Effect of Connectivity in Associative Memory Models

(preliminary version)

János Komlós Ramamohan Paturi
University of California, San Diego
La Jolla, CA 92093

Abstract

We investigate how good connectivity properties translate into good error–correcting behaviour in sparse networks of threshold elements. We determine how the eigenvalues of the interconnection graph (which in turn reflect connectivity properties) relate to the quantities, number of items stored, amount of error–correction, radius of attraction, and rate of convergence, in an associative memory model consisting of a sparse network of threshold elements or neurons.

0 Prologue

It has been a long–standing goal to understand the cognitive processes of biological systems in order to explain biological behavior and obtain insight in building artificial systems capable of cognition. Recently, several researchers in physics, cognitive psychology, electrical engineering and computer science [8], [22] have proposed *low–level* models consisting of networks of *neuron–like* elements to explain cognitive processes such as associative memory. Their studies emphasize the robust computational properties of an ensemble of interacting neuron–like elements which evolve using a learning mechanism. Even though these models are far–fetched abstractions of any biological system like the human brain, they can *plausibly* explain some of the cognitive processes from a computational point of view. On the other hand, the specificity and the simplicity of these models allows one to study rigorously the computational aspects of these models. More importantly, independent of biological realism, such a study would throw light on the nature of learning mechanisms in a reasonably powerful model like a network of neuron–like elements.

1 Introduction

Assume that a dynamical system has a large number of stable states with a substantial domain of attraction around them. That is, the system started at a state in the domain of attraction would converge to the stable state. We can then regard such a system as an *associative memory*. In this framework, stored items are represented by stable states, and attraction to them can be interpreted as retrieval of full information from partial information.

Assume, in addition, that the system is equipped with a *learning* mechanism, by which it adapts itself to remember new items.

We are interested in modeling and studying such systems. Since human brain is a primary example of an associative memory, the basic inspiration for modeling comes from neural systems. Below, we present the main ideas involved in the modeling process.

Following McCulloch and Pitts [17], we can model associative memory by an interconnected system of *abstract* neurons with binary activity. Learning in such a system takes place by means of synaptic changes. In fact,

0272-5428/88/0000/0138$01.00 © 1988 IEEE

Hebb [6] postulated that when a neuron repeatedly excites a second neuron, some lasting change takes place that increases the efficiency of the influence of the first neuron on the second. In addition, the studies of Lashley (see Lashley [14]) have shown that biological memory is highly distributed. Hence, it is reasonable to assume that interconnections among the neurons collectively encode information and that the synaptic interconnection strengths reflect the correlated activity at the corresponding neurons (Hebb's rule).

Models of memory based on these and other similar ideas have been studied by various researchers (see Hinton and Anderson [7], Kohonen [10] and Kohonen [11] for history and references), and, more recently, by Hopfield [8]. This is the model we are going to use in this paper. A closely related model is studied by Little [14], and Little and Shaw [15].

The model used by Hopfield consists of a system of *fully* interconnected neurons or linear threshold elements where each interconnection is symmetric and has a certain weight. Each neuron in the system can be in one of two states ± 1. The state of the entire system can be represented by an n-dimensional vector, where n is the number of neurons in the system. The components of the vector denote the states of the corresponding neurons. The weight of each interconnection is given by w_{ij} with $w_{ij} = w_{ji}$.

Each neuron updates its state based on whether a linear form computed by the weights of their interconnections and the current state is above or below its threshold value. We will assume in this paper that all thresholds are zero. Hence, with the system in state x, neuron i resets its state to $\mathrm{sgn}(\sum_{j \neq i} w_{ij} x_j)$ where the function sgn is defined as

$$\mathrm{sgn}(x) = \begin{cases} +1 & \text{if } x \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

We consider two modes of dynamic operation of the system. In the *synchronous* mode, at every time step, every neuron updates its state simultaneously. In the *asynchronous* mode, at any instance, at most one neuron can update its state with each neuron eventually getting its turn.

