

INVESTIGATION OF CAPACITY AND DYNAMICS OF HOPFIELD MODEL OF NEURAL NETWORK

THESIS SUBMITTED FOR
THE AWARD OF THE DEGREE OF
DOCTOR OF PHILOSOPHY

RAVINDRA SHARMA

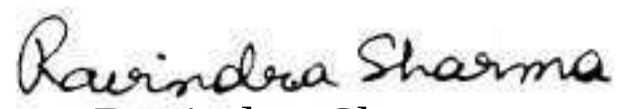


**Department of Computer/Information Sciences
School of Mathematics & Computer/Information Sciences
UNIVERSITY OF HYDERABAD
HYDERABAD, INDIA**

Department of Computer & Information Sciences
School of Mathematics & Computer/Information
Sciences
UNIVERSITY OF HYDERABAD

This is to certify that I, Ravindra Shanna, have carried out the research embodied in the present thesis for the full period prescribed under the Ph.D. ordinances of the University.

I declare, to the best of my knowledge, that, no part of this thesis was earlier submitted for the award of any research degree of any University.


Ravindra Sharma



Prof. Arun Kumar Pujari
Supervisor &
Head of the Department.



Prof. V.Kannan
Dean of the School

Acknowledgements

This dissertation is the outcome of kind cooperation, help, good-will and support in the forms, both technical and moral, extended by several noble individuals with whom I have been associated during the course of this research work.

I express my sincere thanks and gratitude to my research supervisor Prof. Arun Kumar Pujari for his able guidance, constant support, expert advice, encouragement and constructive criticism throughout the course of this research work. I have been immensely benefitted by his scholarship and inspired by his dedication to work. I shall always cherish my association with my respected teacher. I am also grateful to the members of his family.

I thank all other faculty members of the Department of Computer Science for their encouragement and interest in my work. Prof. M. Seetaramayya and Prof. V.Kannan, the Deans of the School are respectfully acknowledged for the helpful attitude within the legal bounds, of the University. I thank the staff of the AI Lab for their help extended to do this research work.

For doing a work like this the support of the people you live with is essential. All the members of my family, Sri Arvind Sharma, Smt. Susheela Sharma, Dr.Govind Sharma, Dr. Samata Sharma, Dr. Purnima Sharma, Dr. S.K. Sharma, Dr. Aruna Sharma, Dr.R.P. Sharma, Smt.Devi Purohit, Sri Surendra R Purohit, Sri R.D. Sharma, Smt. Manju Sharma, Kum. Keerti, Master Kapil, Master Bharat, Kum. Deepti, Master Gaurav, Kum. Chitra, Master Rohit, Master Varun, Kum. Vandana, Master Harsh and Master Sagar have been my main and constant source of inspiration and support. They provided encouragement when giving up seemed like the best thing to do. I am grateful to all of them.

No other words are adequate to express my everlasting gratitude to Ms. B.T.GangU,

Mr. K.Lakshmi Narayana, Ms.Shinymol Antony Kondayil and Ms. **E.Uma**, for their frequent discussion, valuable suggestions and criticism. I shall always remain indebted to them for this kind help.

I am grateful to the management and teachers of the institutions in Hyderabad where I received formal education. Foundations for future learning was laid by my teachers of St. **Paul's** High School where I received my first formal education. I remain indebted to all the teachers of the school. My humble thanks are due to the teachers of **Kendriya Vidyalaya Golkonda**, where I studied for two years, for inspiring me to do experiments in various laboratories and to analyse the results critically. **Osmania** University, has provided me an opportunity to do my graduate and post-graduate studies. I am deeply indebted to the faculty members and other staff of this institution.

I gratefully acknowledge the encouragement of Prof. A.Chandra **Sekhar Bhat**, Dean Students Welfare, for his keen interest in my work and the help extended to complete this research work.

I am grateful to the Indian Space Research **Organisation** for sponsoring part of this research work in the form of Senior Research Fellowship under RESPOND project. Without this fellowship I would not have been able to complete my dissertation.

I am thankful to the drivers and other staff of the University transport section for making the journey to the campus safe and comfortable. I am also thankful to the catering staff of the hostel for the delicious food served during my stay on the University campus. Thanks are also due to the Library staff for the prompt services.

I am also thankful to everyone who has been directly or indirectly associated with my work.

Table of Contents

1	OVERVIEW	1
1.1	INTRODUCTION.	1
1.2	OVERVIEW OF PRESENT WORK.	1
1.3	OUTLINE OF THE DISSERTATION.	3
2	HOPFIELD MODEL OF NEURAL NETWORK	4
2.1	INTRODUCTION.	4
2.2	ARTIFICIAL NEURAL NETWORKS.	5
2.3	HOPFIELD MODEL OF NEURAL NETWORK.	6
2.4	HOPFIELD CLASS NETWORKS.	11
2.5	CAPACITY ESTIMATES OF HOPFIELD NETWORK.	14
2.5.1	Hopfield's Estimates.	14
2.5.2	Estimates Using Hyperplane Counting Argument.	15
2.5.3	Critical Views About Definition 2.3.	16
2.5.4	Estimates Using Coding Theory.	17
2.5.5	Perfect Storage Condition.	18
2.5.6	Upper and Lower Bounds of Capacity.	19
2.5.7	Capacity Estimates For Delta Learning Rule.	21
2.5.8	Presence of Spurious Stable States.	22
2.5.9	Hysteretic Property.	23
2.6	ASYMMETRY IN HOPFIELD NETWORKS.	24
2.6.1	Asymmetry and Dilution.	26

Table of **Contents**

2.6.2	Asymmetric Learning Rule	27
2.7	BIASED CANDIDATE STATE VECTORS.	28
2.8	CORRELATED CANDIDATE STATE VECTORS.	30
2.9	CAPACITY WITH BIASED AND CORRELATED PATTERNS.	30
2.9.1	Low Level Activity With Bipolar Neurons.	30
2.9.2	Low Level Activity With Binary Neurons.	32
2.9.3	Improving Retrieval	32
2.9.4	Improving Performance By Adaptive Threshold	33
2.9.5	Non-local Learning Rules.	34
2.9.6	Generalised Learning Rule.	36
2.9.7	Biased State Vectors In Clipped Network	37
2.9.8	Generalised Rule For Correlated Patterns	38
2.9.9	Existence of Required Network	40
2.9.10	Semantic and Spatial Correlation	42
2.9.11	Hierarchical Correlation	43
2.10	MARKED CANDIDATE STATE VECTORS.	44
2.11	MULTI-VALUED NEURAL NETWORKS.	46
2.11.1	q-State Potts model	46
2.11.2	q-state Ising Spin Networks.	47
2.11.3	q-state Hopfield network	50
2.11.4	Continuous Valued Neurons	52
2.12	HIGHER ORDER NEURAL NETWORKS.	52
2.13	CONCLUSION.	53
3	CHANGES IN DIAGONAL ELEMENTS	55
3.1	INTRODUCTION.	55

Table of Contents

3.2	MOTIVATION FOR STUDY OF DIAGONAL CHANGES.	56
3.3	AN EXAMPLE	57
3.4	CRITICAL VALUE OF J_{ii} FOR BIPOLAR NEURONS.	60
3.5	IMPACT OF DIAGONAL CHANGES ON NETWORK DYNAMICS . . .	65
3.6	CONDITIONAL STABILITY.	68
3.7	GEOMETRICAL INTERPRETATION.	70
3.8	THRESHOLD CHANGES AND CRITICAL VALUES.	71
3.9	DIAGONAL CHANGES vs THRESHOLD CHANGES.	74
3.10	DIAGONAL CHANGES AND ENERGY FUNCTION.	75
3.10.1	Change in Diagonal Elements of Synaptic Matrix.	75
3.11	DIAGONAL CHANGES IN NETWORK WITH BINARY NEURONS . . .	77
3.12	CONCLUSION.	78
4	POLYHEDRAL COMBINATORICS FOR NEURAL NETWORKS	79
4.1	INTRODUCTION.	79
4.2	POLYHEDRAL COMBINATORICS.	81
4.3	SIMILAR WORKS.	82
4.4	BASIS OF THE PRESENT WORK.	83
4.5	HOPFIELD NETWORK FOR ONE STABLE STATE.	84
4.5.1	Construction of J^{ϵ} and 0	85
4.6	HOPFIELD NETWORK WITH TWO STABLE VECTORS.	86
4.6.1	Construction of $J^{\epsilon^1 \epsilon^2}$ and 0	87
4.7	ORDER OF VECTORS IN TWO STABLE STATE FORMULATION . . .	89
4.7.1	Multiple Weights	91
4.8	HOPFIELD NETWORK WITH MORE STABLE VECTORS.	92
4.8.1	Three Stable Vectors	92

Table of *Contents*

4.8.2	Construction of $J^{\xi^1 \xi^2 \xi^3}$ and 0.	92
4.8.3	More Than Three Stable Vectors.	95
4.9	STABLE VECTORS WITH SPECIFIC NUMBER OF 1 BITS.	96
4.10	CONCLUSION.	97
5	CONCLUSION	99
5.1	SURVEY ON CAPACITY.	99
5.2	STUDY OF SELF-FEEDBACK IN HOPFIELD NETWORK	100
5.3	POLYHEDRAL COMBINATORICS.	101
	Bibliography	102

Chapter 1

OVERVIEW

1.1 INTRODUCTION

Neural networks have been used for wide variety of applications of complex adaptive systems. The application areas in which neural networks are being extensively applied are Pattern Recognition, Speech, Vision, Robotics, Industrial Process Control, Knowledge Data Bases, On-Line Simulation and Decision making, Intelligent Artificial Organ, and Physiological Software and Services [Soucek89]. Fuzzy reasoning, very high speed signal processing and process control, high reliability, and high performance-price ratio are some of the other emerging features of neural networks. Successful application of neural networks to real world problems depends strongly on the knowledge of learning properties and performance [Karayiannis93]. This is the main motivation behind the present thesis. A humble attempt has been made to study Hopfield model and suggest new learning mechanisms. This could help in a better understanding of functioning of neural networks and application of neural networks to new areas.

1.2 OVERVIEW OF PRESENT WORK

This thesis makes an extensive study of the Hopfield model of neural network [Hopfield82] (hereafter referred to as *Hopfield Network*) and reports some new theoretical and experimental results. Much of the efforts in neural networks research is directed at implementation and realization rather than theory. For example, the convergence property of neural

network are still described by empirical and **rule-of-thumb** terms, even though **mathe-** mathematical techniques are available for studying the dynamic behaviour of neural networks [Sezan90]. The major contributions of this thesis are listed here.

1. It reports a survey of the research related to several modifications to the **Hopfield** network, the effect of these changes on the performance of the network, various definitions of capacity, and capacity estimates. There have been a very large number of research articles on capacity of Hopfield network and several modifications to this network reported in several diversified disciplines. These results are reported in this thesis in a unified way. One of the modifications to the Hopfield network is self-feedback character of neurons. This aspect has been selected for a detailed study.
2. A study of dynamics of Hopfield network is made based on numerical simulations. The emphasis of this study is on dynamics of Hopfield network with **self-feedback**, which seems to have received less attention by researchers. Some experimental observations and theoretical conclusions on the study of direct self-feedback in Hopfield network are reported.
3. An algorithm is proposed to make any two state vectors stable in the Hopfield network with **self-feedback**. These are the only states that are stable states of the network and spurious states do not exist. This algorithm is generalised to make more number of state vectors stable in the network. The underlying principle of this algorithm is based on polyhedral combinatorics. The set of stable states defines a convex **polytope** and the energy function is designed to be a facet touching only the specified stable states.

1.3 OUTLINE OF THE DISSERTATION

The dissertation consists of five chapters. Chapter 2 introduces **Hopfield** network with various notations, definitions, learning rules and update rules. This chapter also reports a survey of the research related to the estimates of the capacity of Hopfield network and that of several modifications of **this** model.

Chapter 3 reports some experimental observations and theoretical conclusions on the study of Hopfield network with direct self-feedback.

In Chapter 4 the techniques of polyhedral combinatorics are used to analyse the geometry of stable state vectors and its associated image. This chapter also describes construction mechanisms of **synaptic** matrix to make any one state vector as stable state and any two state vectors as stable states in the Hopfield network with self-feedback. It also generalises this algorithm to handle more number of stable vectors. A construction mechanism to make all vectors having a specific number of ones as stable states is also given in this chapter.

Chapter 5 summarises the contributions and limitations of the work reported in the dissertation and considers possible routes for further research work.

Chapter 2

HOPFIELD MODEL OF NEURAL NETWORK

2.1 INTRODUCTION

Human beings are constantly thinking since ages about the reasons for human capabilities and incapacities. Successful attempts have been made to design and develop systems that emulate human capabilities or help overcome human incapacities. The human brain, which has taken millions of years to evolve to its present architecture excels at tasks such as vision, speech, information retrieval, complex pattern recognition, all of which are extremely difficult tasks for conventional computers. A number of mechanisms have been which seems to enable human brain to handle various problems. These mechanisms include association, generalisation and self-organisation.

The hope to reproduce at least some of the flexibility and power of human brain by artificial means has led to the subject of study known as *Neural Networks*, *Neural Computing*, *Neurocomputing* or *Brainlike Computation* [Anderson92]. Neurocomputing is a fundamentally new and **different** approach to information processing. Neurocomputing is concerned with parallel, distributed and adaptive information processing systems that develop information processing capabilities in adaptive response to an information environment [Hecht-Nielsen91]. It is not necessary that the architecture of brain is copied as it is to the extent to which it has been understood. Implementation of the functions of brain by whatever means possible is the guiding force in **neurocomputing**.

In this chapter the concept of artificial neural networks is introduced in Section 2.2.

Hop field network [Hopfield82] is one of the simplest and most widely used neural network models. This model is described in Section 2.3. Search for better performance and application orientation has motivated researchers to consider various modifications to the Hopfield network. Section 2.4 to Section 2.12 report a survey of research related to various modifications to the Hopfield network, the effect of these changes on the performance of the network, various definitions of storage capacity, and capacity estimates. This survey is an attempt to present in an unified way the results of research articles reported in several diversified disciplines. This survey may help in an extensive study to provide an insight into dynamics of Hopfield network which may lead to precise design of Hopfield networks.

2.2 ARTIFICIAL NEURAL NETWORKS

The primary information processing structures of interest in **neurocomputing** are *Artificial Neural Networks (ANN)*. The potential of artificial neural network relies on massively parallel architecture composed of large but finite number of *artificial neurons* which act as simple computational elements connected by edges with variable weights. In this work hereafter artificial neural network is referred to as **neural network** and artificial neuron is referred to as neuron.

There are various models of neural networks which have been reported in literature. Some of trend setting models of neural networks are **Perceptron** [Rosenblatt58, Minsky69, Rumelhart86], Adaptive Neural Network [Widrow60], Linear **Associator** Model [Kohonen72, Anderson72], Little and Shaw model [Little75], Pattern associating and concept forming model [Amari77], Hopfield model [Hopfield82, Hopfield84, Hopfield86], **Grossberg** Models [Grossberg80], Self-organising Network [Kohonen84], and **Boltzmann** machine [Hinton84, Ackley85].

These models have certain aspects in common. Eight major aspects of a neural network have been identified in [Rumelhart86] are a set of processing units, a state of activation, an output function for each unit, a pattern of connectivity among units, a propagation rule for propagating patterns of activities through the network, an activation rule for combining the inputs to a unit with its current state to produce a new level of activation, a learning rule whereby patterns of connectivity are modified by experience, and an environment within which the system must operate.

All models of neural networks exhibit some basic characteristics which are different from the other computing paradigm. Some of the characteristic features of neural networks are Model free estimators (no mathematical model of how a system output depends on its input), Self-organisation (network carries out corresponding changes in its structure when the performance requirements are changed), Distributed encoding of information (information is superimposed and stored in the weights between neurons), Generalisation (a neural network is capable of generalising from a limited set of correctly learned functions to an entire class of special purpose functions), and **Geometrization** of computation (neural activity burrows a trajectory in the state space of large dimension and each point in the state space defines a snapshot of a possible neural network configuration).

2.3 HOPFIELD MODEL OF NEURAL NETWORK

Many researchers consider the **Hopfield** network [Hopfield82] as a model which has extensively influenced the field of neural networks. This section gives some relevant details of Hopfield network.

Topology:- Pattern of connectivity of a neuron with other neurons is referred to as topology of neural network. Neural networks can be broadly classified into two classes based

on topology of the network. These classes are *feed-back neural networks* (architecture can be described as an undirected graph) and *feed-forward neural networks* (neurons are arranged in layers with directed synapses between one layer and next layer).

A connection (synapses) between a pair of neurons in a feed-back neural network is characterised by a *synaptic weight* (*connection weight*). Higher value of **synaptic** weight between a pair of neurons indicate the tendency of the pair of neurons to be simultaneously active. The synaptic weights are determined by the *learning rule* used to train the network and these are represented as a *synaptic matrix*. If the output of each neuron is connected to all the other neurons, the network is said to be a *fully connected network*. If the output of a neuron is connected as an input to the same neuron, then the neuron is said to have a direct *self-feedback* character. In a fully connected feed-back neural network if the synaptic weight from neuron j to neuron i (J_{ij}) is same as the synaptic weight between neuron i to neuron j (J_{ji}) for all pairs of neurons then the synaptic matrix is symmetrical. The first subscript associated with J specifies **post-synaptic** neuron and second subscript specify **pre-synaptic** neuron.

Hopfield network is a fully connected, feed-back neural network of N neurons. The **Hopfield** network is uniquely defined by (J, θ) . Where J is a $N \times N$ symmetric matrix and θ is $N \times 1$ threshold vector having components θ_i , which is the threshold for neuron i . Each choice of J and θ defines a specific Hopfield network with N neurons.

State of a neuron:- Activity level of a neuron (represented as σ_i) is also known as its state. The neurons in [Hopfield82] are **two-state** neurons. **Two-state** neurons which are assigned a value of 0 for inactive (OFF) state and 1 for active (ON) state are termed as *binary neurons*. **Two-state** neurons which are assigned a value of **+1** for active state and **-1** for inactive state of neurons are called *bipolar neurons*. The binary (0,1) and bipolar

(-1,1) neuron representations are equivalent. States of these neurons are related as

$$\sigma_i^{\text{bipolar}} = 2\sigma_i^{\text{binary}} - 1, \quad \forall i = 1, 2, \dots, N$$

State of Hopfield Network:- The states of all neurons (σ_i , for all i) at any instant of time t is the state of the Hopfield network and is represented by a state vector ξ . The i^{th} component of the state vector ξ is represented by ξ_i . The state of neuron t is same as the i^{th} component of state vector i.e., $\xi_i = \sigma_i$ at any given instant. In this work ξ_i is replaced by σ_i and vice versa without specifically mentioning it. This is however being done considering the time interval factor. The state of Hopfield network can be visualised as a pattern of activities of the neurons and thus the state vector is also termed as a *pattern*. *Stable states* of a Hopfield network are the states of the network which do not change under normal functioning of neural network.

Learning:- Learning in Hopfield network is the process of making certain states of network as stable states. This is achieved by determination of the **synaptic** matrix and threshold vector. Strategies for learning are broadly divided into two classes. These are *supervised learning* and *unsupervised learning*. In supervised learning the network is supplied with a sequence of examples. Each example conveys the required output state vector for a given input state vector. Usually, the training process is continued until the neural network learns all the examples. In unsupervised learning the *learning set* consists of state vectors that are to be made stable vectors of the network. This work deals with unsupervised learning and some of the parameters in terms of which the learning process of Hopfield network can be specified are listed below.

Candidate state vectors:- The P state vectors ξ_i^μ ($\mu = 1, 2, \dots, P$; $t = 1, 2, \dots, N$), that are to be made stable states of N neuron Hopfield network being designed are known as candidate state vectors. In this work N represents the number of neurons in a network and P represents the number of candidate state vectors.

