fanals to be activated (*i.e.* $u_q^p = 1$ for different $q$), this being possible when one or several inputs $x_j$ are erased.

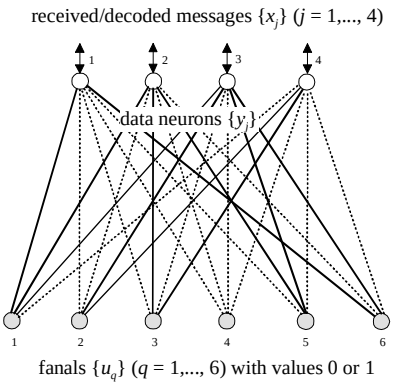


Figure 4. Bipartite graph for the encoding/decoding of $Q = 6$ codewords of $J = 4$ bits (+1+1+1-1,-1-1+1+1,-1+1-1+1, -1-1-1-1,-1+1+1-1,+1-1+1-1). Full lines correspond to value +1, dashes to -1.

$\sigma$ is the activation threshold of fanals. To perform a true ML decoding, $\sigma$ must be $-\infty$ but, depending on the application, a finite value may be given to $\sigma$, that is, fanals must satisfy a lower bound of activity to be taken into account. For instance, having $\sigma = 0$ and all inputs erased, the condition $z_{\max}^{p} > 0$ of (14) keep all fanal values to zero. So, this algorithm performs a kind of soft-output decoding, able to process totally or partially erased incoming messages.

A way to implement the maximum function with neurons comes from the following equivalence, where *A* and *B* are two real numbers:

$$\max(A,B) = \frac{A+B}{2} + \left|\frac{A-B}{2}\right| \qquad (17)$$

Using neurons with transfer function:

$$output = \max(0, \Sigma\, input) \qquad (18)$$

this equivalence may be implemented by the circuit of figure 5 for any value of *A* and *B*, provided that at least one is positive.

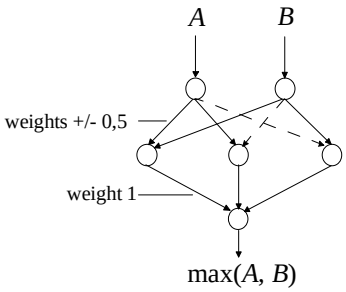


Figure 5. Neural implementation of the maximum function for two numbers *A* and *B* (one at least being positive), using neurons having transfer function *output* = max(0, *input*). Full lines correspond to positive weights, dashes to negative ones.

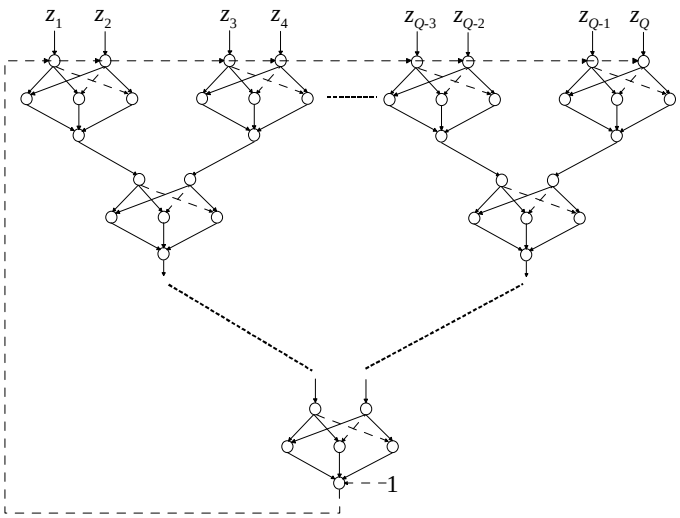


Figure 6. Neural implementation of the maximum function for *Q* (a power of 2) values, at least one of them being positive. The feedback loop assures that only neurons which have maximum incoming signals remain active, with output equal to 1

By cascading the circuit of figure 5 into a succession of maxima researches, the maximum function can be implemented for any number of inputs provided that at least one value is positive. Figure 6 depicts such a scheme for *Q*, a power of two, values. The maximum value stemming from the selector is lowered by 1 and the result is used as an inhibitory input for all neurons of the first layer. Therefore, only neurons which have maximal incoming signals remain active, with output equal to 1.