Hopfield described the asynchronous dynamics of the system using an *energy landscape*. The energy of the system is given by the negative of the quadratic form associated with the weight matrix. More precisely, the energy $\mathcal{E}(x)$ of the system in state x is given by $-\frac{1}{2} \sum_{i,j} w_{ij} x_i x_j$. In this landscape, stable states are local minima. Also, it can be easily seen that an asynchronous step does not increase the energy of the system. This guarantees that the system will eventually reach a stable state when operated asynchronously. Such a convergence is not guaranteed in the case of synchronous operation.

We now define precisely how Hebb's rule selects the weights of the interconnections. To store a vector in the system, this rule requires that each interconnection remember the correlation of the states of the two neurons it interconnects. More precisely, we set the weights $w_{ij} = v_i v_j$ to remember a single vector v. With this choice of weights, the system has a stable point at state v. A state is called stable if no transition out of it is possible. In other words, for each i, $v_i = \mathrm{sgn}(\sum_{j \neq i} w_{ij} v_j)$. Note that the notion of stability does not depend on the mode of operation.

Moreover, for this choice of weights, when the system is started at a state x within a distance $n/2$ from the stored vector v, it gets into state v in one synchronous step. In the asynchronous mode of operation, the state of the system converges monotonically to the stable state v. (As customary, we measure the distance between two ± 1 vectors or states by the number of components in which they differ.) It is this *attracting* nature of the system that gives it an associative memory or *error-correcting* capability.

To store several vectors in the system, we simply add the corresponding weights to-

gether. The hope is that if the stored vectors are sufficiently different, such a linear addition of weights would not cause much interference in the error–correcting behavior of the system. We call each such stored vector a *fundamental* memory.

When we store a number of fundamental memories in the system, we expect each of them to be stable and to attract all the vectors within a ρn distance for some constant $\rho > 0$. Sometimes we relax the requirements that each fundamental memory be stable and that the system converge to a stable state. The system is still considered to be error–correcting, if every vector within a distance ρn from a fundamental memory eventually ends up within a distance of εn for some $\varepsilon < \rho$. We call this εn *residual error.*

Another important characteristic of the system is the *rate of convergence.*

Given *any* set of m fundamental memories, we would like to store them in the system. But this requirement is somewhat at odds with the requirement of error-correction. For we cannot expect to store vectors which are too close to each other. A reasonable minimal requirement in this connection is that we would like to store almost all sets of m vectors. Therefore, we will take a set of m *random vectors* as our set of fundamental memories and expect the system to remember them with *probability near 1.* This randomness is often achieved by coding the input.

(We remark that not every set of m vectors can be made stable even for $m = 4$.)

When $m = 1$, we have already seen that the fundamental memory is a stable state of the system which attracts all vectors within a distance $n/2$ in one synchronous step. When we have a number of fundamental memories, the retrieval of a memory will be disturbed by the *noise* created by the other fundamental memories. Yet, we hope that this noise is not overwhelming when the number of fundamental memories is not too large. Hence, the main question is to determine the amount of error-correction and the rate of convergence as a function of the number m of fundamental memories. In the following section, we present some of the known answers to this question.

2 Previous Results

Several researchers have described and predicted the features of the model using simulations and approximate calculations based on some independence assumptions. Also, this model is related to models of spin glasses. We refer the reader to Hopfield [8], Amit, Gutfreund, and Sompolinsky [2] and Mezard, Parisi, and Virasoro [19] for a wealth of information.

Basic questions about the absolute stability of the global pattern formation in dynamical systems have been studied by Grossberg [5], and Cohen and Grossberg [3] using Liapunov functions.

In the following, we survey some of the *rigorously* proved results in the case when the system of neurons is **fully** interconnected. Let m denote the number of fundamental memories.

McEliece, Posner, Rodemich and Venkatesh [18] determined the maximum number of stable fundamental memories and the convergence properties in the presence of *random* errors.

- If $m < n/(4\log n)$, then (with probability near 1) *all* fundamental memories will be stable. Also, for any fundamental memory, the system can correct *most* patterns of less than $n/2$ errors in one synchronous step.
- If $n/(4\log n) < m < n/(2\log n)$, then still *most* fundamental memories will be stable with the above described capability of correcting most patterns of errors.