Basin of attraction:- The set of all vectors, that converge to a stable state vector ξ^μ is the basin of attraction of ξ^μ . There is no general shape of basins of attraction. Learning in Hopfield network amounts to organizing the space of network states into basins of attraction of **preassigned** candidate state vectors.

Radius of direct attraction:- The radius of a sphere around each of the stable state vector, such that any vector in the sphere converges to the corresponding stable state vector in one step [Chandru93].

Speed of convergence:- A stable state is recalled if under the influence of a stimulus the Hopfield network drifts rapidly into the stable state. Speed of convergence is the measure of number of steps a network takes to converge to a fixed point after receiving the initial state. The dimensionality of the state space and number of basins of attractions in principle do not effect the speed of convergence [Kosko92].

Spurious state vector:- The stable state vectors of the designed Hopfield network which do not **belong** to the set of candidate state vectors are called the spurious state vectors.

The learning process has some **difficulties**. It is not always possible to formulate a Hopfield network with all members of given learning set as stable states. Some of the candidate state vectors may not become stable states of the network. Thus the stable states of a Hopfield network can be seen as belonging to two categories, *stored state vectors* (members of learning set) and spurious state vectors. A learning procedure with the following properties is considered to be an effective synthesis procedure for Hopfield networks [Farrell90].

Each candidate state vector is a stable state in the resulting Hopfield network. Each stable state vector must be attractive. In other words, each stored vector should have a domain of attraction. Learning rule should attempt to have a precise control on the extent of the domain of attraction of each stable state vector. Spurious stable states should be eliminated or **minimized**.

Learning rules can also be classified as *Local learning rule* or *Non-local learning rule* depending on the nature of information used to construct the **Hopfield** network. Information physically available at a synapses is called the local information. **Synaptic** weight changes in local learning rule depend only on local activity. Non-local learning rules also consider the activity in other nearby **synapses**.

Update rule:- A rule for evaluating the state of one or more neurons under the existing conditions and changing them if necessary is called a *transition rule*, activation rule or **update rule**. Each neuron receives either an external input in the form of initial state and/or weighted inputs from other neurons and the resulting state is evaluated using the update rule.

Dynamics of computation:- After the formulation of Hopfield network, the network is presented with an *initial (probe) state vector*. For a Hopfield network with N two-state neurons, initial state vector can be any one of the 2^N possible vectors. The update rule along with the order in which it is applied defines the dynamics of computation. Update rule of Hopfield network uses linear sum of product of connection states and their connection weights to determine the next state. The sum is called the local *field of neuron* or *potential of neuron*. Depending on the relationship between the potential and the threshold of a neuron the next state of the neuron is determined. In Hopfield network the update rule is discrete in time. During each time interval, state of one (**asynchronous update**) or more than one (**synchronous update**) neurons is evaluated. For asynchronous update the neuron to be updated in a unit time interval is selected randomly or in a deterministic way (fixed sequence of neuron, neuron receiving maximum local field, or neuron receiving maximum local field). In synchronous update when all the neurons are updated in a unit time interval, it is termed as **fully parallel** operation. When the synchronous update operation is not fully parallel, the selection of neurons can be done using one of the methods used for asynchronous update.

Trajectory on **Hypercube**:- Network activity is represented by a trajectory on the **hypercube** in the N-dimensional space. The state space of **Hopfield** network is **represented** by a hypercube. Each vertex of the hypercube represents a network state. The state transitions **from** initial state to a stable state can be viewed as a trajectory in the hypercube from one vertex, representing the initial state to another vertex representing a stable state (equilibrium state). An attract is a special network state, or a restricted set of states, to which the dynamical process, governing the time evolution of the network, brings the network, after a long enough time, from large classes of initial network states [Amit89].

In [Amit89] three basic types of trajectory behaviours have been identified for **asymptotic** neural network dynamics are *Fixed points* (trajectories which lead the network to remain on a single state for an appreciable period), *Limit cycles* (trajectories which lead rapidly to small cycles of states) and *Chaotic* (trajectories which wander **aperiodically** in an **uncorrelated** way in the space of the network states). A *two step limit cycle* in dynamics of Hopfield network is a situation when the network oscillates between two states of the network. A two step limit cycle can be seen as a bi-directional fixed point. **Energy Function**:- The central feature of Hopfield network is that each state of such networks can be associated with a quantity called *Energy (E)*. An energy function and a statistical methodology to describe the relaxation of symmetric network is introduced in [Hopfield82]. The energy function of Hopfield network is function of its state ξ at time t .

2.4 HOPFIELD CLASS NETWORKS

In recent years several models of neural networks have been proposed. Researchers have been proposing new models motivated by biological neural systems, application under study or improvement of performance of an existing model. A model may perfectly

suit the purpose for which it has been designed, but in the presence of many models, it becomes necessary to compare the general performance of various models. Such a study will help in evaluating various models and also serve as a basis for designing new models or modifying existing models. Though presence of many models of neural network necessitates a comparative study, because of large variations in the nature of these models this task is difficult. It is more so in the absence of a common criterion. Though common criterion are necessary for detailed comparison of the models, deciding this meaningful set of such criteria may itself require a separate research.

An estimate of the number of state vectors that can be made fixed points in a **Hopfield** network can serve as common criterion for evaluation of Hopfield networks functioning as associative memory. This measure is known as *storage capacity* or *capacity of Hopfield network*. Detail study of different aspects of storage capacity have been of interest to the researchers. Many different definitions and estimates of the capacity have been reported in different context.

Hopfield network, like any other neural network model has a limit beyond which the performance is not predictable. Search for better performance, higher storage capacity and application orientation of this model has motivated researchers to consider different options. The models which vary from the original Hopfield network but fits in its general framework can be said to belong to *Hopfield class of **neural** networks* or briefly *Hopfield class networks*.

The various modifications to Hopfield network till now have been mostly attempted in isolation. Each of these changes have an impact on the storage capacity. Several possibilities related with each of these aspects provide a very large number of combinations. An extensive study of these aspects and hence the various combinations of these aspects may provide an insight into the dynamics and application of Hopfield class networks. Such a study may also help in precise design of Hopfield class networks.

The following sections of this chapter reports a survey of the research related to various modifications to the **Hopfield** network, the effect of these changes on the performance of the network, various definitions of capacity, and capacity estimates reported in literature. There have been very large number of research articles related with the above mentioned issues reported on several diversified disciplines. Hence, it becomes difficult **for** any researcher to have access to all the results. Hence an attempt has been made to report these results in a unified way available at one place. For the study of Hopfield **class** networks the parameters based on which various modifications and capacity estimates reported in literature can be characterized are listed below to give an overview.

1. Nature of **synaptic matrix**:- (i) Fully connected (ii) Diluted (iii) Symmetric (iv) Asymmetric (v) Binary or clipped (vi) **Two-dimensional** (vii) Higher order.
2. Learning **rules**:- (i) Local learning rules (ii) Non-local learning rules.
3. Nature of values assigned to **neurons**:- (i) **Two-state** neuron (ia) Binary neuron (ib) Bipolar neuron (ii) Multi-valued neuron (iia) Multi-state neuron (q-state Potts Glass neuron, **q-state ising** spin neuron, **q-state** Hopfield neuron) (iib) Continuous neuron.
4. Nature of candidate state **vectors**:- (i) Randomly chosen candidate state vectors (ii) Specific candidate state vectors (iii) Biased candidate state vectors (iv) Unbiased candidate state vectors (v) Correlated candidate state vectors (vi) **Uncorrelated** candidate state vectors (vii) Sparse candidate state vectors (viii) **Marked** candidate state vectors (ix) Complex candidate state vectors.
5. Retrieval **parameters**:- (i) One step retrieval (ii) Iterative or fixed point retrieval (iii) High fidelity retrieval (iv) Low fidelity retrieval (v) Presence of Spurious **states**.

6. Update **interval**:- (i) Discrete time (ii) Continuous time.

2.5 CAPACITY ESTIMATES OF HOPFIELD NETWORK

2.5.1 Hopfield's Estimates

The estimate of the number of candidate state vectors that can be stored accurately in [Hopfield82] is obtained by conducting computer simulations and some analysis. Brief description of Hopfield network and storage estimate reported in [Hopfield82] is given here.

To make randomly selected set of P candidate state vectors as fixed points in the Hopfield network of N binary neurons, the **synaptic weights** J_{ij} ($i, j = 1, 2, \dots, N$) are obtained as

$$J_{ij} = \sum_{\mu=1}^P (2\xi_i^{\mu} - 1)(2\xi_j^{\mu} - 1) \quad \forall i \neq j \quad \text{and} \quad J_{ii} = 0 \quad \forall i = j$$

Neurons are randomly and **asynchronously** evaluated using the update rule

$$\sigma_i(t+1) = \begin{cases} 1 & \text{if } \sum_{j=1}^N J_{ij} \sigma_j(t) > \theta_i \\ 0 & \text{if } \sum_{j=1}^N J_{ij} \sigma_j(t) < \theta_i \end{cases}$$

Under these conditions with $P = 0.5N$ the assigned candidate state vectors are always stable and exactly recoverable. With $P = 0.15N$ about half of the candidate state vectors evolved to fixed points with very less errors. But the rest evolved to quite different fixed points. Thus $0.15N$ candidate state vectors can be simultaneously stored by Hopfield network with N neurons before error in recall is severe.

DEFINITION 2.1:- The storage capacity (a) of a Hopfield network is defined as the ratio of number of candidate state vectors (P) that are made stored stable states to the number of neurons (N) in the network, $\alpha \approx \frac{P}{N}$. Thus storage capacity is the number of candidate state vectors made stable state per neuron. **Critical storage capacity** α_c

is defined as that storage capacity of **Hopfield** network beyond which it is not possible to store candidate state vectors without affecting the stability of already stored state vectors.

The storage capacity at any given level of accuracy can be increased by a factor of 2 by a judicious choice of the threshold of individual neurons [Hopfield82]. This can also be achieved by using bipolar neurons with **synaptic** weights, $J_{ij} = \sum_{\mu=1}^P \xi_i^{\mu} \xi_j^{\mu}$, $J_{ii} = 0$ and all threshold elements value zero. This modified prescription improves the level of accuracy and all randomly selected candidate state vectors **upto** $P = 0.15N$ can be accurately stored and recalled. The critical storage capacity $\alpha_c = 0.15$.

2.5.2 Estimates Using **Hyperplane** Counting Argument

In [Abumostafa85], using a **hyperplane** counting argument from pattern recognition, it has been shown that the number of candidate state vectors that can be made stable in Hopfield network is bounded above by the number of neurons N in the network. Bipolar neurons, fixed real numbers as synaptic weights, and undirected connections ($J_{ij} = J_{ji}$) without **self-feedback** ($J_{ii} = 0$) are considered in [Abumostafa85]. Given an initial vector, the neurons are updated randomly and **asynchronously** until a stable state is reached.

DEFINITION 2.2:- Information capacity of a memory is defined as the logarithm of the number of cases it can distinguish.

For the Hopfield network, information capacity can be obtained by estimating the number of different sets of values of J_{ij} and θ_i that can be distinguished merely by observing the state transition scheme of the neurons. This corresponds to the number of distinct networks of fixed N neurons. The key factor in estimating the number of distinct networks is the known estimate for the number of threshold functions. By estimating the upper bound and lower bound on capacity it is concluded that **the information** capacity of Hopfield network with N neurons is exactly of the order of N^3 bytes. A definition of

maximum storage capacity is also proposed in [Abumostafa85].

DEFINITION 2.3:- The number of candidate state vectors that can be made fixed points in a **Hopfield** network of N neurons is the maximum value of $1C$, **such** that, any K vectors of N binary entries can be made stable in the network by a proper choice of synaptic matrix J and threshold vector θ . Then $\alpha_c = \frac{K}{N}$ is the critical storage capacity of Hopfield network.

It is proved in [Abumostafa85] that the number of stable states K can be **atmost** N . In other words, Hopfield network cannot have more than N arbitrary stable states.

2.5.3 Critical Views About Definition 2.3

Different views of the Definition 2.3 of capacity given in [Abumostafa85] is proposed in [Bruck88, Prados89, Montgomery86]. In [Bruck88] it is proved that for Hopfield networks with zero diagonal synaptic matrix there exists pairs of state vectors which cannot be made fixed point simultaneously. Thus according to Definition 2.3, K cannot even be 2 and storage capacity can atmost be 1. In [Montgomery86] it is proved that there are many (**atleast** $N2^{N-1}$ out of possible 2^{2N}) pairs of state vectors for which Hopfield network cannot be constructed. Thus there is no guarantee for the existence of Hopfield network for $K = 2$.

Under these circumstances there is a need for alternate definition of capacity of **Hop-**field network. In [Prados89] it is proposed that the capacity of Hopfield networks should be mentioned only in terms of probability associated with storing randomly chosen candidate state vectors. In [Bruck88] an alternative definition of Hopfield networks is proposed based on notion of descriptor of a set. Descriptor of a set S denoted by D_S is a list of rules which are true in the set.

DEFINITION 2.4:- The storage capacity of a Hopfield network of order N is defined as the maximal integer K , such that, for any set M of K vectors with descriptor D_M

there exists **synaptic** matrix $\mathbf{J}$ and threshold vector θ such that M is contained in the set of stable states of the network $\mathbf{M}_N$. The storage capacity with respect to the descriptor D is denoted by CD^*

When D is an empty set then CD is equivalent to the Definition 2.3 of **capacity**. D can be formalised such that it describes a particular and unique set and then the question of finding the storage capacity turns out to be a question of designing a **Hopfield** network such that this set is contained in the set of stable states of the network $\mathbf{M}_N$. A simple upper bound on CD is the size of the largest set with descriptor D . With D getting more restrictive the storage capacity tends to increase, but also the above upper bound on storage capacity tend to decrease. The main drawback of the Definition 2.3 is that it does not hold true for the candidate state vectors which differ in one bit only. This can be avoided in the Definition 2.4 of storage capacity.

2.5.4 Estimates Using Coding Theory

Using techniques from coding theory, especially random coding and sphere hardening, the storage capacity of Hopfield network with **bipolar** neurons has been rigorously studied in [McEliece87]. For randomly selected P candidate state vectors synaptic matrix is built using **Hebb's** Learning Rule [Hebb49] (sum of outer product learning rule) given below

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^P \xi_i^{\mu} \xi_j^{\mu} \quad \forall i \neq j \text{ and } J_{ii} = 0 \quad \forall i = j$$

The definition of capacity given in [McEliece87] in terms of maximum number of candidate state vectors that can be made stable is

DEFINITION 2.5:- The asymptotic **capacity** (represented as P_{max}) of a Hopfield network is the maximum number of randomly chosen candidate state vectors P that are exactly recoverable by the Hopfield network when the limit of number of neurons N is infinity.

One step retrieval and Iterative retrieval

Input state vector $\xi^\mu(t = 0)$ represents the state of the network at time $t = 0$. If $\xi(t = 1)$ is taken as the output of the network using synchronous update it is known as *one step retrieval*. The input state vector is considered to be not more than ρN ($0 < \rho < 2$) bits away from a stored stable state. The asymptotic capacity estimates [McEliece87] under the condition that any given initial vector should always converge to a unique stored stable state with high probability is $P_{max} = \frac{(1-2\rho)^2}{4} \frac{N}{\log N}$.

The update process (synchronous or asynchronous) can be performed until $\xi(t+1) = \xi(t)$ before the output vector is retrieved. This retrieval is known as *fixed point retrieval* or iterative *retrieval*. In order that any given initial state vector should always converge to a unique fixed point, the asymptotic capacity is given by $P_{max} = \frac{N}{4 \log N}$.

Consider the relaxed condition that the unique stored stable state should be recovered with high probability, for a given initial state vector, except for a **vanishingly** small fraction of the fixed points. With this relaxed condition the capacity estimate reported in [McEliece87] is twice of that with rigid retrieval criterion. The asymptotic storage capacity for one-step retrieval is less than the storage capacity for iterative retrieval. For one-step retrieval there is a restriction on the state vectors which can be given as input state vector to the network. The storage capacity decreases as the number of possible initial state vectors increases.

2.5.5 Perfect Storage Condition

Estimates of the size of the set of P vectors to be completely determined by the **synaptic** matrix is reported in [Sussmann89]. For a given P candidate state vectors, **perfect storage** condition is that the synaptic matrix depends only on the set of $\bar{P}$, which consists of the candidate state vectors ξ^μ and their negatives. *Exactly recoverable condition* is that all

members of set $\bar{P}$ are the only local minima of the synaptic matrix. The *recoverable condition* is that the members of set $\bar{P}$ are contained in the set of local minima of the synaptic matrix. Perfect storage is a necessary but not sufficient condition for exact **recoverability**. It is expected that there will be a range of values of P for which some state vectors are perfectly stored but not exactly recoverable. The range for which perfect storage hold is much larger than the one for which recoverability is proved in [Hopfield82] and [McEliece87].

The capacity estimate reported in [Sussmann89] is given here. It has been proved that if P behaves like $\frac{N}{\kappa \log N}$, where κ is a constant and P candidate state vectors are randomly selected, then $\bar{P}$ is uniquely determined by the synaptic weights with high probability as N goes to infinity. The randomness of P is more general than the same given in [McEliece87] and [Hopfield82]. The P candidate state vectors are chosen with arbitrary probability distribution P_N . When the components of ξ^μ are independent and equal to $+1$ or -1 with probability $\frac{1}{2}$ then the value of K turns out to be $(2 \log 2)^{-1}$ i.e. 0.7. For perfect storage condition with the components of ξ^μ being $+1$ or -1 with probability $\frac{1}{2}$, asymptotic storage capacity is given as $P_{max} = \frac{N}{0.7 \log N}$.

This asymptotic storage capacity is higher than that given in Section 2.5.4 for relaxed retrieval condition. This **difference** is because the results of Section 2.5.4 are based on retrieval condition and the results given here are based on perfect storage condition. The probability condition is same for both the cases.

2.5.6 Upper and Lower Bounds of Capacity

In this section results of upper bound and lower bound on capacity reported in [Houselander86] are described.

DEFINITION 2.6:- Storage capacity is the maximum number of stable binary vectors

that can be stored on average.

The capacity of **Hopfield** networks trained using **Hebb's** learning rule is limited due to errors caused by the effective correlation noise of the initial state vector with each stable state vector, other than the stable state vector (exemplar) where we want the initial vector to stabilize. The upper and lower bound of the capacity are established on the assumption that the maximum error signal occurs when the correlation noise from each stable state vector is aligned and of the opposite polarity to the signal from the exemplar.

DEFINITION 2.7:- The *upper bound of capacity* is the number of stable state vectors required to guarantee **atleast** one error per network. The *lower bound of capacity* is the number of stable state vectors that can be stored before an error occurs.

For Hopfield network with N bipolar neurons, **synaptic** matrix constructed using Hebb's learning rule, and with synchronous update, the bounds on capacity are given as

$$P_{upper} \approx 1.25(N - 1)^{\frac{1}{2}} + 2r$$

$$P_{lower} \approx 0.75(N - 1)^{\frac{1}{2}} + 2r$$

where r represents the number of stable state vectors that are aligned with the exemplar.

A modified version of Hopfield network was also considered in [Houselander90]. In the original Hopfield network [Hopfield82] the diagonal elements of the synaptic matrix are zero. The Hopfield network is modified by allowing non-zero diagonal elements in the synaptic matrix. Diagonal elements of synaptic matrix is equal to number of candidate state vectors, i.e., $J_{ii} = P$.

The bounds of storage capacity for this modified Hopfield network of N bipolar neurons, with **synchronous** update are given as

$$P_{upper} \approx \{(N - 1) + 0.8(N - 1)^{\frac{1}{2}}\} \times \{0.8(N - 1)^{\frac{1}{2}} - 1\}^{-1} + 2r$$

$$P_{lower} \approx \{(N-1) + 1.4(N-1)^{\frac{1}{2}}\} \times \{1.4(N-1)^{\frac{1}{2}} - 1\}^{-1} + 2r$$

It can be observed that the capacity of this modified Hopfield network is greater than the capacity of Hopfield network discussed in Section 2.5.1.