## IV. INCREASING THE LEARNING DIVERSITY

The learning diversity of the network in figure 2 is limited by the properties of equation (9). For $M > L$, the codewords are no longer orthogonal and errors creep into the estimates of $d_s^{\mu}$, which are not elements of a code and cannot be corrected.

In order to increase diversity beyond *L* while keeping the bipartite model of figure 2, we need to orthogonalize the messages (and then to make them corrigible) with the aid of small bipartite encoders, also similar to the model of figure 2. The incoming binary (antipodal) message of length *k* is segmented in *B* blocks of length $\kappa = k/B$ and with each block is associated a set of $l = 2^{\kappa}$ orthogonal (Walsh-Hadamard) codewords of length *l*. Furthermore, since it is no longer possible to have perfectly orthogonal codewords $\{a_i\}$ of length *L* when $M > L$, these will be simply obtained by random drawing. Their mean Hamming distance is then *L*/2 with a standard deviation $\sqrt{L}$ (binomial law).

Figure 7 gives an example of such a construction for $k = 24$ and $B = 4$ ($\kappa = 6$). The decoding of local and global codes relies on equations (11) to (16), in which $J = Q = 64$ for local codes and $J = L$ and $Q = M$ for the global code.

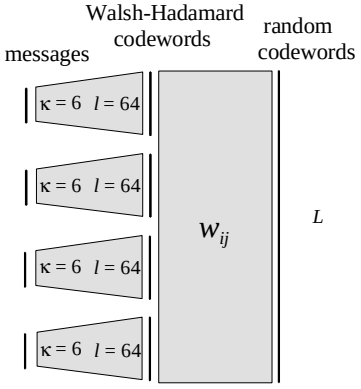


Figure 7. Architecture of a coded Hopfield network with segmented input. In this example, the incoming binary messages of length $k = 24$ are segmented in $B = 4$ blocks, each addressed by $\kappa = 6$ bits. For every block, the $l = 2^{\kappa} = 64$ possible values of the sub-message are associated with as many Walsh-Hadamard codewords of length 64. Then, the 256 binary values obtained from the concatenation of these codewords are linked to random codewords of length *L* through a bipartite coded Hopfield network with weights $w_{ij}$.

Figure 8 gives the result of simulations for the composite network of figure 7, for three values of *L*: 256, 512 and 1024. *M* messages are learnt and then recalled whereas one of the four segments is not provided with input information. In this experiment, we have $\gamma = 1/8$ and $\sigma = 0$. The maximum number of iterations is $p_{\max} = 2$, the transfer of information between local decoders and the global decoder being performed synchronously.

We can observe that learning diversity can significantly go beyond *L* (about 3 times the value of *L* for a word error rate of

$10^{-1}$). Compared to HNN (having roughly the same number of weights as the coded network with $L$ = 512), diversity is multiplied by 22, 45 and 75 for the three increasing values of $L$ and still with a word error rate of $10^{-1}$. As explained at the end of section II, these strong gains on diversity come both from the representation of messages with a limited number of bits and from the discrimination capability of the different decoders. As for capacity, the gains are not so large (between 2 and 3) and $\eta_{max}$ is not significantly increased, compared to the value given by (6), because $P$, the number of possible levels for weights $w_{ij}$ has slightly increased.