When m is larger than $cn/\log n$, in particular, when $m = \alpha n$, the fundamental memories are not retrievable exactly, but one still may find stable states in their vicinity. This is

suggested by the 'energy landscape' results of Newman [20]. In particular, Newman proves that, for all fundamental memories, all the vectors which are exactly at a distance of ρn from the fundamental memory have energy in excess of at least μn^2 above the energy level of the fundamental vector. Thus, when starting from a fundamental memory, the system cannot wander away too far.

The behaviour of the system in the presence of worst–case errors was addressed by Komlós and Paturi [12], who proved the following results.

α_s, α_a, ρ_s, ρ_a, ρ_b are absolute constants.

- In the synchronous case, if $m \leq \alpha_s n$, and if the system is started *anywhere* within a distance of $\rho_s n$ from a fundamental memory v, then, in about $\log(n/m)$ synchronous steps, it will end up within a distance $ne^{-n/(4m)}$ from v.

  In particular, when $m < n/(4 \log n)$, the system will *converge* to v in $O(\log \log n)$ synchronous steps.

- In the asynchronous case, if $m \leq \alpha_a n$, and if the system is started *anywhere* within a distance of $\rho_a n$ from a fundamental memory v, then it will *converge* to a stable state within a distance of $ne^{-n/(4m)}$ from v.

  In particular, when $m < n/(4 \log n)$, the system will converge to v.

- For any fundamental memory v, the maximum energy of any state *within* a distance of $\rho_a n$ from v is less than the minimum energy of any state *at* a distance of $\rho_b n$ from v , and there are no stable states in the annuli defined by the radii $\rho_b n$ and $ne^{-n/(4m)}$ centered at the fundamental memories.

3 General Interconnections

The previous models seem to rely on their dense interconnections for associative memory properties. Neither the physiological data nor the VLSI technology support such dense interconnections. In this paper, we consider models in which the neurons are less densely interconnected. We try to determine the properties of the underlying interconnection graph that are responsible for the emergence of associative memory.

Below, we try to indicate how the interconnections influence the degree of error–correction and the rate of convergence.

When the system is fully interconnected, and when we have just one fundamental memory, we can retrieve the fundamental memory even in the presence of $n/2$ errors. Furthermore, this error–correction takes only one synchronous step.

If we have several fundamental memories, the main difficulty is in showing that the noise introduced by the combined Hebb weights of the other fundamental memories would not completely destroy the error–correction capabilities observed in the simple case of storing one vector. On the other hand, we can no longer expect one–step synchronous convergence. Even if we store as few as two vectors, there is enough noise to slow down the system to a $\log \log n$ convergence. But, as we showed in [12], this $\log \log n$ convergence time is retained even when we have cn fundamental memories.

When the system is not fully interconnected, storing even one vector introduces new problems. Firstly, if the graph does not have sufficiently good *connectivity* properties, then, for a d-regular interconnection graph, as few as $d/2$ errors can make full retrieval impossible (see Examples 1 and 2). In addition, even if the graph does have good connectivity properties, the one–step error correction found in fully interconnected graphs gives way to a more gradual error correction. In fact, convergence time is roughly equal to the diameter of the graph.

When we want to store more than one vector, in addition to the connectivity proper-

ties that ensure good error–correction in the one–vector case, we need a graph that is not too sparse. It can be seen (see Example 3) that if we store at least two vectors, then mere stability of these vectors requires that every neuron be connected to more than $\log n$ other neurons. When we have several fundamental memories, the main result is that good connectivity and a large degree give us convergence; in fact, the convergence time is bounded by the sum of the diameter and $\log \log n$.

So far, we have tacitly assumed that the system has not exceeded its *memory capacity*. We now address the question of capacity.

We distinguish between *full capacity*, the maximum number of fundamental memories that can be stored such that they are fully retrievable, and *partial capacity*, the maximum number of fundamental memories that can be stored such that a large fraction of the bits can be retrieved with few residual errors.