2.5.7 Capacity Estimates For Delta Learning Rule

The definition of storage capacity, Delta learning rule, the numerical simulations and capacity estimates reported in [Prados89] are given here.

DEFINITION 2.8:- Storage capacity is the number of randomly selected candidate state vectors that can almost always be made fixed points in a Hopfield network.

The Delta learning rule for construction of synaptic matrix is based on the process of adjusting the synaptic weights to make a candidate state vector as a stable state of the network. A candidate state vector ξ^S is made stable state of a Hopfield network by adjusting the synaptic weights by using the equation $\Delta J_{ij} = \frac{1}{2}(\xi_i^S - \sigma_i)\xi_j^S$ where σ_i is calculated as

$$\sigma_i = \begin{cases} 1 & \text{if } \sum_{j=1, j \neq i}^N J_{ij}\xi_j^S > 0 \\ -1 & \text{if } \sum_{j=1, j \neq i}^N J_{ij}\xi_j^S < 0 \end{cases}$$

This procedure is repeated until the given state vector ξ^S becomes stable state of the network. But in the process of making ξ^S stable, other stable state vectors may be affected. Ideally, this procedure is repeated until all candidate state vectors become the stable states of the network. It is not possible to attain this condition for many sets of candidate state vectors. When such a situation is encountered the procedure is terminated after few iterations to avoid endless execution of this procedure.

In [Prados89] attempts to make P candidate state vectors as stable states of network are also reported. A new set of randomly generated P candidate state vectors is used in each attempt. The Delta rule is used to make the selected set of P candidate state vectors stable. Attempt is said to be successful if the selected set of P candidate state

vectors become stable states. P is gradually increased to observe the extent to which candidate state vectors can be made stable states.

At lower values of P more number of attempts are successful. As the number of candidate state vectors are increased keeping N fixed, the number of attempts that are successful, decreased. Ultimately, beyond a value of P the number of attempts that are successful either became zero or negligible. It is **found** that this critical value of P is always more than N . Thus it is possible to store more than N candidate state vectors in a network.

The capacity estimates in [Prados89] are better than those reported in [Hopfield82, Bruck88, McEliece87]. This can be attributed to the following reasons. The Delta learning rule takes more time in comparison to other rules like **Hebb's** learning rule to make a candidate state vector stable in the **Hopfield** network. Some capacity estimates [Bruck88, McEliece87] are applicable to any set of P state vectors, whereas [Prados89] deals with the several attempts to make randomly selected different sets of P state vectors as stable states. In Section 2.8 it has been discussed that the presence of correlation between the state vectors will increase the storage capacity. The random sets of P stable state vectors with $P > N$ may be the set of vectors with more correlation.

2.5.8 Presence of Spurious Stable States

In the previous sections the results on the capacity estimates are discussed without considering stable states. In this section the impact of spurious stable states on the storage capacity is considered. The performance of Hopfield network can be improved by exploring mechanisms to make more number of candidate state vectors stable or by reduction of spurious **stable** states. Continuous *unlearning algorithm* which enables continuous unlearning of the spurious stable states that goes along with the learning of candidate state vectors is reported in [Youn89]. In this algorithm the outer product of

a spurious stable state multiplied by the unlearning rate is subtracted from the synaptic weights. The purpose of the unlearning rate is to adjust the synaptic matrix in small steps to reflect the incremental unlearning.

By using this algorithm, the number of stored stable states are more and the number of spurious states are less compared to **Hebb's** learning rule and the Delta learning rule. The improvement is greater when the network is heavily loaded. With smaller unlearning rate more number of iterations are required to converge to a stable synaptic matrix. With large unlearning rate, some example enters in an infinite **loop** during the process of adjustment of synaptic weights.

2.5.9 **Hysteretic** Property

In Section 2.5.6 it is observed that the presence of non-zero diagonal elements (self-feedback character of neurons) in the synaptic matrix improves the storage capacity of **Hopfield** network. Hysteresis is the lag between releasing of stress and cessation of strain in materials subjected to magnetism. This property was introduced in neurons of neural networks in [Braham88]. **Self-feedback** in neurons is equivalent to **hysteretic** property in a discrete model. A neuron with self-feedback J_{ii} and hysteresis width b_i is equivalent to a neuron without self-connection and with hysteresis $b'_i (= b_i + J_{ii})$. The recalling ability of an associated memory network composed of **two-state** neurons with hysteretic property as investigated in [Yanai90] is described here.

Consider a Hopfield network with N bipolar neurons. Each neuron takes a value of +1 or -1 with probability $\frac{1}{2}$. Given P candidate state vectors to be made stable in the network the synaptic matrix is constructed using **Hebb's** learning rule. When hysteresis width of i^{th} neuron is b_i the next state of the i^{th} neuron σ_i at next time step $t + 1$ is

determined using the following equation

$$\sigma_i(t+1) = \text{sgn}\left(\sum_{j=1}^N J_{ij}\sigma_j(t) + b_i\sigma_i(t)\right)$$

In the presence of hysteretic property the response of a neuron is given in the Figure 4.1. Each neuron can have different hysteresis width. For simplicity the hysteresis width of all the N neurons is considered to be the same, i.e., b . For the present network

$$\alpha_c = \frac{(1+b)^2}{2\log N - \log\log N}.$$

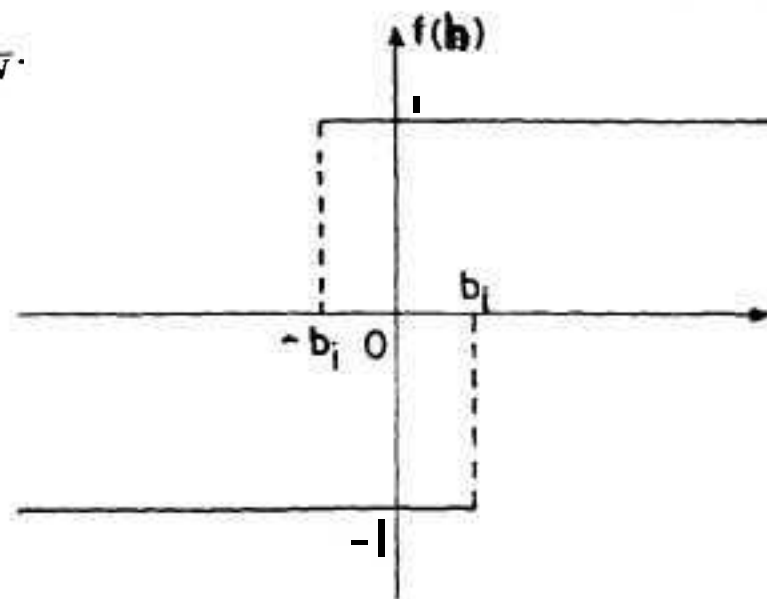


Figure 4.1: Response property of a neuron with hysteretic property.

The presence of hysteresis improves recalling ability. The reason for this improvement is due to the reluctance of the neurons to change state. When the state of the network is not far from a stable state, degradation of recalling process is caused by the neurons moving from right state to wrong state. With hysteretic property, neurons tend to stay in their current state. At the same time, there are neurons moving from the wrong state to the right ones. Hysteresis tends to prevent these changes and hence there is a trade-off. It is observed that the prevention of neuron state change from right state to wrong state is more effective than the prevention of state change from wrong state to a right state under proper conditions [Yanai90]. Numerical experiments also suggests that the basin of attraction is the largest when hysteretic width is close to the value of $\frac{P}{N}$.

2.6 ASYMMETRY IN HOPFIELD NETWORKS

Hopfield network [Hopfield82] has a synaptic matrix which is symmetric. One of the drawbacks of symmetric networks is the presence of spurious stable states. Much work has been done on the nature and origin of spurious states [Amit85a, Feigelman86]. A small amount of noise can be useful for escaping from spurious states. Such noise can be added by introduction of weak random asymmetry into a symmetric well functioning Hopfield network [Feigelman86]. However, such addition of noise does not change the qualitative performance very drastically, even at high level of asymmetry.

Hopfield network with asymmetric synaptic matrix can also be referred to as *Asymmetric Hopfield Network*. The dynamics of asymmetric Hopfield network has been discussed in [Derrida87, Crisanti88, Noest88a, Gardner89e, Mori89, Kree92]. Asymmetry in synaptic matrix can be introduced by (i) Asymmetric learning rule, (ii) Random changes, or (iii) Dilution.

1. **Asymmetric learning rule:-** There is no differentiation in the **post-synaptic** and **pre-synaptic** neuron in a symmetric learning rule. If **pre-synaptic** and post-synaptic neurons are **differentiated**, then the synaptic matrix will be asymmetrical.
2. **Random changes:-** In a symmetric synaptic matrix, asymmetry can be introduced by randomly selecting and changing the synaptic weights to some other value. Random selection of a pair of synaptic weights and swapping the values will also introduce asymmetry. The number of random changes depend on the degree to which asymmetry is to be introduced.
3. **Dilution:-** Dilution is a special case of random changes. Synaptic values are randomly selected from the symmetric synaptic matrix and the selected values are

changed to zero. These changes in **synaptic** matrix can be viewed as representing *Diluted Hopfield network* (network which are not fully connected). Study of Diluted Hopfield network is reported in [Derrida87, Wong91, Patrick92, Patrick90, Evans89, Derrida89, Kuhlmann92, Garces92]. Damage to a Hopfield network (*Damaged Hopfield network*) can also be considered as a case of dilution. Networks are able to adapt to the damage (up to a limit) introduced by dilution and are able to recover completely [Hendrich91]. This implies that associative memories using the Hopfield network paradigm are useful even in the presence of faults. Although dilution and asymmetry are two **different** and distinct concepts most of the **relevant** works reported in literature use dilution as a means of introducing asymmetry in Hopfield network. In other words, dilution is used to achieve asymmetry but symmetric dilution does not lead to asymmetry.

Capacity estimates of Asymmetric Hopfield networks where dilution has been used for introduction of asymmetry is given in Section 2.6.1. Capacity estimates of network constructed using asymmetric learning rule is given in Section 2.6.2.

2.6.1 Asymmetry and Dilution

An asymmetric and diluted version of Hopfield network as proposed in [Derrida87] is described here.

For bipolar neurons the synaptic weights are obtained as

$$J_{ij} = C_{ij} \sum_{\mu=1}^P \xi_i^{\mu} \xi_j^{\mu} \quad (2.1)$$

where $C_{ij} \in \{0,1\}$ is an independent random number which represents dilution and asymmetry. It is drawn from the distribution

$$Pr(C_{ij}) = \frac{c}{N} \delta(C_{ij} - 1) + (1 - \frac{c}{N}) \delta(C_{ij}) \quad (2-2)$$

Where c is a constant representing mean connectivity per neuron. $\delta(0) = 1$ and $\delta(x) = 0, x \neq 0$. Eq. 2.2 is a case of random dilution. The probability for a **synaptic** weight to remain intact is $\frac{c}{N}$. The probability for a synaptic weight to be made zero is $(1 - \frac{c}{N})$. Because $\frac{c}{N} \rightarrow 0$ as $N \rightarrow \infty$, the analysis is carried out under the **restriction** that $c \ll \ln N$. This is a case of *extreme dilution*. The neurons are updated **according** to the following rule

$$\sigma_i(t + \Delta t) = \begin{cases} +1 & \text{with probability } (1 + \exp[-2h_i(t)/T_0])^{-1} \\ -1 & \text{with probability } (1 + \exp[2h_i(t)/T_0])^{-1} \end{cases} \quad (2.3)$$

where $h_i(t) = \sum_{j=1}^N J_{ij} \sigma_j(t)$. Storage capacity α , at reduced temperature T is given by

$$\alpha = \frac{P - 1}{c} \quad \text{and} \quad T = \frac{1}{c} \quad (2.4)$$

At $T = 0$, the critical value of capacity $\alpha_c = \frac{1}{2} = 0.6366$. The α_c is larger than 0.15 for non-diluted symmetric cases. For $\alpha < \alpha_c$ two initial configurations close to stable state vectors, remain close to the stable state vector but do not become identical. When some of the stable state vectors are correlated there exists regimes for which the system remembers the state vectors, but cannot distinguish them. The extremely diluted network can retrieve the stable state vectors that have been stored in it. The storage capacity measured per remaining synapses, is significantly higher than that of fully connected synaptic matrix.

2.6.2 Asymmetric Learning Rule

This section describes an asymmetric version of **Hopfield** network using asymmetric learning rule reported in [Gardner89e].

The synaptic matrix for network with bipolar neurons is constructed using the modified **Hebb's** rule. The synaptic weight J_{ij} between the **post-synaptic** neuron i and **pre-synaptic** neuron j is changed only if the post-synaptic bit i of the candidate state vector

is active (+1). The learning rule has asymmetric **synaptic** matrix as it differentiates between **post-synaptic** and **pre-synaptic** neurons. In a candidate state **vector**, if ξ_i is +1 and ξ_j is -1, then this pair of neurons contribute to reduce the **value** of J_{ij} but J_{ji} is not affected.

By introducing the dilution and asymmetry parameter $C_{ij} \in \{0, 1\}$ the final synaptic weights are obtained as $J_{ij} = C_{ij} \bar{J}_{ij}$, where $\bar{J}_{ij} = \frac{1}{2} \sum_{\mu=1}^P (\xi_i^\mu + 1) \xi_j^\mu$. The argument for extreme dilution in Section 2.6.1 are also applicable here. Every neuron $\sigma_j = 1$ contributes to the post-synaptic potential of neuron i

$$MO = \sum_{j=1, (j \neq i)}^N J_{ij} (\sigma_j(t) + 1)$$

If the post-synaptic potential exceeds a threshold θ_i then the post-synaptic neuron is activated. For simplicity uniform threshold is considered for all neurons, $\theta_i = \theta_0 > 0$.

$$\sigma_i(t + \Delta t) = \text{sgn}(h_i(t) - \theta_0)$$

Updating procedure is given by

$$\sigma_i(t + \Delta t) = \begin{cases} 1 & \text{with probability } [1 + \exp(-2h_i(t) - 2\theta_0)]^{-1} \\ -1 & \text{with probability } [1 + \exp(2h_i(t) - 2\theta_0)]^{-1} \end{cases}$$

The capacity of the network is found to depend on the **threshold** of the post-synaptic neuron. Capacity is optimal for $\theta_0 \approx 0.1$ and no retrieval is possible for $\theta_0 > 0.5$. Under the condition $c \rightarrow 0$, $N \rightarrow \infty$, $P \rightarrow \infty$, with reduced temperature $T = \frac{T_0}{c}$ and reduced threshold $\theta = \frac{\theta_0}{c}$ the capacity with respect to existing couplings is obtained as

$$\alpha_c = \frac{P - 1}{c} \quad (2.5)$$

This value of critical capacity is the same as that given in **Eq. 2.4**. The capacity of this model varies with threshold of the post-synaptic neuron. This network classifies input state vectors according to their mean activity and their overlap with the stable state vectors.

2.7 BIASED CANDIDATE STATE VECTORS

The dynamics of Hopfield network is studied by selection of candidate state vectors randomly or in a deterministic way. For **two-state** neurons, the randomly selected candidate state vectors on an average have 50% active neurons and 50% passive neurons. These vectors are termed as *unbiased candidate state vectors*.

The candidate state vectors for which mean percentage of active elements is different from 50% are called *biased candidate state vectors*. In deterministic way the candidate state vectors with a particular activity level or bias can be chosen. All candidate state vectors with a specific bias are naturally correlated with each other. Such candidate state vectors are called *correlated candidate state vectors*.

The *bias parameter* or *activity level* a of a state vector ξ^μ is given by $a = \frac{1}{N} \sum_{i=1}^N \xi_i^\mu$. The biased candidate state vectors with low level of activity are known as *sparse candidate state vectors*. The total number of active elements in a candidate state vector is called as **magnetization M** of a candidate state vectors. Ratio of active neurons to the total number of neurons is known as *magnetization per spin* of a candidate state vector.

In Hopfield networks with N , **two-state** neurons the biased and unbiased state vectors in terms of activity level are given in Table 2.1. This table also gives the number of active elements and excess of active neurons over passive neurons in state vector with bias parameter a . Study of neural networks with biased candidate state vectors have been reported in [Evans89, Amit87, Viswanathan93, Penna90, Tsodyks88].

Table 2.1 : The activity level a and bias of state vectors		
	binary neurons	bipolar neurons
Unbiased state vector	$a = \frac{1}{2}$	$a = 0$
State vector with fully active bias	$a = 1$	$a = 1$
State vector with fully inactive bias	$a = 0$	$a = -1$
Number of active elements in state vector with bias parameter a	aN	$\frac{(1+a)N}{2}$
Excess of active neurons over passive neurons in state vector with bias a	$(2a - 1)N$	aN

2.8 CORRELATED CANDIDATE STATE VECTORS

A mechanism for selection of candidate state vectors is based on common descriptor. Only those state vectors with certain description are eligible for selection as candidate state vectors. *Correlation m* between two candidate state vectors ξ^μ and ξ^ν is defined as

$$m = \frac{1}{N} \sum_{i=1}^N \xi_i^\mu \xi_i^\nu$$

For bipolar neurons $-1 < m < 1$ and for binary neurons $0 < m < 1$. The correlation between a set of P candidate state vectors can be represented in a **correlation matrix**

$$(C_{\mu\nu}) : C_{\mu\nu} = \frac{1}{N} \sum_{i=1}^N \xi_i^\mu \xi_i^\nu, \quad \mu, \nu = 1, 2, \dots, P \quad (2.6)$$

Semantic correlation is the correlation between various candidate state vectors. Syntactic correlation is the correlation between neuronal sites. Hierarchical correlation deals with candidate state vectors grouped into clusters.

Storage and retrieval of correlated candidate state vectors reduce the number of possible candidate state vectors that can be made stable. The space of interaction is also

reduced. The number of candidate state vectors that can be stored in a Hopfield network with N neurons is more for correlated candidate state vectors compared to **uncorrelated** candidate state vectors. However the information capacity of the network is less for correlated candidate state vectors compared to uncorrelated candidate state vectors.

2.9 CAPACITY WITH BIASED AND CORRELATED PATTERNS

2.9.1 Low Level Activity With Bipolar Neurons

In [Amit87] a modified Hopfield network to allow the storage of biased candidate state vectors is proposed. This work deals with the study of associative memory whose mean activities differ from 50%.

State of every element ξ_i^μ in the candidate state vectors can be chosen independently with probability $P(\xi_i^\mu) = \frac{1}{2}(1+a)\delta(\xi_i^\mu - 1) + \frac{1}{2}(1-a)\delta(\xi_i^\mu + 1)$. The average of each ξ is a ($\langle \xi_i^\mu \rangle$) and the mean activity in each candidate state vector is $\frac{1}{2}(1+a)$, $-1 < a < 1$. With such a distribution the candidate state vectors are necessarily correlated in a simple way i.e. $\langle \xi_i^\mu \xi_j^\mu \rangle = a^2$.

The stored stable states become unstable at very low storage level. Even at small values of the bias parameter a , the Hopfield dynamics is catastrophic. This is due to the fact that the noise generated by other stable states in the retrieval of each stored stable state does not average to zero. To overcome this difficulty the **synaptic** weights are derived by the following non-local learning rule

$$\bar{J}_{ij} = \frac{1}{N} \sum_{\mu=1}^P (\xi_i^\mu - a)(\xi_j^\mu - a)$$

These synaptic weights avoid catastrophe by shifting the noise back to a zero mean. For finite P , the consequences of traditional dynamics are unsatisfactory. The **near** saturation ($P = \alpha_c N$) spurious states are found to dominate energy landscape. **Although,**

the stored stable states themselves are stable below $\alpha_c(a)$ (upto a small fraction of errors), their basin of attraction, and hence fault tolerance of the network decrease sharply with increase in a .