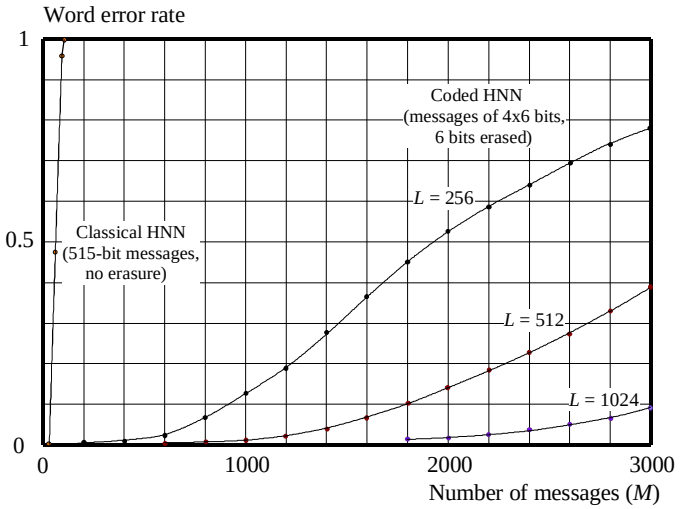


Figure 8. Compared performance of the coded Hopfield network of figure 7 ($k$ = 4x6 bits and $L$ = 256, 512 and 1024) and a classical HNN with $n$ = 515 (that is, roughly with the same number of weights as the coded network with $L$ = 512). The word error rate is the proportion of messages not correctly recovered (at least one value is wrong). There is no erasure in the case of HNN whereas one of the four blocks is not provided with information in the case of the coded network. For a word error rate of $10^{-1}$, the coding gain in diversity is about 22, 45 and 75 for the three values of $L$.

## V. Conclusions and Perspectives

The results presented in this paper illustrate the role that error correcting codes, combined with sparse data representation, can play in neural networks. The gains in learning diversity are considerable and increase quasi-linearly with the length $L$ of the global code in composite schemes similar to that of figure 7. For a given and fixed value $k$ of the message length, this also means that the diversity grows linearly with the number of informational weights while the diversity in classical HNN grows in proportion of the square root of this number, and even less, considering the denominator in relation (3). In other terms, the coded network diversity follows a quadratic law with the number of neurons instead of a linear law for the classical HNN, as was anticipated in section II.

If the long term perspective is to design and build machines that behave as the cortex, diversity is a much more important parameter than capacity. The number of pieces of information counts more than their sizes. From a cognitive point of view, it is better to learn (and possibly combine) 1000 messages of 10 characters than to learn 10 messages of 1000 characters!

The comparison we have made between classical and coded HNN does not take into account the complexity of the different decoders, especially the global decoder which has to consider $M$ codewords of length $L$, that is $ML$ binary connections. Though the structure and connections of these networks are established once for all, independently of the messages to learn and recall, it is unrealistic to contemplate circuits based on the principles described in section III, in particular the too well structured circuit depicted in figure 6, which is not biologically plausible. Nonetheless, the recent developments in coding theory taught us that the ML decoding of long codes is not insurmountable if distributed coding is considered. Then, the question that is asked now is: is it possible to replace a unique neural decoder having to process $M$ messages of length $L$ with a small number of elementary neural decoders handling $M' \ll M$ messages of length $L' \ll L$ (like a turbo product decoder, for instance)? If the answer is positive, then we would be able to design realistic neural networks offering both very large learning diversity and robust recalling.

## References


[1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes", *Proc. of IEEE ICC '93*, Geneva, pp. 1064-1070, May 1993.

[2] R. J. McEliece, D. J. C. MacKay and J.-F. Cheng, "Turbo decoding as an instance of Pearl's 'belief propagation' algorithm", *IEEE Journal on Selected Areas in Commun.*, vol. 16, no. 2, pp. 140-152, Feb. 1998.

[3] R. G. Gallager, "Low-density parity-check codes", *IRE Trans. Inform. Theory*, Vol. IT-8, pp. 21-28, Jan. 1962.

[4] D. J. C. MacKay and R. M. Neal, "Good codes based on very sparse matrices," in *Cryptography and Coding 5th IMA Conf.*, C. Boyd, Ed., *Lecture Notes in Computer Science*, no. 1025. Berlin, Germany, Springer, pp. 100–111, 1995.

[5] J. J. Hopfield and D. W. Tank, "'Neural' computation of decisions of optimization problems," *Biol. Cybern.*, vol. 52, pp. 141–152, 1985.

[6] John J. Hopfield (2007) Hopfield network. Scholarpedia, 2(5):1977.

[7] R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh, "The capacity of the Hopfield associative memory," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 461–482, 1987.