In the case of *fully* interconnected systems, the residual error is given by $ne^{-n/(4m)} = ne^{-1/(4\alpha)}$, where n is the number of neurons, m is the number of fundamental memories stored, and $\alpha = m/n$. Hence, the full capacity is $O(n/\log n)$, and the partial capacity is linear in n.

In the case of *general* connections (with good connectivity properties and sufficiently large degrees), the residual error is governed by the same formula with $\alpha = m/d$. Hence, the full capacity is $O(d/\log n)$, and the partial capacity is linear in d.

This phenomena of diminished capacity has been observed before by other researchers (McClelland, [16], Kinzel [9]).

Fortunately, the degree of error correction does not decrease. If the graph is highly connected, one can still recover the fundamental memories in the presence of *arbitrary* error patterns, and the number of errors allowable is still proportional to n. Also, the decrease of errors is sufficiently rapid, so that terminating the synchronous algorithm after a small number of steps, leaves negligibly few errors in the retrieval.

It is, of course, necessary to explain what we mean by good connectivity properties in all these statements. The key parameter we will work with is the ratio of the second eigenvalue to the first. This ratio was shown by Alon and Milman [1] to reflect connectivity properties.

Example 1. Let G be the d–dimensional hypercube, where d is an odd integer. We assume, without loss of generality, that we want to store the vector of all 1's. Let I be a set of neurons which form a $\lceil d/2 \rceil$–dimensional subcube. It is easy to see that if the system is initiated with -1's for the neurons in I and $+1$'s for the remaining neurons, the state remains unchanged.

This shows that we cannot even correct $\sqrt{2^d} = \sqrt{n}$ worst–case errors in a hypercube. As we will see later, this inability to correct errors is due to the fact that the largest two eigenvalues of the hypercube graph (equal to d and $d-2$) are too close to each other, which reflects a low degree of connectivity.

Example 2. Assume that the interconnection graph has a clique of size $d/2 + 1$, and we store the vector of all 1's. When the system is initiated with -1's for the neurons corresponding to this clique, the neurons in the clique will never change their state.

Example 3. It is easy to see that any two stable states have the following property: if they are identical in the neighbourhood of a vertex i, then they must be identical at i, too.

Now, let the interconnection graph be d-regular. The probability that two randomly selected fundamental memories have unequal values at i, but equal values at i's neighbours, is $2^{-(d+1)}$. Thus, if $n \gg 2^{d+1}$, then the probability that both fundamental memories are stable is near 0.

The structure of the remainder of the paper is the following. First, we consider the simple case of storing one vector to gain an understanding of the connectivity properties

required for error–correction. Following that, we present our results in the general case of storing many fundamental memories. Finally, we present the combinatorial tools required.

4 Results

Let $G = \langle V, E \rangle$ be the undirected graph that represents the interconnections among the neurons. Let A be the adjacency matrix of the G, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ be the eigenvalues of A, and let $\mu = \max_{i \geq 2} |\lambda_i|$. We will assume that G is a d–regular graph, so that $\lambda_1 = d$. $c_1, c_2, \ldots$ are absolute constants.

Remark. The assumption that G is regular is technical. For non-regular G, the standard technique is to consider the eigenvalues of the matrix Q rather than those of the adjacency matrix A. Here $Q = D - A$, where D is the diagonal matrix containing the degrees of G in the diagonal. Q is positive (semi)definite, and has smallest eigenvalue 0 with eigenvector (1,1, ..., 1). The second smallest eigenvalue λ (equal to $\lambda_1 - \lambda_2$ for regular graphs) is critical for expanding properties. For more information, see Alon and Milman [1].

One Vector Case: Without loss of generality, we will assume that we want to remember the vector of all 1's. In this case, *the learning algorithm amounts to each neuron resetting its state to the majority state among its adjacent neurons.* Let x be a vector which has ρn -1's and $(1-\rho)n$ 1's in it. Let $I \subset V$ be the set of the ρn neurons with the state -1. For the system in state x to improve itself and converge to the vector of all 1's, we need that many neurons be adjacent to more neurons outside of I than inside of I. This seems to imply that G should have good expansion properties, which, in turn, are related to the eigenvalues. In fact, it is clear from Examples 1 and 2 that graphs which have a small subset of vertices with many of their incident edges concentrated within the subset, may have trouble correcting errors. We measure this concentration using the *average degree* of subgraphs in relation to their size. Since the average degree of any graph is bounded from above by its largest eigenvalue, measuring the concentration is reduced to finding upper bounds on the largest eigenvalues of subgraphs of G. We find several such bounds in Lemma 2.