To have a network which can effectively store and retrieve candidate state vectors, it is not only sufficient to modify the **synaptic** weights but also to modify the dynamical process. With mean bias parameter a , the network is constrained so that it wanders mostly among states that have the preferred mean activity. There must be global control on the dynamics of the network which prevent too high or too low activity. Thus the network activity should be much lower than 50% whether or not the network is retrieving. This control restricts the region in state space in which a healthy neural network can move. The dynamics is restricted rigidly to states with a given value $M = \sum_{i=1}^N \sigma_i = Na$. The number of neurons that are on (+1) is $\frac{(1+a)}{2}$. The candidate state vectors are stable states of the network. Spurious states with macroscopic overlaps with small number of candidate state vectors, do not appear. Such a network has a higher storage capacity $\alpha_c(a)$ than that of network storing random or unbiased candidate state vectors $\alpha_c(a) > \alpha_c(0)$ for $|a| < 0.99$.

2.9.2 Low Level Activity With Binary Neurons

Stability of candidate state vectors with low level of activity in **Hopfield** network with binary neurons is reported in [Tsodyks88]. The main variation is the use of binary neurons instead of bipolar neurons as in Section 2.9.1. Maximal storage capacity **obtained** for this model using the meanfieldtheory is $\alpha_c \simeq \frac{1}{2a}$.

2.9.3 Improving Retrieval

A method for retrieving **information** from Hopfield network storing candidate state vectors with low **level** activity is proposed in [Penna90]. This method uses the traditional

Hopfield dynamics by doubling the number of neurons in a suitable manner. For P candidate state vectors with bipolar neurons the **synaptic** matrix is constructed by using **Hebb's** learning rule. The states of neurons is updated using the equation

$$\sigma_i(t+1) = \text{sgn}\left(\sum_{j=1, (j \neq i)}^N J_{ij} \sigma_j(t)\right)$$

If the correlation between any two candidate state vectors is not close to zero i.e., $|m| < \frac{1}{\sqrt{N}}$, they do not represent fixed points of dynamics. The dynamics of Hopfield network Cannot distinguish between two similar candidate state vectors for which correlation m is not close to zero. In order to satisfy the condition of **vanishing correlation** a necessary condition is that all candidate state vectors must have about 50% of active neurons. But in [Amit87] it has been shown that this may be undesirable in most cases where it is required to reach retrieval for low activity candidate state vectors.

The method of retrieving information from synaptic matrix storing low level activity candidate state vectors as reported in [Penna90] is given below.

1. Candidate state vectors with $2N$ neuron are created by doubling the initial number of neurons such that $\zeta_{2i}^\mu = \zeta_{2i+1}^\mu + \xi_i^\mu$, where ζ_i^μ is the i^{th} neuron state of the doubled candidate state vector μ .
2. A neuron $\xi_i^\mu = -1$ is randomly selected from the original network.
3. One of the corresponding ζ_{2i}^μ or ζ_{2i+1}^μ neurons of the doubled candidate state vector is randomly chosen and then its state is changed to active state.

The Steps 2 and 3 are repeated until 50% components of the doubled candidate state vectors are active for all the candidate state vectors. Using this prescription P candidate state vectors with low and different activity levels are stored and it is possible to retrieve these vectors. The storage capacity is found to be two times the capacity

given in [Hopfield82]. This is due to hidden neurons. It was also found that smaller the fraction of active neurons the better is the quality of retrieval.

2.9.4 Improving Performance By Adaptive Threshold

A modified Hopfield model with adaptive neural threshold and global inhibitory interaction between neurons is proposed in [Buhmann89]. This network is capable of near optimal storage of candidate state vectors at low activity. It also exhibits a high (close to optimal) storage capacity, no spurious states and a special state of no recognition. By adjusting threshold a choice can be made between effective storage and good associativity.

The P candidate state vectors with bias a are chosen according to the distribution $P(\xi_i^\mu) = a\delta(\xi_i^\mu - 1) + (1-a)\delta(\xi_i^\mu)$. The synaptic weights between binary neurons are chosen according to Hebb's hypothesis of co-operating neuron assemblies [Hebb49] (assembly of neurons with mutual excitatory interactions). There is a competition among such assemblies (inhibitory interaction between neurons belonging to different assemblies). The synaptic weights are derived using the equation

$$J_{ij} = \frac{1}{a(1-a)N} \sum_{\mu=1}^P (\xi_i^\mu - a)(\xi_j^\mu - a) - \frac{\gamma}{aN}, \quad i \neq j$$

where, γ is the inhibition factor.

The synaptic matrix is symmetric and diagonal elements are zero. The neurons are updated asynchronously according to probabilistic rule. The rule is based on a local field $h_i = \sum_{j=1}^N J_{ij}\sigma_j$. With probability $f_i = \{1 + \exp[-(h_i - \theta)/T]\}^{-1}$ neuron i fires at time $t + \Delta t$, otherwise it is quiet. θ is the threshold and T is the network temperature.

At $T = 0$ and a parameter range defined by $0 + \gamma < 1 - a$, all states exhibiting macroscopic overlap with one candidate state vector are stable states. In the limit of small a value ($a \ll 0.1$) for sufficiently strong inhibition ($\gamma > \gamma_c = \frac{1-a}{2}$) the network states are always unstable. For a low level of activity a the storage capacity increases as

$\alpha_c = -a(\ln a)^{-1}$. The information stored per synapses could be as large as 0.38 for $\rho = 0.75$, $a = 5.6 \times 10^{-7}$.

2.9.5 Non-local Learning Rules

The local learning rules have some limitation. While using these rules it may either be difficult or not possible to deal with correlated candidate state vectors. When the candidate state vectors are uncorrelated random variables, the combined overlap of a candidate state vector with all other candidate state vectors is of $O(\sqrt{\frac{P}{N}})$. Hence when P is finite, the stable states of a **Hopfield** network contains a finite fraction of errors, which increases with a [Amit85a]. When α reaches a critical value α_c , there is a dramatic increase in the level of errors and the Hopfield network ceases to function as an effective associative memory system. The critical value α_c depends on details of the model but it is always less than 0.14 [Amit85a, Sompolinsky86]. Non-local learning rules can suppress the adverse effects of the overlaps among the candidate state vectors [Personnaz85, Kanter87].

Pseudo-inverse Learning Rule

In [Personnaz85] a model based on non-local learning rule called *Pseudo-inverse learning rule* is proposed. This model is capable of storing correlated or uncorrelated state vectors which are linearly independent. The **synaptic** weights are given by

$$J_{ij} = \frac{1}{N} \sum_{\mu, \nu=1}^P \xi_i^\mu \xi_j^\nu (C^{-1})_{\mu\nu} \quad (2.7)$$

where $(C^{-1})_{\mu\nu}$ is inverse of correlation matrix given in Eq. 2.6.

The local field of a neuron i is defined as

$$\bar{h}_i = \sum_{j=1}^N J_{ij} \sigma_j \quad (2.8)$$

The self-coupling term in Eq. 2.8 restricts severely the size of basins of attraction of the stable state vectors especially for large a . For small values of $a < \frac{1}{2}$, the self-coupling term significantly reduce the basins of attraction.

Above $a = \frac{1}{2}$ a state vector configuration which differs from a stored stable state vector by one spin will not flow to that stable state vector. Although the candidate state vectors can be made stable upto $a = 1$, the maximum capacity of the system for providing associative memory is $\alpha_c = \frac{1}{2}$.

Eliminating Self-Coupling Term

A modification of the above model by eliminating self-coupling term is proposed in [Kanter87]. The local field of neuron i is given by $h_i = \sum_{j=1, (j \neq i)}^N J_{ij} \sigma_j$. The neurons are updated using the equation $\sigma_i = \text{sgn}(h_i)$.

For synchronous as well as asynchronous update the radius of attraction R decreases monotonically and vanishes at $\alpha_c = 1$. There is an increase in the number of spurious stable states as a increases. But the occurrence of linear combinations of stable state vectors as spurious states is very rare. Thus the presence of spurious states do not affect the basin of attraction of the stored stable state,

2.9.6 Generalised Learning Rule

In [Dotsenko91] a generalised model based on generalised learning rule has been proposed. Hebb's learning rule [Hebb49] and Pseudo-inverse rule [Kanter87, Personnaz85] are two special cases of this generalised rule. This subsection deals with model proposed in [Dotsenko91] and capacity estimates of the model.

Network consists of N bipolar neurons. Synaptic matrix is constructed using the equation

$$J_{ij} = \frac{1}{N} \sum_{\mu, \nu=1}^P \xi_i^\mu (\mathbf{I} + \lambda C)^{-1}_{\mu\nu} \xi_j^\nu \quad (2.9)$$

where $C_{\mu\nu}$ is correlation matrix given by Eq. 2.6.

With $A = 0$, this generalised rule corresponds to **Hebb's** learning rule. In the limit $A \rightarrow \infty$ the structure of J tends to the synaptic matrix constructed using **Pseudo-inverse** learning rule. The extreme values of $\alpha_c(\lambda)$ are $\alpha_c(\lambda = 0) = 0.14$ and $\alpha_c(\lambda \rightarrow \infty) = 1.07$.

2.9.7 Biased State Vectors In Clipped Network

This section describes a neural network model with bipolar synaptic weights $\{+1, -1\}$ reported in [Viswanathan93].

The condition for each of the P candidate state vectors with bias a , to be fixed point in a network with N neurons is

$$\Upsilon_i^\mu = \xi_i^\mu \sum_{j=1}^N \frac{J_{ij} \xi_j^\mu}{\sqrt{N}} > \kappa$$

The neurones updated **asynchronously** using the update rule $\sigma_i(t+1) = \text{sgn}(\sum_{j=1, (j \neq i)}^N J_{ij} \sigma_j(t))$.

It is further assumed that the synaptic couplings which each neuron receives are constrained to have a bias $\sum_{j=1}^N J_{ij} = r\sqrt{N}$. The number r (bias of coupling) **measures** the excess of excitatory couplings over inhibitory couplings. The storage capacity as a function of coupling bias for $\kappa = 0$ and the pattern bias a is given in Figure 4.2. The Figure 4.3 maps the capacity as a function of coupling bias for $a = 0.6$ with κ values 0, 1, 2.

It can be **concluded** that for every value of κ , the storage capacity is optimal for a value of bias in the couplings which is independent of bias in the candidate state vector, as long as a is non-zero. For random candidate state vectors ($a = 0$) the critical capacity $\alpha_c = 0.83$ is independent of coupling bias r . For $K = 0$, $r = 1$ (Figure 4.2) is an optimal value for the coupling bias. The optimal value of r increases slightly to about 2 as K increases to 2 (Figure 4.3). The peaks in the storage capacity is sharper for candidate state vectors which are most severely biased, indicating that only a narrow range of coupling bias values store these vectors efficiently.

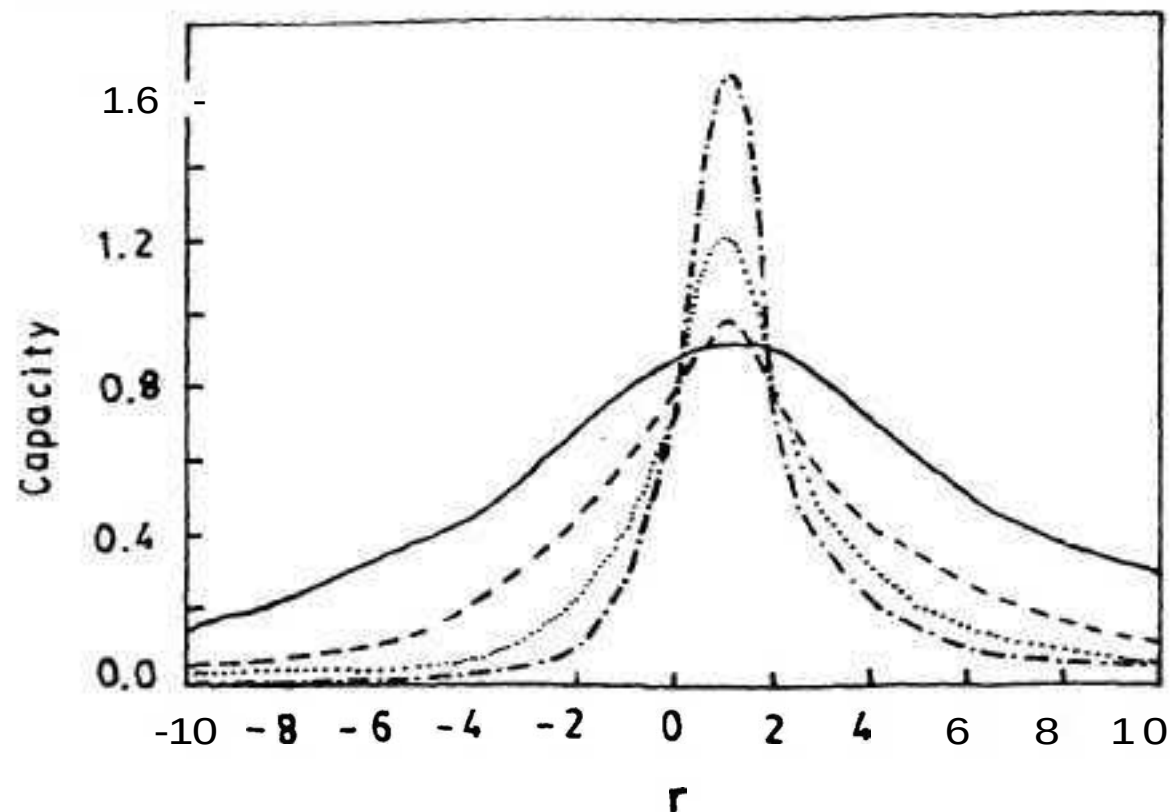


Figure 4.2: Capacity as a function of r for $\kappa = 0$ and $a = 0.2, 0.4, 0.6$ and 0.8 . The higher peaks are for larger values of a .

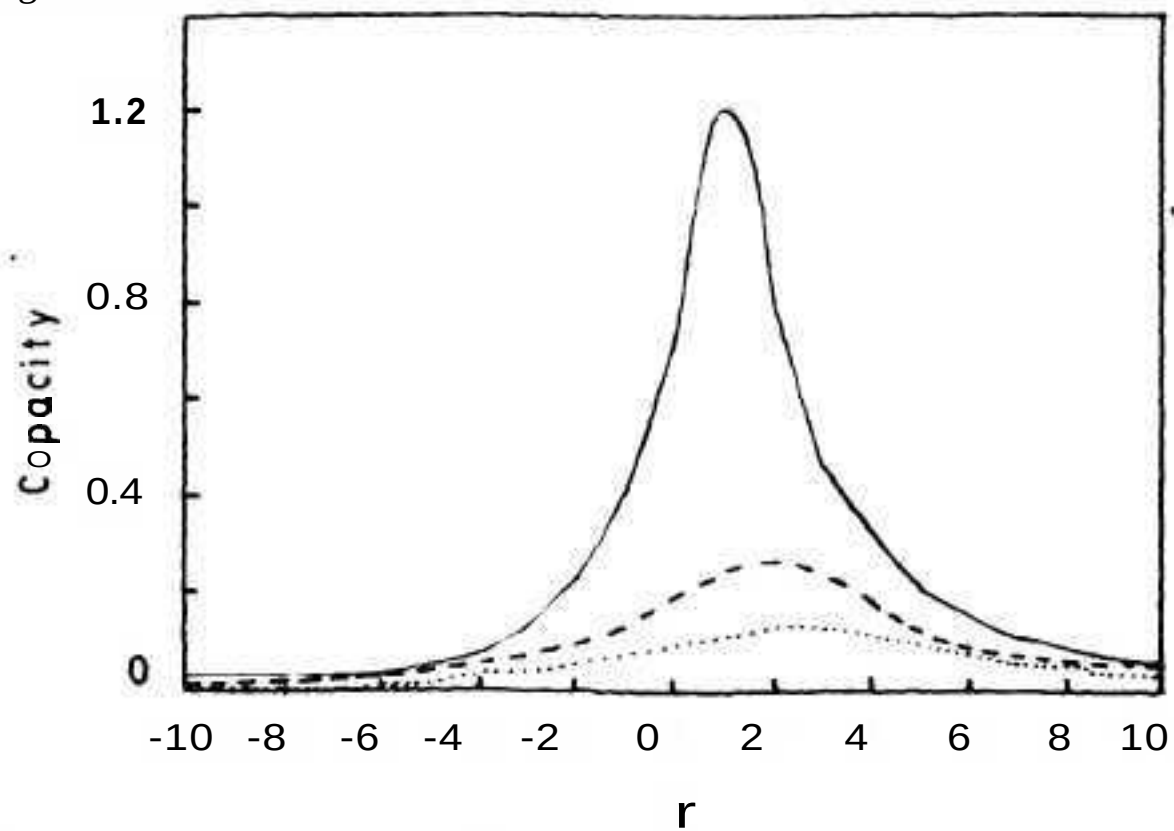


Figure 4.3: Capacity as a function of r for $K = 0, 1$, and 2 . The upper curves are for smaller values of K .

2.9.8 Generalised Rule For Correlated Patterns

In [Der92] a model has been proposed, which enables storage of strongly correlated candi-

couplings, with an intermediate form between Hebb's learning rule and **Pseudo-inverse** rule described in Section 2.9.5. This model which combines the probability density function for generation of candidate state vectors (Section 2.9.1 and Section 2.9.4) with the generalised learning rule (Section 2.9.6) is described in this section.

The ξ^μ (with bias a) are chosen by probability distribution $P(\xi) = a\delta(\xi - (1 - a)) + (1 - a)\delta(\xi + a)$, with $\langle \xi \rangle = 0$ and $\langle \xi^2 \rangle = a(1 - a)$. The **synaptic** matrix is constructed using the formulation given in Eq. 2.11.

In the limit $A \rightarrow 0$ this model resembles **Hopfield** network storing correlated **candidate** state vectors. In the limit $A \rightarrow \infty$, this model resembles the **Pseudo-inverse** version of the model storing strongly correlated candidate state vectors. The threshold is considered to be same for all neurons. The next state of the neuron i is obtained as

$$\sigma_i(t+1) = \begin{cases} 0 & \text{if } \sum_{j=1, (j \neq i)}^N J_{ij} \sigma_j(t) - \theta_i \geq 0 \\ 1 & \text{if } \sum_{j=1, (j \neq i)}^N J_{ij} \sigma_j(t) - \theta_i < 0 \end{cases}$$

By appropriate adjustment in the threshold θ , optimal capacity can be obtained. The optimal value of θ at which maximum capacity is reached is given by

$$\theta = \frac{a(1-a)(1+\lambda a)}{(1 + \frac{\lambda}{2 \ln(\frac{1}{a})})^2}$$

At this value of θ the capacity, $\alpha_c \approx \frac{1}{2} * \frac{1}{\ln(\frac{1}{a})}$ for $a \ll 1$. Thus the critical capacity of this model does not depend on A , but the optimal value of the threshold depends on it. The results of this section and Section 2.9.6 show that generalised learning rule is very rich in its behaviour and theoretically very robust. For storing **uncorrelated** candidate state vectors critical capacity can be increased from $\alpha_c = 0.14$ (at $A = 0$) upto $\alpha_c = 1$ ($A \rightarrow \infty$). For model storing correlated candidate state vectors the **critical** capacity can reach a value mentioned above for any value of the parameter λ , provided the threshold is chosen in an optimal way.

2.9.9 Existence of Required Network

The emphasis of the capacity consideration in this chapter till now has been on the number of candidate state vectors that can be stored and retrieved with a given prescription for construction of **Hopfield** class networks. There is another approach to the study of storage capacity with emphasis on the availability of a **synaptic** matrix which will ensure stability conditions for the given set of candidate state vectors.