The importance of concentration was noted earlier by Pippenger [21], and Dwork, Peleg, Pippenger, and Upfal [4] in the construction of fault-tolerant networks.

We will relate now eigenvalues of the graph to convergence properties of the system.

Lemma 1 *If G has $\rho n < n/2$ errors, then at most $\rho' n$ errors remain after one step, where*

$$\rho' = c_1 \rho (\frac{\mu}{d})^2$$

Thus, in the one vector case, we see that even a bounded degree graph with good connectivity properties (small μ/d) guarantees the recovery of the memory in the presence of a constant fraction of errors. In fact, it takes $O(\frac{\log n}{\log(d/\mu)})$ steps for synchronous convergence.

Let us see now how the error–correction behavior and the rate of convergence will be modified when we store a number of fundamental memories.

General Case: Let us choose m fundamental memories, $v^1, v^2, \ldots, v^m$, independently from a uniform distribution. Write $\alpha = m/d$, where d is the degree of the graph. (Recall that this α played a crucial role in the fully connected case.) Let W^i, $i = 1, 2, \ldots, m$, be the weight matrices corresponding to v^i:

$$W^i_{jk} = \begin{cases} v^i_j v^i_k & \{j,k\} \in E \\ 0 & \text{otherwise} \end{cases}$$

for $j, k = 1, 2, \ldots, n$. The weight matrix of the system is $W = \sum_i W^i$.

We define the energy of the system at state x as $\mathcal{E}(x) = \sum_{i=1}^m \mathcal{E}^i(x) = -1/2 \sum_{i=1}^m (x, W^i x)$ where $\mathcal{E}^i$ represents the energy component

due to the i–th fundamental memory. It can be seen easily that an asynchronous step never increases the energy. This fact guarantees asynchronous convergence in any system with symmetric weights.

Now we present our results in a precise form for systems with arbitrary interconnections. Notice their similarity to the results in the fully interconnected case.

β_s, β_a, ρ_s, ρ_a, ρ_b are absolute constants. Let

$$\beta = (m+\mu)/d = \alpha + (\mu/d) \text{ and } \varepsilon = e^{-c_2/\beta}$$

The parameter β plays an analogous role to that of the parameter $\alpha = m/n$ in the case of fully connected systems. ε represents the residual error in the recall procedure.

The following statements hold with probability $1 - o_n(1)$.

Theorem 1 *In the synchronous case, if* $m + \mu \leq \beta_s d$, *and if the system is started within a distance of* $\rho_s n$ *from a fundamental memory, then, in*

$$O(\log(1/\beta) + \frac{1}{\beta \log(d\beta/\mu)})$$

$$= O(\log\log(1/\varepsilon) + \frac{\log(1/\varepsilon)}{\log[d/(\mu\log(1/\varepsilon))]})$$

synchronous steps, it will end up within a distance εn *from the fundamental memory, i.e., it will get within a distance* εn *of the fundamental memory, and then it will remain (for ever) within that distance.*

When $m + \mu = O(d/\log n)$, *the system will* converge *to the fundamental memory in time*

$$O(\log\log n + \frac{\log n}{\log[d/(\mu \log n)]})$$

In addition, if $\mu = O(\sqrt{d})$, *(which is the case for most d-regular graphs), and* $d > (\log n)^{2+\xi}$, *then the system will converge to the fundamental memory in time*

$$O\left(\log\log n + \frac{\log n}{\log d}\right)$$

Notice that in the case $d = n - 1$, the last formula gives back our earlier result [12] of a $\log\log n$ synchronous convergence.