In [Baldi87] the maximum number of candidate state vectors that can be stored in the Hopfield networks is given by

$$P_{max} \approx K_2 2^{C_2 N} \quad (2.10)$$

where the values of $K_2 = 1.0505$ and $C_2 = 0.2874$. This will give a $P_{max} = 2N$. For correlated candidate state vectors, each with bias a , where $1 - a \sim \frac{1}{N}$, there is a prescription where P_{max} is of the order of $\frac{N^2}{(\ln N)^2}$.

In [Gardner88a] the study of space of **interactions** in the Hopfield network is studied in detail. The synaptic weights are considered as dynamical variables. The synaptic weights need not be explicitly prescribed in terms of candidate state vectors. A mechanism to estimate the storage capacity for optimal network is **devised**. The task is to choose synaptic weights such that P prescribed candidate state vectors are the fixed points in a Hopfield network with **bipolar** neurons. The dynamics of the system is defined by

$$\sigma_i(t+1) = \text{sgn}(h_i(t) - \theta_i)$$

where

$$h_i(t) = \frac{1}{\sqrt{N}} \sum_{j=1, (j \neq i)}^N J_{ij} \sigma_j(t) \quad (2.11)$$

The synaptic weights $J_{ij} \neq J_{ji}$. The synaptic weight J_{ij} are defined so that $\sum_{j=1, (j \neq i)} J_{ij}^2 = N$ at each site i . The **neuronal** configuration σ_i is thus a fixed point of dynamics provided the quantity $h_i = \sigma_i(h_i\{\sigma_j\} - \theta_i)$ is positive for all sites i .

The requirement that each candidate state vector is a fixed point is not sufficient to guarantee a finite basin of attraction and a stronger condition

$$\sum_j (x_j - \theta_j) > \kappa \quad (2.12)$$

where κ is a positive constant, which is imposed at each site i and for each candidate state vector. Larger values of K imply larger basins of attraction.

The typical fractional volumes of the space of solutions for the **synaptic** weights J_{ij} to Eq. 2.12 and Eq. 2.11 is calculated. The volume vanishes above a value α_c , which depends on the stability K and this determines the maximum storage capacity of the network.

For **uncorrelated** candidate state vectors, the thresholds θ_i are set to zero. For $AC = 0$, the volume vanishes as a increases towards 2. This determines the maximum storage capacity $\alpha_{c(max)} = 2$. This is in agreement to results of [Baldi87] Eq. 2.10. The upper storage capacity $\alpha_c(\kappa)$ is found to decrease with AC . That is as AC increases and basins of attraction become larger, then the number of stored stable states decreases.

These calculations are repeated for correlated candidate state vectors with same activity level. The ξ_i^μ are independent random variables with distribution $P(\xi_i^\mu) = \frac{1}{2}(1+a)\delta(\xi_i^\mu - 1) + \frac{1}{2}(1-a)\delta(\xi_i^\mu + 1)$. For $AC = 0$ and small values of a , the maximum storage capacity $\alpha_{c(max)} = 2(1 + \frac{2a^2}{3} + O(a^4))$. As a tends to 1, α_c diverges as

$$\alpha_c = -\frac{1}{(1-a)\ln(1-a)}.$$

Although the storage capacity increases with correlation between candidate state vectors, the amount of information per candidate state vector decreases.

DEFINITION 2.9:- The total information capacity I , is the total number of bits stored in the candidate state vectors.

Total information capacity can be measured using the following equation.

$$I = \frac{N^2}{\ln 2} \alpha_c(a) \left\{ \frac{1}{2}(1-a) \ln\left(\frac{1}{2}(1-a)\right) + \frac{1}{2}(1+a) \ln\left(\frac{1}{2}(1+a)\right) \right\}$$

For random candidate state vectors $a = 0$ $I = 2N^2$. The information capacity I , however decreases slightly with a . For small a , $I \approx \frac{N^2}{2 \ln 2} \approx 0.721 N^2$.

The information capacity of a set of candidate state vectors decreases as the candidate state vectors become more correlated. A given information cannot be learnt any faster by introducing redundancy [Wendemuth93]. There is no learning time gain in spreading the same information over a large set of correlated candidate state vectors.

In [Gardner88b] it is shown that to go beyond the storage capacity of optimal network it is necessary to allow a minimal fraction f of bit errors. For each value of a and AC there is a minimum fraction f_{min} of wrong bits.

2.9.10 Semantic and Spatial Correlation

The correlation between candidate state vectors has an impact on the synaptic weights. This section deals with semantic correlation and spatial correlation between candidate state vectors'

Semantic correlation:- Correlation between different candidate state vectors to be made stable is called semantic correlation. Semantic correlation between two candidate state vectors ξ^μ and ξ^ν is of the form

$$\langle \xi_i^\mu \xi_i^\nu \rangle = C_{\mu\nu} \delta_{ij} \quad \forall (i, j), \quad \forall (\mu, \nu)$$

where $C_{\mu\nu}$ is the correlation matrix which is formulated using Eq. 2.6.

The correlation of a pattern with itself $C_{\mu\mu} = 1$. Correlation between two candidate state vectors ξ^μ and ξ^ν can be calculated as the Hamming distance between the two candidate state vectors, $C_{\mu\nu} = \frac{1}{N} \sum_i |\xi_i^\mu - \xi_i^\nu|$. It has been shown in [Tarkowski92, Wendemuth93] that for semantic correlation the critical capacity do not depend on the type of distribution in correlation matrix but depends only on the difference in the maximum and minimum value $(C_{max} - C_{min})$ in the correlation matrix.

Spatial correlation:- Correlation among the different sites of the network is called spatial correlation. Spatial correlation is of the form

$$\langle f_i f_j \rangle = C_{ij} \delta_{\mu\nu} \quad \forall (i,j), \quad \forall (\mu,\nu)$$

It has been shown in [Wendemuth93, Lewenstein92] that in the case of storage of the spatially correlated candidate state vectors in the Hopfield network the critical capacity ratio α_c somewhat exceeds $\alpha_c(\kappa=0) = 2$.

2.9.11 Hierarchical Correlation

The organisation of objects with well defined relations of similarity into hierarchical tree arises naturally in many cases of data classification and analysis. Such objects are said to be hierarchically correlated. Attempts to incorporate such a structure in a neural network have been reported in [Toulouse86]. Hopfield model has been extended in [Gutfreund88] to allow the storage and retrieval of hierarchically correlated candidate state vectors. The overlap between these candidate state vectors form a hierarchical tree. The proposal given in [Gutfreund88] is described here.

The hierarchical tree of candidate state vectors is constructed as follows. At the first level of hierarchy P_1 candidate state vectors $\xi^\mu (\mu = 1, 2, \dots, P_1)$, with bias a are generated. Every component ξ_i^μ is chosen independently with the probability given by

$$P(\xi_i^\mu) = \frac{1}{2}(1+a)\delta(\xi_i^\mu - 1) + \frac{1}{2}(1-a)\delta(\xi_i^\mu + 1) \quad \forall \xi_i^\mu$$

These candidate state vectors serve as ancestors for the next level. At the second level a new correlation parameter ϕ , ($0 < \phi < 1$) is specified. For each of the P_1 candidate state vector ξ^μ , P_2 descendants $\xi^{\mu\phi}$ ($\phi = 1, 2, \dots, P_2$), are generated. The candidate state vectors are grouped into the clusters with high correlation between candidate state vectors within the same cluster and lower correlation between candidate state vectors in different clusters.

To retrieve a particular candidate state vector ξ^{μ} , attempt is made to identify the ancestor. The information about the ancestor is transferred to the network as an external field h_i on each neuron by $h_i = h \xi_i^{\mu}$ where h is an external field conjugate to the ancestor state. It is observed that critical storage capacity is reached at value of h dependent on b . The value of b should not be too small, so that the overlap with the ancestor candidate state vector is sufficiently large. a should not be too large, to ensure a clear separation from the basins of attraction of the other ancestor candidate state vectors.

2.10 MARKED CANDIDATE STATE VECTORS

Hebb's learning rule allows the storage of P candidate state vectors ξ , where each component takes on one of the two values with equal probability. It is reported in [Amit89] that in the limit $N \rightarrow \infty$, it was possible to retrieve the candidate state vectors, with less than 3% error, as long as $P < 0.1387N$. Beyond this value it is not possible to retrieve the stored stable states. This is called *blackout catastrophe*.

It has been argued that such a behaviour is not realistic and modifications of the original Hebb's learning rule, to avoid blackout catastrophe have been subjected to many studies [Parisi86a, Nadal86, Nicolis90, Fontanari88]. In [Fontanari88] it has been shown that by marking a finite number of candidate state vectors it is possible to retrieve them, even when the unmarked candidate state vectors exceed $0.138N$. By marking candidate state vectors the network is modified in such a way that even if the network is overloaded ($\alpha > \alpha_c$) it will have marked candidate state vectors as stable states. In this approach, however, blackout is not avoided, but delayed.

Privileged status for a set of candidate state vectors can be introduced within the Hebb's learning rule. In [Nicolis90] it is proposed that set of **candidate** state vectors P is categorized into two groups, P_1 marked candidate state vectors ξ^{μ} ($\mu = 1, 2, \dots, P_1$),

with privilege status having an associated weight one and P_2 unmarked candidate state vectors ζ^ν ($\nu = 1, 2, \dots, P_2$), having a weightage γ ($0 < \gamma < 1$). The following equation is used to determine the synaptic weights.

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^A \zeta_i^\mu \zeta_j^\mu + \frac{\gamma}{N} \sum_{\nu=1}^{P_2} \zeta_i^\nu \zeta_j^\nu$$

α_1 is finite as $N \rightarrow \infty$ and $P_2 = \alpha_2 N$. The terms α_1 and α_2 represent the storage capacity of marked and unmarked candidate state vectors respectively. For simplicity $\alpha_1 = \alpha_2 = \alpha$.

The conclusion of [Nicolis90] is that under the conditions mentioned above the critical capacity is lower than that reported in [Fontanari88]. Since $2\alpha_c > 0.138$, total blackout is delayed. The critical capacity for retrieval of marked and unmarked candidate state vectors as a function of γ is given in Figure 4.4. It can be observed that for a given value of γ , there corresponds two critical values of α i.e. α_P and α_μ . For $\alpha < \alpha_\mu$, the network is capable of retrieving the unmarked as well as marked candidate state vectors. For $\alpha_\mu < \alpha < \alpha_P$, the network may retrieve only the marked candidate state vectors. Above $\alpha > \alpha_P$, the network cannot retrieve any candidate state vector, it is a blackout. Thus the learning rule proposed in [Nicolis90] enables storage and retrieval of certain candidate state vectors beyond the well established critical storage capacity.

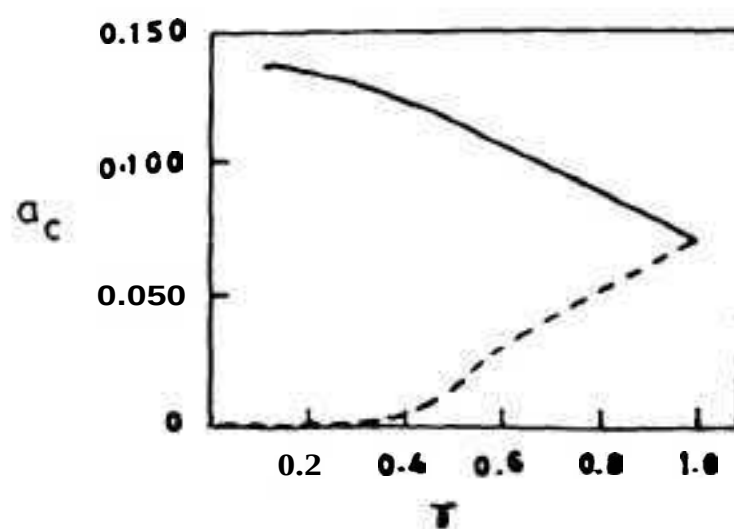


Figure 4.4: Critical capacity α_c for retrieval of marked (solid line) and unmarked (dashed line) candidate state vectors as a function of γ .

2.11 MULTI-VALUED NEURAL NETWORKS

In the previous sections **Hopfield** class networks with **two-state** neurons are considered. Although this choice is often natural and useful, there are several networks with more than two neuron states (*multi-valued neural states*) reported in literature [Kanter88, Bolle91, Yedidia89, Bolle92, Prados, Stiefvater92, Baram91, Rieger90, Noest88a]. Multi-valued neurons can be 1) Multi-state neurons or 2) Continuous valued neurons.

2.11.1 q -State Potts model

The q -state Potts model [Elderfield83] has been introduced in the theory of neural network in [Kanter88]. In this model state of each neuron σ_i is viewed as a Potts glass spin. The neuron can take any one of the q values $1, 2, \dots, Q$, where Q is any integer greater than 1. The state of the network is the instantaneous configuration of all the neurons at a given time. Neural networks based on q -state Potts Spin glass neuron have been studied in [Kanter88, Vogt92, Bolle91, Elderfield83]. The synaptic weight J_{ij}^{kl} determines the contributions of a signal fired by the j^{th} pre-synaptic neuron in state k to the post-synaptic potential which acts on the i^{th} neuron in state l . The P candidate state vectors ξ^μ are taken to be quenched random variables, assuming the values $1, 2, \dots, Q$, with equal probability. For P candidate state vectors synaptic weights are obtained as

$$J_{ij}^{kl} = \frac{1}{q^2 N} \sum_{\mu=1}^P m_{\xi_i^\mu, k} m_{\xi_j^\mu, l} \quad \forall k, l \in 1, 2, \dots, Q$$

where, $m_{\xi_i^\mu, r}$ is an operator which obeys the Potts symmetry constraint and is given by $m_{\xi_i^\mu, r} = q \delta_{\xi_i^\mu, r} - 1$. ξ_i^μ, r is q state Potts variable representing that i^{th} component of candidate state vector ξ^μ is having state r . This synaptic matrix will have $N^2 q^2$ entries and is symmetric i.e., $J_{ij}^{kl} = J_{ji}^{lk}$.

The potential h_{σ_i} on neuron i (in state σ_i) is

$$h_{\sigma_i} = - \sum_{j=1}^N \sum_{k,l=1}^q J_{ij}^{kl} m_{\sigma_i,k} m_{\sigma_j,l}$$

The induced local fields for each of the q Potts states is calculated. At zero temperature the state of the neuron in the next time step is fixed to be the state which minimizes the induced local field. The stable states of the system are those configurations in which every neuron state σ_i is in a Potts state which gives a minimum value of h_{σ_i} .

Using replica symmetric theory the theoretical results for storage capacity of q -state Potts glass model of neural networks given in [Kanter88] is

q	3	4	5	9
Critical Capacity $\alpha_c(q)$	0.415	0.82	1.37	4.8

The maximum capacity is represented as $\alpha_c \simeq \frac{q(q-1)}{4} 0.138$. The capacity of q -state Potts glass model is higher than the **Hopfield** network. This is partially due to the fact that in the q -state Potts glass model each synapse between a pair of neurons have q^2 terms in the **synaptic** matrix. For each candidate state vector the embedded information between a pair of neurons have q^2 different possibilities. Higher capacity can also be attributed to the presence of $N \log_2 q$ bits for each candidate state vector.

2.11.2 q -state Ising Spin Networks

In q -state Ising Spin Networks the neuron can take any one of the q values in the set $\{-1 = S_1 < S_2 < \dots < S_{q-1} < S_q = +1\}$. The elements have a zero mean. A **state** of the system will be denoted by $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_N)$. Neural **networks** based on q -state Ising Spin neurons have been reported in [Rieger90, Bolle92, Yedidia89, Baram91]. A fully connected q -state Ising spin glass neural network with **self-feedback**, in case of low loading (i.e. for a finite number of candidate state vectors) is studied in [Bolle92].

Capacity Estimates:- The ability of a neural network with multi-state neurons to store candidate state vectors has been systematically studied in [Rieger90]. Some of the capacity estimates for **q-state** Ising spin neurons as reported in [Rieger90] are listed here,

1. Using **pseudo-inverse** learning rule it is possible to store N candidate state vectors in a network with N neurons. However, the basin of attraction for retrieval without error, shrink with increasing q .
2. Using conventional **Hebb's** learning rule it is possible to store **uncorrelated candidate** state vectors. The critical storage capacity at zero temperature is approximately given as $\alpha_c(q) \simeq 0.3q^{-2}$ for $q \gg 1$.
3. For continuously varying activities all linear combinations of the candidate state vectors within the cube $[\sigma_{min}, \sigma_{max}]^N$ are degenerate in energy and the network is not usable for associative memory or pattern recognition. Thus for these **applications** discrete values should be used.
4. With increasing q , spurious states other than linear combinations of the candidate state vectors occur. These spurious states have higher energy and are also **metastable**.

Three-state Ising Spin Network:- A model of neural network with three states (1,0,-1) neurons is proposed in [Yedidia89]. The **synaptic** weights are given by

$$J_{ij} = \frac{C_{ij}}{c} \sum_{\mu=1}^P \xi_i^{\mu} \xi_j^{\mu} \quad i, j = 1, 2, \dots, N$$

where C_{ij} are random independent parameters (independent of C_{ji}) which take the **value** 1 with probability $\frac{c}{N}$ and 0 with probability $1 - \frac{c}{N}$. c represents the mean number of synapses per neuron.

Synchronous or asynchronous dynamics can be used for operating the network. The local field h_i of neuron i is given by $h_i = \sum_{j=1}^N J_{ij} \sigma_j$. The state of the **neurons** are updated according to rule

$$\sigma_i(t+1) = \begin{cases} +1 & \text{with probability } \frac{\exp((h_i - \theta)/T)}{Z} \\ 0 & \text{with probability } Z \\ -1 & \text{with probability } \frac{\exp((-h_i - \theta)/T)}{Z} \end{cases}$$

where $Z = \exp((h_i - \theta)/T) + 1 + \exp((-h_i - \theta)/T)$, θ is the threshold, and T is the temperature.

This model have identical behaviour with **two-state** binary neuron model, in the limit of low activity of candidate state vectors and nearly optimal thresholds. The reason of this similarity is that nearly all the errors of the **three-state** models will consist of neurons having 0 when they should be **+1** or **-1**. These errors are less serious compared to neurons having value **+1** when it should be **-1** or vice versa.

The **three-state** model has some advantages over **two-state** models. **Three-state** models stores about twice as much information because the critical capacity α_c is nearly the same, but the active neuron can now be 1 or **-1**, instead of always 1.

Two-state Representation Of Three-state Network:- The extent to which the dynamical behaviour of neural network consisting of **three-state** neurons can be realised in neural network using **two-state** neurons is investigated in [Stark90]. Some observation of the process of conversion of a **three-state** neural network to a **two-state** representation as reported in [Stark90] are given here.

For every **three-state** network (with deterministic dynamics) it can be written as a **two-state** network with twice the number of neurons. When more noise is added, there is no **two-state** neural network with equivalent probabilistic dynamics for a given **three-state** neural network. However, the **three-state** network does always have a **two-state** representation at **sufficient** low noise levels. The **two-state representation** are replacing

one **three-state** neuron with **two-state** neurons. This is fully in accordance with the observation in [Yedidia89] that the storage capacity of **three-state** network is roughly double that of an **two-state** network. The **three-state** neural network model do not have any new behaviour as a result of **generalizations** from **two-state** to **three-state** levels.

2.11.3 *q-state Hopfield network*

The state of neurons are similar to q-state Ising Spin neurons, but there are no bounds of **+1** or **-1** on highest neuron state and lowest state respectively. The states of neurons are selected **equidistantly** and symmetrically around 0. A neural network model based on such neurons have been proposed in [Prados, Stiefvater92].