Theorem 2 *In the asynchronous case, if* $m + \mu \leq \beta_a d$, *and if the system is started within a distance of* ρ_a *from a fundamental memory, then it will* converge *to a stable state within a distance of* εn *from the fundamental memory.*

In particular, when $m + \mu = O(d/\log n)$, *the system will converge to the fundamental memory.*

Theorem 3 *For any fundamental memory* v, *the maximum energy of any state* within *a distance of* $\rho_a n$ *from* v *is less than the minimum energy of any state* at *a distance of* $\rho_b n$ *from* v, *and there are no stable states in the annuli defined by the radii* $\rho_b n$ *and* εn *centered at the fundamental memories.*

5 Combinatorial Tools

An important step in proving our theorems is to characterize the *dynamics* of the system. Let us first consider the synchronous case. Let x be a vector at a distance of $\rho n \leq \rho_s n$ from some fundamental memory. In one synchronous step, the system, started at state x, will move to a state x', which is at a distance of $\rho' n$ from that fundamental memory. Our goal is to determine the relationship between ρ and ρ'. In fact, it turns out that the *convergence in the synchronous case is monotone.* We present this relationship in the Main Lemma. All our theorems in the synchronous case follow from this lemma.

In the asynchronous case, we establish energy barriers for vectors within a distance of $\rho_a n$ from a fundamental memory. We use these barriers to show that the system cannot go too far from the fundamental memory and consequently converges near or to the fundamental memory, since the main lemma guarantees the existence of an annulus centered at the fundamental memory, which is free of stable states. Here, we use the simple fact that

in any symmetrically interconnected system asynchronous convergence is guaranteed.

In proving these lemmas, we face the task of estimating the sum of $m-1$ independent quantities of the form $S = \sum_{(i,j)\in E_H}(y_i, y_j)$, where y_i are independent random ± 1 vectors with uniform distribution, and H is a subgraph of G with E_H as its set of edges. For this purpose, we derive an upper bound on the moment generating function of S. This will show that the tail of the distribution of S is governed by the largest eigenvalue of the subgraph H. Hence, we also need estimates on the eigenvalues of subgraphs of G.

5.1 Moment Generating Function

The theorem below will be used for estimating the amount of noise in the system. We think it is interesting on its own right.

Let G be a simple graph with n vertices and N edges. We assign independent random variables X_i to the vertices of G, each taking the values ± 1 with equal probabilities $1/2 - 1/2$. We define the sum

$$S = \sum_{\substack{\{i,j\} \text{ is an} \\ \text{edge of } G}} X_i X_j$$

(thus $\mathrm{E}S = 0$ and $\sigma^2 = \mathrm{Var}(S) = N$).

Let $K_2 = 1/2\, N$, and for $r \geq 3$, let K_r denote the number of simple cycles of length r in G. The following theorem gives an estimate on the Laplace transform of the distribution of S.

Theorem 4 *The moment generating function* $\mathrm{E}e^{tS}$ *of* S *can be bounded as*

$$\mathrm{E}e^{-tS} \leq \mathrm{E}e^{tS} \leq e^{\sum_{r\geq 2} K_r t^r} \leq e^{1/2\, Nt^2/(1-\lambda_1 t)}$$

for $0 \leq t < 1/\lambda_1$.

The theorem says that $\mathrm{E}e^{tS} \leq e^{cNt^2}$ as long as $|t| < 1/\lambda_1$. Thus, λ_1 determines the behaviour of the tail of the distribution of S. Namely, $S/\sigma(S)$ behaves approximately as standard normal in the range $|x| \ll \sigma/\lambda_1$.

Remark: The proof below will actually show that the above inequality

$$\mathrm{E}e^{-tS} \leq \mathrm{E}e^{tS} \leq e^{1/2\, \sigma^2 t^2/(1-\lambda_1 t)}, \; 0 \leq t < 1/\lambda_1$$

holds for $S = \sum_{1\leq i<j\leq n} a_{ij} X_i X_j$, where $A = \{a_{ij}\}$ is an *arbitrary* non-negative symmetric matrix, λ_1 is the largest eigenvalue of A, and N was replaced in general by $\sigma^2 = \sum_{i<j} a_{ij}^2$, the variance of S. As a matter of fact, A could even have negative entries, but then λ_1 should be defined as the largest eigenvalue of the matrix, whose entries are the absolute values of the corresponding entries of A.

Corollary 1 *For any* $y > 0$,

$$\mathrm{P}(S \geq y) \leq e^{-1/2\, y^2/(\sigma^2 + \lambda_1 y)}$$

5.2 Estimates of Eigenvalues

To apply Theorem 4 to subgraphs of G, we need to bound their largest eigenvalues. The largest eigenvalue is also useful in bounding the number of edges in a graph.

For a subgraph H of G, we define the neighbourhood graph $N\{H\}$ of H as the graph determined by those edges of G (and the implied vertices) which are incident to at least one vertex in H. The largest eigenvalue of a subgraph H of G is denoted by $\lambda_1(H)$.

Lemma 2 *If* H *is a subgraph of* G *with* ρn *vertices, then the maximal eigenvalue of* H *satisfies*

$$\lambda_1(H) \leq \rho\lambda_1(G) + (1-\rho)\lambda_2(G)$$

Consequently, the number of edges in H *is at most* $1/2\, [\rho^2\lambda_1(G) + \rho\lambda_2(G)]\, n$. *Also,*

$$\lambda_1(N\{H\}) \leq 2\,[\sqrt{\rho}\lambda_1(G) + (1-\sqrt{\rho})\mu(G)]$$

In general, if the matrix B *is obtained from* A *(the adjacency matrix of* G*) by keeping only* ρn *rows and* γn *columns, and replacing all other elements of* A *with 0's, then, for any vector* u *of unit length,*

$$|\, Bu\,| \leq \sqrt{\rho\gamma}\,\lambda_1(A) + (1-\sqrt{\rho\gamma})\mu(A)$$

or, which is the same, for any two vectors x and y,

$$|(x, By)| \leq [\sqrt{\rho\gamma}\,\lambda_1 + (1 - \sqrt{\rho\gamma})\mu]\,|x|\,|y|$$

Consequently, if I and J are two subsets of the vertex set of G with $|I| = \rho n$, $|J| = \gamma n$, then the number $e(J, I)$ of (directed) edges going from J to I is at most $[\rho\gamma\lambda_1 + \sqrt{\rho\gamma}\mu]\,n$, and the largest eigenvalue of the graph H determined by the edges between J to I, is bounded as

$$\lambda_1(H) \leq 2\,[\sqrt{\rho}\lambda_1(G) + (1 - \sqrt{\rho})\mu(G)]$$

5.3 Error Correction Lemma

We now present the lemma that determines the rate of error–correction of the system in the synchronous case.

Main Lemma *Let $\rho_s > 0$ and*

$$\epsilon = e^{-c_2 \frac{d}{m+\mu}}$$

The following holds with probability $1 - o_n(1)$:

Let x be a vector at a distance of ρn from a fundamental memory, where $\epsilon \leq \rho \leq \rho_s$. Let x' be the resulting state after one step of the synchronous algorithm, given that the system is started in state x.

Then, the distance of x' from the fundamental memory is at most $f(\rho)n$, where $f(\rho)$ is given by

$$f(\rho) = c_3\rho\,(h(\rho) + [\frac{\mu}{d}\log\frac{1}{\rho}]^{2/3})$$

Corollary 2 *The number of residual errors in the synchronous case is at most ϵn.*

This lemma brings out the two key parameters: the ratio $\frac{\mu}{d}$ that governs the convergence, and the number of fundamental memories $m = \alpha d$ that together with $\frac{\mu}{d}$ determine the number of remaining errors. It is not hard to see that the parameter μ could actually be replaced by λ_2 here and throughout the whole paper.

5.4 Energy–Barrier Lemma

In the following, we give bounds for the energy of the system in the vicinity of a fundamental memory.

Consider a vector x at a distance of $\rho n < n/2$ from a fundamental memory v. Let $I, |I| = \rho n$, be the set of co-ordinates in which x and v differ. Then the following lemma holds.