Four-state Hopfield Neural Network:- Multi-state neuron with four-states where any neuron takes on states 3, 1, -1, -3 is considered in [Prados]. The **synaptic** weights are obtained as

$$J_{ij} = \sum_{\mu=1}^P \xi_i^{\mu} \xi_j^{\mu} \quad \forall i \neq j \quad \text{and} \quad J_{ii} = 0 \forall i = j$$

If application of this equation does not store all **candidate** state vectors, the synaptic weights can be modified using a learning rule similar to the Delta learning rule. **Suppose** the bit **i** of a candidate state vector ξ_i^{μ} changes when ξ^{μ} is given as initial input vector to the neural network. Since the **ith** row of the weight matrix determines the next **state** of neuron **i**, each weight in the **ith** row can be changed in the direction that will cause ξ_i^{μ} to approach ξ_i^{μ} . The change $\Delta J_{ij} = L(\xi_i^{\mu} - \sigma_i)\xi_j^{\mu}$, where σ_i is the output state of neuron **i** when the input vector ξ^{μ} is given as an input vector. **L** is a learning constant. Next

state of neuron i is determined as

$$\sigma_i = \begin{cases} 3 & \text{if } \sum_{j=1}^N J_{ij} \sigma_j > \theta \\ 1 & \text{if } \theta \geq \sum_{j=1}^N J_{ij} \sigma_j > 0 \\ -1 & \text{if } 0 \geq \sum_{j=1}^N J_{ij} \sigma_j > -\theta \\ -3 & \text{if } -\theta \geq \sum_{j=1}^N J_{ij} \sigma_j \end{cases}$$

Large value of convergence ratio θ/L necessitate more frequent changes in **synaptic** weight. If the ratio is too small, then it may not be possible to learn the candidate state vector. For **q-state** neural networks the convergence ratio should **be** selected to enable storage of any candidate state vector. The following convergence ratio will allow any candidate state vector to be stored

$$\frac{\theta}{L} = V_{max}^2 V_{diff} (N - 1)$$

where, V_{max} is the maximum allowable magnitude for the output of a neuron and V_{diff} is the maximum difference between any two adjacent output values.

Following conclusions are made based on the experiments. Four-state **Hopfield** network can store more candidate state vectors for a given number of neurons than the binary models. A binary model of N neurons used to store N bit candidate state **vectors** can be converted to a four-state model of $N/2$ neurons to store four-state candidate state vectors of length $N/2$. This reduces the number of synaptic weights significantly.

The capacity of the network in terms of candidate state vectors **is** bounded by the number of neurons, but the information capacity increases as the size of the **neuron** alphabet increases. For a neuron that take k different **values**, the total information is the number of candidate state **vectors** (P) **times** the information in each state **vector** ($N \log_2 k$). This can be compared to the capacity of the binary Hopfield model which **is** **maximum** of N candidate state vectors. The information capacity is N^2 **bits**.

2.11.4 Continuous Valued Neurons

The output from a continuous valued (analogue) neuron and hence its state is any value from continuous range of values. The time evolution of the state u_i of analogue neuron is described by

$$\frac{du_i}{dt} = -u_i + \sum_{j=1}^N J_{ij} z_j$$

U_i ($-\infty < u_i < \infty$; $i = 1, 2, \dots, N$) are set of real variables which are output of neuron having graded response $z_i = f(u_i)$. The neural networks based on such **neurons** are known as **Analogue neural network** [Hopfield84, Fukai92, Marcus90]. The **storage** capacity of analogue neural network decreases with decreasing analogue gain. **Decreasing** the analogue gain dramatically suppresses the number of spurious states. Thus it **implies** that for analogue neural networks with an appropriately reduced analogue **gain** would considerably improve the network performance in return for the slight decrease in the storage capacity.

2.12 HIGHER ORDER NEURAL NETWORKS

Several models for large interconnected networks of neurons with emergent collective behaviour have been proposed by use of ideas borrowed from statistical mechanics. The number of candidate state **vectors** that can be made fixed points in such **systems** is of order N . To achieve greater flexibility and programming **capability** several researchers have noticed that **Hamiltonians** of higher order, i.e., **defined** by an **algebraic** form of degree d , could easily be introduced. Moreover, such forms arise naturally in **optimization** problems.

The capacity estimate of second order **Hopfield** network is in given **Eq.2.10**. This equation is applicable to higher order **Hopfield** networks by replacing K_2 with K_d and

replacing C_2 with C_d . The values of K_d and C_d for different values of d as given in [Baldi87] are given below.

d	K_d	C_d	d	K_d	C_d
1	1	0	5	1.3031	0.5721
2	1.0505	0.2874	10	1.7032	0.7215
3	1.1320	0.4265	100	6.6705	0.9461
4	1.2178	0.5124	1000	39.3100	0.9916

2.13 CONCLUSION

This chapter introduces **Hopfield** network with various **notations**, definitions, learning rules, and update rules. This chapter also reports a survey of research related to various modifications to the Hopfield network, the effect of these changes on the performance of the network, various definitions of capacity, and capacity estimates. These modifications to Hopfield network till now have been mostly attempted in isolation. Use of more than one modification may help in precise design of Hopfield class networks. This chapter attempts to provide a unified basis for such a study. Neurones with self-feedback character is one of the modifications to Hopfield network which has been selected for detailed study (Chapter 3).

Capacity of a Hopfield class network is one of the criterion based on which the performance of the network can be assessed. In this chapter it **has** been observed that the **basic** concept of the capacity is understood in very diversified (and **sometimes** contradictory) manner. Further, if the capacity of a network is the **measure** of performance of a network then it should also be one of the guiding force behind the development of learning **rules**. In the absence of a clear and unique concept **of** capacity it is difficult to **use this concept** for design of learning rules.

Consider a learning rule which makes all the input vectors to have a **unique** stable state. Such a learning **rule**, if developed will have the highest possible **capacity** as **per** Definition 2.1 . But such an algorithm will not have any practical value as the output can give completely accurate indication of what the input was. In Chapter 4, a learning rule has been proposed based on Definition 2.4 . This rule will enable all the state vectors having **upto** specific number of bits as 1, stable in the network.

All the definitions of capacity consider the number of candidate state vectors that can be made stable, but do not consider the presence or absence of spurious **states**. The presence of spurious state degrades the performance of a network. Hence measure of performance of network should not exclude this aspect. It has also been observed that **most** of the research related with **Hopfield** class network has considered the random selection of candidate state vectors. Such results are likely to fail to make some specified set of candidate state vectors stable in the network. This has motivated us to design learning rule which eliminates presence of spurious states and make any set of two **candidate** state vectors stable in the network (Section 4.6).

Chapter 3

CHANGES IN DIAGONAL ELEMENTS

3.1 INTRODUCTION

In the **Hopfield** network [Hopfield82] a neuron cannot give direct self-feedback. The **ab-**sence of direct self-feedback is based on the concept of stability, capacity and local minima of the energy function [Hopfield82, Hopfield84, Amit89]. Although majority of researchers consider Hopfield network having no direct **self-feedback**, there is no biological evidence supporting this hypothesis. In fact, in certain biological studies it is observed that a neuron takes a feedback from itself directly [Carpenter90]. Chapter 2 reports results of some research **articles** [Houselander90, Prados89, Braham88, Sezan90, Gindi88, Yanai90] which consider Hopfield network with direct self-feedback. Moreover, some related concept of changes in threshold elements [Der92] has also been reported in Chapter 2.

In this chapter some experimental observations and theoretical conclusions on the study of direct self-feedback in Hopfield network is reported. Section 3.2 deals with the motivation for the study of changes in diagonal elements. Section 3.3, using an example introduces the concept of diagonal element **changes** and **its** effect on dynamics of Hopfield network. In Section 3.4 critical values for diagonal elements which provide the condition for changes of state of bipolar neurons are **proposed**. Theoretical **basis** of increase in value of diagonal elements leading to the neuron attaining **no-change-state** is given **Section** 3.5 . Section 3.6 deals with the stability of a **neuron** given fixed state of one or more other neurons. A geometrical interpretation of **neuron** state **changes** is given in **Section** 3.7 .

In Section 3.8 the critical values for changes in threshold are obtained and the effect of such changes is **analysed**. Section 3.9 illustrates the difference between the threshold changes and diagonal changes. In this section it is shown that these types of changes are not truly complementary. Section 3.10 deals with the study of energy function for changes in the diagonal elements. Section 3.3 to Section 3.7 and Section 3.10 reports the study of diagonal element changes in **Hopfield** network with bipolar neurons. Section 3.11 deals with changes in diagonal elements of Hopfield network with binary neurons. The conclusions of study of diagonal element changes are given in Section **3.12** .

3.2 MOTIVATION FOR STUDY OF DIAGONAL CHANGES

Any learning rule to construct **synaptic** matrix makes use of the specified set of candidate state vectors. But the matrix so constructed need not have all these candidate state vectors as its stable states. **Moreover**, the introduction of new stable state vectors or the deletion of an existing stable state is done by making necessary changes in the synaptic matrix. This usually changes the stable status of other state vectors. Further, any study of dynamics of neural networks not only concentrates on the stable states but also concerns with equally important issues like basins of attraction, minimization of energy function etc. Hence, the learning rule which aims only at having a set of stable **states** may not provide proper (adequate) insight to the study of dynamics of neural networks. It may be required to have a separate study.

In [Gindi88] the Hopfield network with **non-zero** diagonal elements (with $J_{ii} = N$) is considered. It is shown that by allowing non-zero diagonal terms in the synaptic **ma-**trix the stable states of the network need not **change**. On the other hand, the **non-zero** diagonal network is shown to outperform the original Hopfield network [Gindi88]. The non-zero diagonal affects dynamics and can be effectively used to improve the recalling

ability of the Hopfield network [Yanai90]. It is proposed that the changes in the diagonal elements may be useful in obtaining a particular neural dynamics, stability of new candidate state vectors of a particular kind with little affect on the existing stable vectors, and removing the stable status of a particular kind of stable state vectors.

3.3 AN EXAMPLE

In this section the concept of diagonal element changes in Hopfield network is being introduced with the help of an example. Consider two Hopfield networks $(\mathbf{A}, \theta)$ and $(\mathbf{B}, \theta)$. The synaptic matrices $\mathbf{A}$ and $\mathbf{B}$, and threshold vector θ are given below.

$$\mathbf{A} = \begin{pmatrix} 110 & 89 & 54 & -76 & -76 \\ 89 & 116 & 12 & -25 & 19 \\ 54 & 12 & 110 & -45 & -17 \\ -76 & -25 & -45 & \mathbf{64} & \mathbf{-15} \\ -76 & 19 & -17 & -15 & 0 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 110 & 89 & 54 & -76 & -76 \\ 89 & 116 & 12 & -25 & 19 \\ 54 & 12 & 0 & -45 & -17 \\ -76 & -25 & -45 & \mathbf{64} & \mathbf{-15} \\ -76 & 19 & -17 & -15 & 110 \end{pmatrix}$$

$$\theta = \begin{pmatrix} 40 \\ 30 \\ 20 \\ 93 \\ 56 \end{pmatrix}$$

The matrix A differs from matrix B in only two diagonal terms J_{33} and J_{55} . For matrix A , $J_{33} = 110$ and $J_{55} = 0$ while for matrix B , $J_{33} = 0$ and $J_{55} = 110$.

The neurons are considered to be bipolar neurons and asynchronous mode is used for updating the Hopfield network. Changes in neuron states are observed by considering the neuron state in initial vector (IV) and neuron state in the corresponding stable state vector (SV). Four possibilities for a neuron state changes are

If IV_i is 1 and corresponding SV_i is 1 then value associated is 1.

If IV_i is 1 and corresponding SV_i is -1 then value associated is 2.

If IV_i is -1 and corresponding SV_i is 1 then value associated is 3.

If IV_i is -1 and corresponding SV_i is -1 then value associated is 4

By observing each input vector and the corresponding stable state, a value 1,2,3 or 4 is associated with each neuron of the Hopfield network. For the Hopfield network (A, θ) and (B, θ) the observation of changes in input vector and corresponding stable vector are listed in Table 3.1. Summary of observations of neuron state changes in Hopfield network (A, θ) and (B, θ) is given in Table 3.2.

It can be observed from Table 3.1 and 3.2 that by changing the diagonal element J_{33} from 110 in (A, θ) to 0 in (B, θ) the number of changes in the neuron state of neuron 3 has increased from 2 out of 31 to 13 out of 31. It can also be observed that by changing the diagonal element J_{55} from 0 in (A, θ) to 110 in (B, θ) the number of changes in the neuron state of neuron 5 has decreased from 13 out of 31 to 6 out of 31.

Hence it is observed that by increasing the diagonal element value the corresponding neuron state is subjected to less changes. By decreasing the diagonal element value the corresponding neuron state is subjected to more changes.

TABLE 3.1: OBSERVATION OF CHANGES IN NEURON STATES IN INPUT VECTOR AND CORRESPONDING STABLE STATE VECTOR										
INPUT VECTOR	HOPFIELD NETWORK									
	(A, θ)					(B, θ)				
	$i=1$	2	3	4	5	1	2	3	4	5
-1 -1 -1 -1 +1	4	4	4	4	1	4	4	4	4	1
-1 -1 -1 +1 -1	4	4	4	1	3	4	4	4	1	4
-1 -1 -1 +1 +1	4	4	4	1	1	4	4	4	1	4
-1 -1 +1 -1 -1	4	4	1	4	4	4	4	2	3	4
-1 -1 +1 -1 +1	4	4	1	4	2	4	4	2	4	1
-1 -1 +1 +1 -1	4	4	2	1	3	4	4	2	1	4
-1 -1 +1 +1 +1	4	4	2	1	1	4	4	2	1	1
-1 +1 -1 -1 -1	4	1	4	4	3	3	1	3	4	4
-1 +1 -1 -1 +1	4	1	4	4	1	4	1	4	4	1
-1 +1 -1 +1 -1	4	1	4	1	3	4	2	4	1	4
-1 +1 -1 +1 +1	4	1	4	1	1	4	1	4	1	1
-1 +1 +1 -1 -1	3	1	1	4	4	3	1	1	4	4
-1 +1 +1 -1 +1	4	1	1	4	1	4	1	2	4	1
-1 +1 +1 +1 -1	3	1	1	2	4	4	2	2	1	4
-1 +1 +1 +1 +1	4	1	1	2	1	4	1	2	1	1
+1 -1 -1 -1 -1	1	4	4	4	4	1	4	3	4	4
+1 -1 -1 -1 +1	1	4	4	4	2	2	4	4	4	1
+1 -1 -1 +1 -1	2	4	4	1	3	2	4	4	1	4
+1 -1 -1 +1 +1	2	4	4	1	1	2	4	4	1	1
+1 -1 +1 -1 -1	1	4	1	4	4	1	4	1	4	4
+1 -1 +1 -1 +1	1	4	1	4	2	1	4	1	4	2
+1 -1 +1 +1 -1	1	4	1	2	4	1	4	1	2	4
+1 -1 +1 +1 +1	1	4	1	2	2	1	4	1	2	2
+1 +1 -1 -1 -1	1	1	4	4	4	1	1	3	4	4
+1 +1 -1 -1 +1	1	1	4	4	2	1	1	3	4	2
+1 +1 -1 +1 -1	1	1	4	2	4	1	1	3	2	4
+1 +1 -1 +1 +1	1	1	4	2	2	1	1	3	2	2
+1 +1 +1 -1 -1	1	1	1	4	4	1	1	1	4	4
+1 +1 +1 -1 +1	1	1	1	4	2	1	1	1	4	2
+1 +1 +1 +1 -1	1	1	1	2	4	1	1	1	2	4
+1 +1 +1 +1 +1	1	1	1	2	2	1	1	1	2	2

TABLE 3.2: SUMMARY OF OBSERVATIONS							
Neuron	Hopfield Network	Number of Observations				No. of obvs. with No Change	No. of obvs. with Change
		1	2	3	4		
1	(A, θ)	14	2	2	13	27	4
	(B, θ)	13	3	2	13	26	5
2	(A, θ)	16	0	0	15	31	0
	(B, θ)	14	2	0	15	29	2
3	(A, θ)	14	2	0	15	29	2
	(B, θ)	9	7	6	9	18	13
4	$(A, 9)$	8	8	0	15	23	8
	(B, θ)	10	6	1	14	24	7
5	$(A, 0)$	8	8	5	10	18	13
	(B, θ)	10	6	0	15	25	6

3.4 CRITICAL VALUE OF J_{ii} FOR BIPOLAR NEURONS

It can be observed from the example given in Section 3.3 that the higher the value of any diagonal element the lesser is the tendency of the corresponding neuron to change state. Thus, a critical value can be determined for a diagonal element J_{ii} so that beyond this value, the state of the corresponding bipolar neuron does not change. This critical value can be termed as *no-change-critical-value* and denoted as J_{ii}^{nc} for neuron i . This critical value depends on threshold associated with the neuron and the synaptic weights with other neurons.

Similarly it can also be observed that decrease in the value of a diagonal element increases the likelihood of change in the state of corresponding neuron. Hence, a critical

value can be reached so that, below this critical value the state of the neuron definitely changes. This critical value can be termed as *sure-change-critical-value* and is denoted for neuron i as J_{ii}^{sc} . These critical values J_{ii}^{nc} and J_{ii}^{sc} are estimated using **sufficient conditions**. But however, these are not necessary **conditions**.

The following Theorem 3.1 gives the estimate of J_{ii}^{nc} and Theorem 3.2 gives estimates of J_{ii}^{sc} .

THEOREM 3.1 In a **Hopfield** network with bipolar neurons, if the **synaptic** matrix satisfies $J_{ii} > \sum_{j=1, j \neq i}^N |J_{ij}| + |\theta_i|$, then neuron i does not change its state. The critical value J_{ii}^{nc} is given by

$$J_{ii}^{nc} = \sum_{j=1, j \neq i}^N |J_{ij}| + |\theta_i|$$

Proof:- In order that neuron i does not change its state from $\sigma_i = +1$, it is necessary that

$$J_{ii} + \sum_{j=1, j \neq i}^N J_{ij} \sigma_j - \theta_i \geq 0$$

For the above expression to be satisfied it is necessary that

$$J_{ii} \geq -\left(\sum_{j=1, j \neq i}^N J_{ij} \sigma_j - \theta_i \right) \quad (3.1)$$

Similarly, in order that neuron i **does** not change its state from $\sigma_i = -1$ it is necessary that

$$-J_{ii} + \sum_{j=1, j \neq i}^N J_{ij} \sigma_j - \theta_i < 0$$

For the above expression to be satisfied it is necessary that

$$J_{ii} > \sum_{j=1, j \neq i}^N J_{ij} \sigma_j - \theta_i \quad (3.2)$$

It can be observed that

$$-|J_{ij}| < J_{ij} \sigma_j < |J_{ij}|, \quad \text{for any } \sigma_j$$

and

$$-I_{ft} I < -ft < I_{ft} I$$

Hence

$$-\sum_{j=1, j \neq i}^N |J_{ij}| - |\theta_i| \leq \sum_{j=1, j \neq i}^N J_{ij} \sigma_j - \theta_i \leq \sum_{j=1, j \neq i}^N |J_{ij}| + |\theta_i| \quad (3.3)$$

If

$$J_{ii} > \sum_{j=1, j \neq i}^N |J_{ij}| + |\theta_i|$$

then

$$J_{ii} \geq \sum_{j=1, j \neq i}^N J_{ij} \sigma_j - \theta_i$$

This satisfies Condition 3.2.

If

$$J_{ii} > \sum_{j=1, j \neq i}^N |J_{ij}| + |\theta_i|$$

then

$$-J_{ii} < -\sum_{j=1, j \neq i}^N |J_{ij}| - |\theta_i|$$

i.e.,

$$-J_{ii} < \sum_{j=1, j \neq i}^N J_{ij} \sigma_j - \theta_i$$

Hence

$$J_{ii} > -\left(\sum_{j=1, j \neq i}^N J_{ij} \sigma_j - \theta_i \right)$$

. This satisfies Condition 3.1.

Hence the theorem is proved.

□

THEOREM 3.2 In a Hopfield network with bipolar neurons, if the **synaptic** matrix satisfies the condition, $J_{ii} < -(\sum_{j=1, j \neq i}^N |J_{ij}| + I_{ft} I)$ then neuron i does change its state when updated.