Energy–Barrier Lemma *With probability $1 - o_n(1)$,*

$$|\mathcal{E}(x) - \mathcal{E}(v) - 2e(I, \bar{I})| \leq d\rho n$$

where $e(I, \bar{I})$ is the number of edges between I and $\bar{I}$.

Corollary 3 *The following holds with probability $1 - o_n(1)$: For all $\rho_1 < \rho_2 < 1/2$ and for every fundamental memory v, the energy of any state* within *a distance of $\rho_1 n$ from v is less than the energy of any state* at *a distance of $\rho_2 n$ from v if $\rho_1 \leq \rho_2/3$.*

References

[1] Alon, N. and Milman, V. D. (1984). Eigenvalues, Expanders and Superconcentrators. *25th IEEE Symposium on Foundations of Computer Science*, 320–322.

[2] Amit, D. J., Gutfreund, H. and Sompolinsky, H. (1987). Statistical Mechanics of Neural Networks near Saturation. *Annals of Physics*, **173**, 30–67.

[3] Cohen, M.A. and Grossberg, S. (1983). Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks. *IEEE Transactions on Systems, Man, and Cybernetics*, **13**, 815–826.

[4] Dwork, C., Peleg, D., Pippenger, N., Upfal, E. (1986). Fault Tolerance in Networks of Bounded Degree. *ACM Symposium on Theory of Computing*, 370–379.

[5] Grossberg, S. (1982). **Studies of Mind and Brain**. Boston:Reidel. 05–213.

[6] Hebb, D. O. (1949). **The Organization of Behavior**. New York:Wiley.

[7] Hinton, G. E. and Anderson, J. A. (Eds.) (1981). **Parallel Models of Associative Memory**. Hillsdale, NJ:Lawrence Erlbaum Associates.

[8] Hopfield, J. J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proc. Natl. Acad. Sci. USA* **79** 2554–2558.

[9] Kinzel, W. (1985). Learning and Pattern Recognition in Spin Glass Models. *Zeitschrift für Physik B.* 205–213.

[10] Kohonen, T. (1984). **Self–Organization and Associative Memory**. New–York: Springer–Verlag.

[11] Kohonen, T. (1987). State of the Art in Neural Computing. *IEEE First International Conference on Neural Networks I*, 77–90.

[12] Komlós, J. and Paturi, R. (1988). Convergence Results in an Associative Memory Model. *Neural Networks*, Vol. 1.

[13] Lashley, Y. S. (1960). In F. A. Beach et al. (Eds.), **The Neurophysiology of Lashley; Selected Papers of K. S. Lashley**. New York:McGraw–Hill.

[14] Little, W. A. (1974). The Existence of Persistent States in the Brain. *Math. Biosci.*, **19**, 101–119.

[15] Little, W. A. and Shaw, G. L. (1978). Analytic Study of the Memory Storage Capacity of a Neural Network. *Math. Biosci.*, **39**, 281–290.

[16] McClelland, J. L. (1986). Resource Requirements of Standard and Programmable Nets. In **Parallel Distributed Processing**, Vol. I. (Rumelhart, D. E., McClelland, J. L. and the PDP Research Group). MIT Press.

[17] McCulloch, W. S. and Pitts, W. H. (1943). A Logical Calculus for Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, **5**, 115–133.

[18] McEliece, R. J., Posner, E. C., Rodemich, E. R., Venkatesh, S. S. (1987). The Capacity of the Hopfield Associative Memory. *IEEE Trans. Information Theory*, **33**, 461–482.

[19] Mezard, M., Parisi, G. and Virasoro, M. A. (1987). **Spin Glass Theory and Beyond**. World Scientific.

[20] Newman, C. M. (1988). Memory Capacity in Neural Network Models: Rigorous Lower Bounds. *Neural Networks*. Vol 1.

[21] Pippenger, N. (1985). On Networks of Noisy Gates. *IEEE 26th Annual Symposium on Foundations of Computer Science*, 30–38.

[22] Rumelhart, D. E., McClelland, J. L. and the PDP Research Group (1986). **Parallel Distributed Processing** Vols. I and II, MIT Press.