The sure-change-critical-value is given by

$$J_{ii}^{sc} = -\left(\sum_{j=1, j \neq i}^N |J_{ij}| + |\theta_i| \right)$$

Proof:- In order that neuron i changes its state from $\sigma_i = +1$, it is necessary that

$$J_{ii} + \sum_{j=1, j \neq i}^N J_{ij}\sigma_j - \theta_i < 0$$

For the above expression to be satisfied it is necessary that

$$J_{ii} < -\left(\sum_{j=1, j \neq i}^N J_{ij}\sigma_j - \theta_i \right) \quad \text{ft)} \quad (3.4)$$

Similarly, in order that neuron i changes its state from $\sigma_i = -1$ it is necessary that

$$-J_{ii} + \sum_{j=1, j \neq i}^N J_{ij}\sigma_j - \theta_i \geq 0$$

For the above expression to be satisfied it is necessary that

$$J_{ii} \leq \sum_{j=1, j \neq i}^N J_{ij}\sigma_j - \theta_i \quad (3.5)$$

Using similar argument as discussed in Theorem 3.1, it can be observed that,

$$-1 \leq J_{ij} \leq 1, \quad \text{for any } j$$

and

$$-1 \leq \theta_i \leq 1$$

Hence

$$-\sum_{j=1, j \neq i}^N |J_{ij}| - |\theta_i| \leq \sum_{j=1, j \neq i}^N J_{ij}\sigma_j - \theta_i \leq \sum_{j=1, j \neq i}^N |J_{ij}| + |\theta_i| \quad (3.6)$$

If

$$J_{ii} < -\left(\sum_{j=1, j \neq i}^N |J_{ij}| + |\theta_i| \right)$$

then

$$J_{ii} < \sum_{j=1, j \neq i}^N J_{ij} \sigma_j - \theta_i$$

This satisfies Condition 3.5.

If

$$J_{ii} < -\left(\sum_{j=1, j \neq i}^N |J_{ij}| + |\theta_i| \right)$$

i.e.

$$-J_{ii} > \sum_{j=1, j \neq i}^N |J_{ij}| + |\theta_i|$$

then from inequality 3.6

$$-J_{ii} > \sum_{j=1, j \neq i}^N J_{ij} \sigma_j - \theta_i$$

i.e.

$$J_{ii} < -\left(\sum_{j=1, j \neq i}^N J_{ij} \sigma_j - \theta_i \right)$$

This satisfies Condition 3.4.

Hence the Theorem is proved.

□

Based on the discussion in Theorem 3.1, it can be observed that each diagonal element reaches its critical value independent of other diagonal elements when the off-diagonal elements are kept unchanged. With all the diagonal elements greater than the **respective no-change-critical-values**, it can be ensured that all the state vectors are stable in the network.

Similarly based on the **discussion** in Theorem 3.2, it is observed that by having all the diagonal elements less than the respective **sure-change-critical-value**, it can be **ensured** that no state vector is stable in the network.

3.5 IMPACT OF DIAGONAL CHANGES ON NETWORK DYNAMICS

In this section an attempt is made to study the changes in dynamics of **Hopfield** network with changes in diagonal elements. The next state of a neuron being updated depends on present state of the neuron, corresponding diagonal element, the **synaptic** weights with other neurons, and corresponding threshold element. These factors determine the change or **no-change** status of a neuron in a particular time unit. A neuron of a network can belong to one of the following categories.

1. **Sure-change state:-** The neuron changes its state whenever it is updated.
2. **Flexible state:-** The neuron is in a change or **no-change** state in different time units depending on the factors listed above.
3. **No-change state:-** The state of a neuron is not subjected to change when updated.

The **diagonal** element changes keeping other elements of synaptic matrix and threshold vector constant can be used to place a neuron in any one of these categories.

Consider a Hopfield network $(\mathbf{D}, \theta)$ with all diagonal element values less than the corresponding J_{ii}^{sc} . This network does not have any stable state. Any input vector to the network $(\mathbf{D}, 0)$ will oscillate between the vector and its complement. The network can be considered to have **bidirectionally** stable states. **Bidirectionally** stable states can be considered equivalent to the concept of stable states if such a situation occurs for some vectors in a network. But, a network like $(\mathbf{D}, 0)$ with all bidirectionally stable states, does not seem to have much practical value. The network with even one neuron in **sure-change** state will not have any direct stable state.

Let a Hopfield network $(\mathbf{E}, 9)$ with all neurons in flexible state is **obtained** by **increasing** the values of diagonal elements of the network $(\mathbf{D}, 6)$. The network $(\mathbf{E}, 9)$ has some stable states. By increasing the values of diagonal elements the network will have **some** more

stable states. Continuing this process further the **Hopfield** network $(F, 0)$ can be attained. In network $(F, 0)$ all diagonal elements are greater than corresponding J_{ii}^{nc} and hence all possible input state vectors are stable. Any further increase in the values of diagonal elements will have no impact on the performance of the network. The class of Hopfield network between $(E, 0)$ and $(F, 0)$ are of special interest. This phenomenon of diagonal changes and changes in the stable states of the network can be used as a learning process. If at a stage of dynamics it is required that a neuron belongs to a particular category then the corresponding diagonal values can be accordingly changed.

For a **2-neuron** Hopfield network this phenomenon is illustrated in Figure 3.1. At the stage I only one $(-1, -1)$ of the four possible input vectors is stable. Neuron 1 is selected and the value of the corresponding diagonal element is increased. When the value of diagonal element corresponding to neuron 1 becomes more than the **no-change-critical-value**, $(1, -1)$ also becomes a stable state of the network. This is due to the fact that the neuron 1 is in a **no-change** state. Whatever state that is associated with neuron 1 from the input vector, the neuron remains in the same state which is also reflected in the stable state. This is depicted as stage II of the network.

Similarly starting from stage I and increasing the value of the diagonal corresponding to neuron 2 it can be seen that the state $(-1, 1)$ also becomes a stable state of the network. This is represented as stage III in Figure 3.1. By having the diagonal elements corresponding to both the neurons, having values more than its **no-change-critical-values** all the input states become the stable states of the network. This situation is represented as stage IV in Figure 3.1.

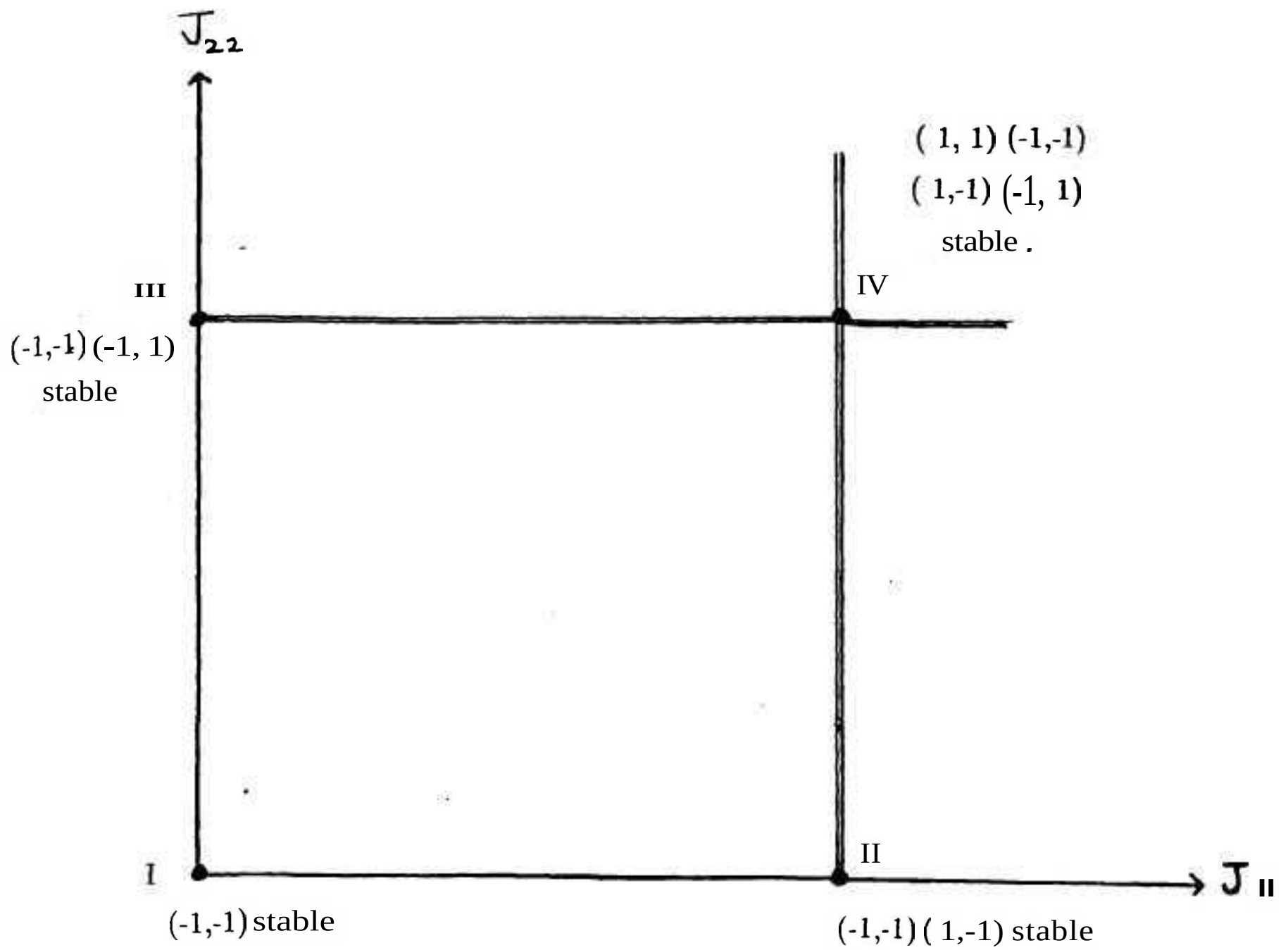


Figure 3.1: Various stages of dynamics of two-neuron Hopfield network with changes in diagonal elements

3.8 CONDITIONAL STABILITY

Increasing the value of a specific set of diagonal elements in a network with large number of neurons it is possible to ensure that the state of these neurons is not changed. However, the more desirable feature of such study is to make specific set of state **vectors** stable and not to make specific set of neurons stable independent of other neurons. For example in a **3-neuron Hopfield** network, the first diagonal element can be increased so that first neuron does not change its state irrespective of the states of other neurons. **Whereas**, if the intention is to make a specific state vector (+1 -1 -1) to be stable and not (+1 +1 +1) then it is necessary to study a kind of **conditional stability** of individual neurons and not the independent stability as discussed in earlier sections of this chapter. Conditional stability of a neuron i can be defined as the stability of a neuron i given a fixed state of one or more other neurons. However, such a study is much more complex than the one discussed for the independent stability. Another simple case namely **pairwise conditional stability** of neurons is considered here. Critical value of the i^{th} diagonal element J_{ii} for a given state of neuron j is denoted as $(J_{ii} | j)$. The critical value of $(J_{ii} | j)$ is characterised in the following theorem.

THEOREM 3.3 The critical value of $(J_{ii}^{nc} | j)$ such that neuron i will not change its state for a given state σ_j of neuron j but independent of other neurons is given by

$$J_{ii} > \sum_{k=1, k \neq i, j}^N |J_{ik}| + |\theta_i| - \sigma_j J_{ij}$$

The critical value $(J_{ii}^{nc} | j)$ is given by

$$(J_{ii}^{nc} | j) = \sum_{k=1, k \neq i, j}^N |J_{ik}| + |\theta_i| - \sigma_j J_{ij}$$

Proof:- In order that neuron i does not change its state from $\sigma_i = +1$, it is necessary that

$$J_{ii} + \sum_{k=1, k \neq i}^N J_{ik} \sigma_k - \theta_i \geq 0$$

For the above expression to be true it is necessary that

$$J_{ii} \geq -\left(\sum_{k=1, k \neq i}^N J_{ik} \sigma_k - \theta_i \right)$$

This implies

$$J_{ii} > -\left(\sum_{k=1, k \neq i, j}^N J_{ik} \sigma_k - \theta_i \right) - J_{ij} \sigma_j \quad (3.7)$$

Similarly, in order that neuron i does not change its state from $\sigma_i = -1$ it is necessary that

$$-J_{ii} + \sum_{k=1, k \neq i}^N J_{ik} \sigma_k - \theta_i < 0$$

For the above expression to be satisfied it is necessary that

$$J_{ii} > \sum_{k \neq i, j} J_{ik} \sigma_k - \theta_i + J_{ij} \sigma_j \quad (3.8)$$

It can be observed that

$$-|J_{ik}| < J_{ik} \sigma_k < |J_{ik}|, \quad \text{for any } \sigma_k$$

and

$$-|\theta_i| \leq -\theta_i \leq |\theta_i|$$

Hence

$$-\sum_{k=1, k \neq i, j}^N |J_{ik}| - |\theta_i| \leq \sum_{k=1, k \neq i, j}^N J_{ik} \sigma_k - \theta_i < \sum_{k=1, k \neq i, j}^N |J_{ik}| + |\theta_i| \quad (3.9)$$

If

$$J_{ii} > \sum_{k=1, k \neq i, j}^N |J_{ik}| + |\theta_i| - \sigma_j J_{ij}$$

then

$$-J_{ii} < -\sum_{k \neq i, j} |J_{ik}| - |\theta_i| + \sigma_j J_{ij}$$

i.e.,

$$-J_{ii} < \sum_{k=1, k \neq i, j}^N J_{ik} \sigma_k - \theta_i + J_{ij} \sigma_j$$

This satisfies condition 3.7 .

Similarly it can be shown that for $J_{ii} > \sum_{k=1, k \neq i, j}^N |J_{ik}| + |\theta_i| - \sigma_j J_{ij}$, condition 3.8 is satisfied.

Hence the theorem is proved. □

In the same way as Theorem 3.3 a conditional, lower critical value $(J_{ij}^* \setminus j)$ for neuron i can be determined so that i^{th} neuron will surely change its state for all possible states of other neurons except neuron j , which is given to be fixed at σ_j .

It is observed that pairwise conditional **no-change-critical-value** is smaller than the independent **no-change-critical-value**. It is also observed that pairwise conditional **sure-change-critical-value** is greater than the independent **sure-change-critical-value**. This is shown below.

$$- |J_{ij}| \leq J_{ij} \sigma_j \text{ Or } |J_{ij}| \geq -J_{ij} \sigma_j$$

Hence

$$(J_{ii}^{nc} \setminus j) < J_{ii}^{nc}$$

Similarly

$$(J_{ii}^{sc} \setminus j) > J_{ii}^{sc}$$

3.7 GEOMETRICAL INTERPRETATION

In this section the **no-change-state** of a neuron is interpreted geometrically. It is **observed** in Theorem 3.1 that, when $J_{ii} > J_{ii}^{nc}$ then due to **no-change-condition** we have,

$$J_{ii} + \sum_{j=1, j \neq i}^N J_{ij} \sigma_j - \theta_i \geq 0$$

irrespective of the value of $\sigma_j, j = 1, 2, \dots, N$ and $j \neq i$. The inequality can be rewritten as

$$\sum_{j=1, j \neq i}^N J_{ij} \sigma_j \geq k \quad \text{where } k = -J_{ii} + 0;$$

This inequality is satisfied by all values of σ_j when $J_{ii} > J_{ii}^{nc}$. The set of all possible values of σ_j defines a $N - 1$ dimensional **hypercube** having vertices defined by $\sigma_j = \pm 1, j \neq i, j = 1, 2, \dots, N$. In this space $\sum_{j=1, j \neq i}^N J_{ij} \sigma_j = k$ defines a **hyperplane** and hence the inequality $\sum_{j=1, j \neq i}^N J_{ij} \sigma_j > k$ defines a half-space in which the complete hypercube is contained. As the value of k increases, the **hyperplane** approaches the hypercube and hence some vertices of the hypercube tend to violate the constraint $\sum_{j=1, j \neq i}^N J_{ij} \sigma_j \geq k$. With the increase of k more and more vertices of the hypercube cross over to the other half-space. It is evident that increase in k is also accomplished by decrease in J_{ii} . Thus in other words decrease in the value of J_{ii} below J_{ii}^{nc} makes more and more neurons to deviate from **no-change-state**. Hence, more and more changes are observed in this process. This justifies the observation in the example given in section 3.3. The concept is illustrated in Figure 5.2.

3.8 THRESHOLD CHANGES AND CRITICAL VALUES

This section deals with the study of changes in threshold elements and its impact on the performance of **Hopfield** network. The observations and the conclusions in this section are restricted to a Hopfield network with all **neurons** belonging to the flexible category. Like critical values of J_{ii} 's, there also exist critical values of θ_i 's. The **critical values** that can be associated with threshold vectors are defined and an estimate is derived for these critical values. Consider a Hopfield network with all **neurons** belonging to flexible

The vertices in *halfspace* above the *hyperplane* k 's represent neurons in the no-change state

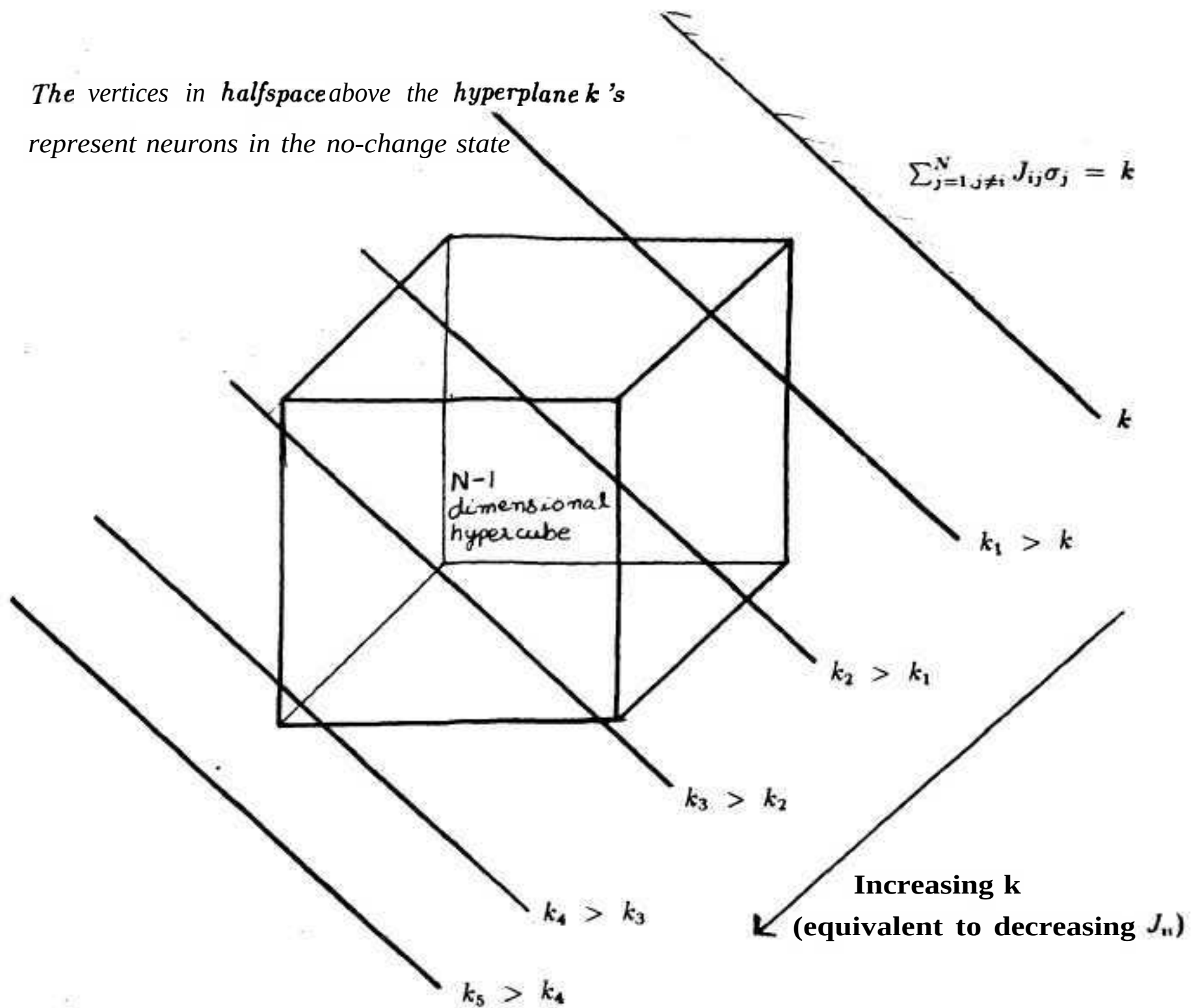


Figure 3.2: Geometrical interpretation of Hopfield dynamics for $N = 4$ neurons.

category. If the threshold θ_i of neuron i is increased to a very high value then,

$$J_{ii} + \sum_{j=1, j \neq i}^N J_{ij} \sigma_j < 0^*$$

Hence, the neuron i changes its state only if $\sigma_i = +1$ and it will not change its state when $\sigma_i = -1$.

Similarly, if θ_i takes on very low value, then

$$J_{ii} + \sum_{j=1, j \neq i}^N J_{ij} \sigma_j \geq \theta_i$$

Hence the neuron i changes its state only if $\sigma_i = -1$ and remains in a **no-change-state** for $\sigma_i = +1$.

Thus, the change in the threshold values (keeping the elements of the **synaptic** matrix constant) ensures changes in the state of neuron only for one of the possible two states. That is, with large values of threshold the corresponding neuron will attain a **+1** state irrespective of the state that is associated with it from the input vector. Similarly with a very low value of threshold the corresponding neuron will attain a **-1** state irrespective of the state associated with the neuron from the input vector. The **definitions** of critical values and its estimates are given below.

Plus **one (+1)** critical value of a threshold of a bipolar neuron (θ_i^P) :- The value of threshold of a neuron i such that all values of threshold below this value will definitely guarantee that the neuron attains a **+1** state. θ_i^P is estimated using a **sufficient condition**, but however, it is not a necessary condition that θ_i should be less than θ_i^P to ensure that the neuron i definitely attains a **+1** state. This critical value is given by

$$\theta_i^P = - \sum_{j=1}^N |J_{ij}|$$

Minus one (-1) critical value of threshold of a bipolar neuron (θ_i^M) :- The value of threshold of a neuron i such that all values of threshold above this value will **definitely**

guarantee that the neuron attains a -1 state. θ_i^M is estimated using a sufficient condition, but however, it is not a necessary condition that θ_i should be greater than θ_i^M to ensure that the neuron i definitely attains a -1 state. This critical value is given by

$$\theta_i^M = \sum_{j=1}^N |J_{ij}|$$

By having threshold values of all neurons greater than θ_i^P it is possible to make the vector with all elements as +1, as the only stable state of the network. Similarly by having threshold values of all neurons less than θ_i^M it is possible to make the vector with all elements as -1, as the only stable state of the network.

3.9 DIAGONAL CHANGES vs THRESHOLD CHANGES

In literature the diagonal element value and threshold value of a neuron are considered to be complementary. In [Hecht-Nielsen91] it has been mentioned that instead of having an explicit threshold, a zero threshold value can be used by changing the diagonal elements of synaptic matrix. It is proposed here that the diagonal element values and the threshold values are not truly complementary.

As mentioned in earlier sections, the changes in the diagonal elements can lead to a situation where all input vectors are stable or an unstable situation where the network has no stable state. These situations are attained just by diagonal element changes keeping all other aspects of the Hopfield network constant. This is different from the effect on dynamics of Hopfield network due to the changes in the threshold elements keeping all other aspects of the network constant. The changes in threshold elements can affect only one of the values associated with the neuron. A threshold value less than θ_i^P ensures that the neuron i ultimately attains a +1 state irrespective of initial state. Similarly threshold values greater than θ_i^M ensures that the neuron i ultimately attains a -1 state irrespective of initial state. With changes in threshold element it is possible to ensure

Chapter 3. CHANGES IN DIAGONAL ELEMENTS

that one type of change of neuron state i.e., +1 to -1 or -1 to +1 does not occur ~~whereas~~ the changes of other type occurs with high probability. Thus it can be concluded that changes in diagonal element are different from changes in the threshold **elements** for a neuron. These two changes are not truly complementary.

Adjusting diagonal element values and threshold values in a **Hopfield** network may help in obtaining the required dynamics. These changes can play an important role in obtaining the required computational performance of the Hopfield network.

3.10 DIAGONAL CHANGES AND ENERGY FUNCTION

The Hopfield network, in fact, performs a local search where the neighbourhood of the search is the immediate neighbourhood of a vertex on the N dimensional **hypercube**. The asynchronous mode of computation at any node i can be viewed as the comparison of the energy E at the current vertex and at the adjacent vertex in the i^{th} direction. **This** comparison is not **affected** if we add a constant term to E .

Let,

$$E_1 = E + \frac{1}{2}J_{ii}$$

Then, obviously the changes in energy function E is same as the **changes** in the energy function E_1 for a Hopfield network i.e., $\Delta E = \Delta E_1$.

3.10.1 Change in Diagonal Elements of Synaptic Matrix

Thus it is observed that if the energy function is changed from E to E_1 , the difference in energy at two consecutive network states is not affected. For the energy function E_1 , even if some real number is added to the diagonal elements, the energy **value** doe* not change at any state. Thus it can be concluded that, in **asynchronous** mode of operation, the energy function E of any Hopfield network $(\mathbf{J}, \boldsymbol{\theta})$ converges to a constant **value**, if **and**

only if, the energy function converges for any other Hopfield network $N(r) = (\mathbf{J}(r), \theta)$. In other words, by adding some values to the diagonal elements of the **synaptic matrix**, the set of locally minimizing states of the energy function is not changed. For a synaptic matrix J if the set of locally minimizing states is denoted by L . The following phenomena can be observed.

1. If J has zero diagonal elements then the set of local minima L is also the set of stable states.
2. If J has strictly positive diagonal elements then each element of L is a stable state and may be some elements which are not in L are also stable states. Moreover, the stability is ensured for asynchronous mode of operation of the Hopfield network [Bruck88].
3. If there is no restriction on the diagonal elements of J and if $\mathbf{J}'$ is obtained by adding some elements to the diagonal **elements** so that these become zero then the set of locally minimizing states L is same for J and $\mathbf{J}'$. But the stable states of $\mathbf{J}'$ are also the stable states of J and there are some additional stable **states** in J which do not correspond to locally minimizing points. The state transition **paths** are different for both the matrices as the updating rule is affected by change in diagonal elements. However, any Hopfield network, having synaptic matrix with non-zero diagonal elements, can be transformed to a Hopfield network having zero diagonal elements so that the stable states of the transformed Hopfield network corresponds to the locally **minimizing states** of **original** Hopfield network.
4. As the updating is affected by the change in the diagonal elements, the state transition for J and $\mathbf{J}'$ are different. Hence for two matrices J and $\mathbf{J}'$ differing **only** in diagonal entries, the **associating functions** of input state vector to output stable

vector are different, even though the stable states are common to both matrices. Hence, getting new stable states for any arbitrary matrix (even with unrestricted diagonal elements) is possible by adding large positive number to the diagonal elements. But however, such a scheme is useful only when the study is restricted to the set of stable states and not for associating input state to stable states.

Discussion in this section about the changes in diagonal elements of the **Hopfield** network is applicable when the collection of stable states of the network are considered. These observations cannot be directly extended to other areas like associative memory. The collection of stable states is of interest in the context of capacity of **Hopfield** network.

3.11 DIAGONAL CHANGES IN NETWORK WITH BINARY NEURONS

This section deals with aspect of diagonal element changes for **Hopfield** network with binary neurons are reported in this section. For the neuron state +1 the analysis is same as given in the previous sections. But for the neuron state 0, its product with the corresponding diagonal element becomes 0. In this case the local field of the neuron does not receive contribution from the diagonal element. Thus the diagonal tuning **mechanism** used for bipolar neurons have a limited role in case of binary neurons. This requires a different study of diagonal element changes for binary neurons.

Experiments have been conducted by starting from a high negative value for **diagonal** elements in the **synaptic** matrix for binary and bipolar neurons. The diagonal elements were gradually increased and its effect on the performance of network with bipolar and binary neurons has been observed. Asynchronous mode of **operation** with maximum absolute value of local field as the basis of selection has been used in these **experiments**. It has been observed that a Hopfield network with binary neurons attain **stability earlier** than bipolar neurons. This is because the values of diagonals do not **have an impact**

when neuron state is 0 for binary neurons.

3.12 CONCLUSION

In this chapter some experimental observations and theoretical conclusions of the study of **Hopfield** network is reported. It is concluded that the changes in diagonal elements and the threshold elements can be used for tuning the Hopfield network to obtain **re-**quired performance. Critical values of diagonal elements (**no-change-critical** value and **sure-change-critical** value) and threshold elements (θ_i^P and θ_i^M) for bipolar neurons have been estimated. It has been observed that **pairwise** conditional **no-change-critical-value** is smaller than the independent **no-change-critical-value**. It is also observed that pairwise conditional **sure-change-critical-value** is greater than the independent **sure-change-critical-value**. The **effect** of diagonal element changes in a network with binary **neurons** have been observed to be different from the effect of these changes on a network with bipolar neurons. The effect of diagonal changes in Hopfield network on the energy **func-**tion have been studied.

Chapter 4

POLYHEDRAL COMBINATORICS FOR NEURAL NETWORKS

4.1 INTRODUCTION

In this chapter the techniques of polyhedral combinatorics are used to analyse the geometry of stable state vectors and its associative image. Polyhedral combinatorics is a tool developed during later half of 1970s by OR researchers. This tool was developed with the hope of analysing and perhaps developing efficient solution techniques for hard combinatorial optimization problems. Polyhedral combinatorics, as conventionally used by OR researchers, can be defined as a set of methodologies to describe the geometrical and combinatorics aspect of feasible region which is a convex *polytope*. In this study the characterization of extreme points and their adjacency relationships are investigated. In addition the *hyperplanes* which define a face of a specific dimension for the polytope are also characterised. This technique proves to be very useful in studying integer and combinatorial programming problems. It is also used to analyse the complexity of the wellknown Simplex method.

The technique polyhedral combinatorics is used in this chapter to **propose** a possible learning technique for **Hopfield** network. The underlying idea is to transform the **N** dimensional state vector to $\frac{N(N+1)}{2}$ dimensions, where the energy **function** becomes a linear function, the supporting hyperplanes of the convex hull of 2^N state vectors **defines** a **synaptic** matrix having stable states as those points which the **hyperplane** touches. In other words, the quadratic character of the energy **function** is transformed to a linear

function in higher dimension. In this higher dimension an attempt is made to study the geometry of the set of 2^N state vectors which forms a **hypercube** in N dimension. The convex hull of these points are taken as a **polytope** and the energy function corresponds to a supporting **hyperplane** of this polytope. The set of stable states can be visualised as the points in the polytope which touch the supporting hyperplane. Thus designing of supporting hyperplane touching a specified set of points will result in constructing a **Hopfield** network $(J, 0)$ having specified set of points as stable states. Thus polyhedral combinatorics approach not only provides a better insight into the problem but also helps to a certain extent training of Hopfield network. Based on this discussion some results are presented in this chapter.

In this work the scope of application of polyhedral combinatorics is restricted to the design of Hopfield with binary neurons $(0, 1)$ operating with asynchronous mode of updating. Some definitions and notations are given in Section 4.2 to provide a basic background for polyhedral combinatorics. Earlier attempts to use polyhedral combinatorics for study and design of neural network are reported in Section 4.3 . In Section 4.4 an attempt is made to explain the basis of the present work. Section 4.5 describes the construction process to make any given state vector as the only stable state in the Hopfield network. The construction process of Hopfield network with two stable **states** is given in Section 4.6 . The sequence in which the two state vectors are considered in the construction process has a bearing on the dynamical behaviour of the Hopfield network. This issue is being reported in Section 4.7 . Techniques of polyhedral combinatorics are further extended to make more than two candidate state vectors stable in Hopfield **net-**work. The construction process is **given** in Section 4.8. In this section an attempt is also made to study the extent to which the present work can be used. Section 4.9 deals with the mechanism to make all the state vectors **upto** a specific number of 1 bits as stable. The conclusions of the attempt to use polyhedral combinatorics techniques for design of

Hopfield network are reported in Section 4.10.

4.2 POLYHEDRAL COMBINATORICS

In this section provides the basic background for polyhedral combinatorics and necessary definitions and notations are introduced [Nemhauser88].

Polyhedron:- A polyhedron $P \subset \mathbb{R}^n$ is a set of points that satisfies a finite number of linear inequalities.

That is, $P = \{x \in \mathbb{R}^n : Ax \leq b\}$, where (A, b) is an $m \times (n + 1)$ matrix.

Polytope:- A polyhedron $P \subset \mathbb{R}^n$ is bounded if there exists an $\omega \in \mathbb{R}_+^1$ such that $P \subset \{x \in \mathbb{R}^n : -\omega \leq x_j \leq \omega \text{ for } j = 1, 2, \dots, n\}$. A bounded polyhedron is called a polytope.

Extreme Point:- $x \in P$ is an extreme point of P if there do not exist $x^1, x^2 \in P, x^1 \neq x^2$, such that x is convex combination of x^1 and x^2 .

Valid Inequality:- The inequality $\pi x \leq \pi_0$ [or (π, π_0)] is a valid inequality for P if it is satisfied by all points in P . It is to be noted that (π, π_0) is a valid inequality if and only if P lies in the half-space $\{x \in \mathbb{R}^n : \pi x \leq \pi_0\}$, or equivalently if and only if $\max \{\pi x : x \in P\} \leq \pi_0$.

Face:- If (π, π_0) is a valid inequality of P , and $F = \{x \in P : \pi x = \pi_0\}$, F is called a face of P , and it is said that (π, π_0) represents F . A face P is said to be proper if $F \neq \emptyset$ and $F \neq P$.

Supporting Hyperplane:- A face represented by (π, π_0) is nonempty if and only if $\max \{\pi x : x \in P\} = \pi_0$. When F is nonempty, it can be said that (π, π_0) supports P . Some authors also term the support as supporting hyperplane.

An extreme point can also be defined using the concept of supporting hyperplane.

Extreme Point:- x is an extreme point of polytope P if and only if there exists a

supporting hyperplane F of P such that F touches P only at x . In other words, $F(\pi, \pi_0)$ is such that x is only point in P with $\pi x = \pi_0$.

If such a supporting hyperplane is defined as $\pi x = \pi_0$ and if all points y in P satisfy $\pi y > \pi_0$ then x is a minimizing extreme point of πx in P . Similarly, by the above argument, if (π, π_0) is a supporting hyperplane touching the extreme points $x^1, x^2, \dots, x^r$ then each of these extreme points is a minimizing point of πx in P . In other words, πx attains its minimum value at $x^1, x^2, \dots, x^r$.

This particular concept is used in showing that supporting hyperplane also provides a set of minimizing points. In the following sections it is shown that energy function associated with Hopfield network (though quadratic in nature) can be transformed to a linear function in higher dimension namely $\mathbb{R}^{n+1}$ and the set of convex hull of set of stable states defines the polytope P . So using the above analysis, that is, by defining a supporting hyperplane, the minimizing points are identified for a given linear function which corresponds to a supporting hyperplane. This in turn defines minimizing points of energy function and hence, defines a set of stable points.

4.3 SIMILAR WORKS

The design of neural networks amenable to linear programming and combinatorial methodology has been noted in literature [Delsatre89, Hao91, Kamp91, Chandru93, Budinich91, Shonkwiler93]. Using the standard techniques of polyhedral combinatorics, a polynomial-time algorithm for designing a neural network is proposed in [Chandru93]. This algorithm gives maximum radius of direct attraction around arbitrary input state vectors. A new sufficient condition that a region be classifiable by a two-layer feed-forward network using threshold activation functions is obtained in [Shonkwiler93]. This condition is obtained

using polyhedral combinatorics by considering classification as characterising the two-set-partitions of the vertices of a hypercube which are separable by a **hyperplane**. The problems of feed-forward neural networks have been related to the theory of **n-dimensional** convex **polytopes** in [Budinich91]. The typical problem is to synthesize a network that is capable of reproducing a set of examples. The learning process thus leads a set of hyperplanes that isolates at least the given examples. It is shown in [Budinich91] that the convex hull of the examples can provide a feed-forward network that solves the problem without uncontrolled generalizations.

4.4 BASIS OF THE PRESENT WORK

The energy function E , associated with the Hopfield network is given by

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N J_{ij} \sigma_i \sigma_j + \sum_{i=1}^N \theta_i \sigma_i \quad (4.1)$$

where σ_i is 0 or 1.

The stable states of the network also corresponds to the local minima of the **energy** function over the hypercube defined by

$$\sigma_i = 0, 1 \quad \forall i = 1, 2, \dots, N$$

By introducing a new variable σ_{ij} and substituting this for the term $\sigma_i \sigma_j$ (and considering $(\sigma_i)^2 = \sigma_i, \forall i$) the quadratic energy function E defined by Equation 4.1 becomes a linear function in $\frac{N(N+1)}{2}$ variables ($\frac{N(N-1)}{2}$ variables of the form $\sigma_{ij}, i \neq j$ and N variables of the form σ_i). Hence,

$$E = -\sum_{i=1}^N \sum_{j < i}^N J_{ij} \sigma_{ij} + \sum_{i=1}^N (\theta_i - \frac{1}{2} J_{ii}) \sigma_i$$

The set vertices of the **hypercube** is now visualised as a set points in the space $\frac{N+1}{2}$ and $E = k$ defines a **hyperplane**. It may be noted that these points do not form a hypercube any longer.

Let $\mathcal{H}$ be the convex hull of this set of points in $\frac{N+1}{2}$ dimension. It is shown in [Pujari83] that each of the 2^N state vectors is an extreme point of $\mathcal{H}$. In other words, given any state vector (ξ) it is possible to construct a supporting hyperplane for $\mathcal{H}$ touching it at ξ only. So based on above discussion a network having just a single stable state (any one of 2^N) can be designed.

For some value of k , the hyperplane $E = k$ defines a face for $\mathcal{H}$ if all the points of $\mathcal{H}$ lie on one side of the plane $E = k$ i.e., $E < k$ for all 2^N points or $E > k$ for all 2^N points. In addition if the face touches the convex set $\mathcal{H}$, then it is said that the face is a support of $\mathcal{H}$.

Hence if the state vectors ξ^1, ξ^2 , etc are to be the stable states then any learning rule would aim at constructing the synaptic matrix (equivalently E), so that the specified set of state vectors are local minima of E . In the present context (constructing E) this can be achieved by obtaining a hyperplane which becomes a support for $\mathcal{H}$ touching it at vertices ξ^1, ξ^2 , etc. The aim of this work is to construct such **Hopfield** network making use of polyhedral characteristics of $\mathcal{H}$ and the supporting **hyperplanes**.

4.5 HOPFIELD NETWORK FOR ONE STABLE VECTOR

In this section a formulation is proposed for construction of a Hopfield network having any one given binary state vector as the only stable state. This formulation is based on the concept mentioned in Section 4.4 that given any ξ it is possible to construct a supporting hyperplane for $\mathcal{H}$ touching it at ξ only. The proposed formulation is for neural networks with binary neurons. The update mechanism used is asynchronous and

maximum local field is used as the basis of selection of neuron to be updated.

4.5.1 Construction of J^ξ and θ

Let ξ^1 be a binary state vector of N dimension. The aim is to construct a **Hopfield** network (J^ξ, θ) such that the state vector ξ^1 is the only stable state in the network. J^ξ is $N \times N$ synaptic matrix and θ is $N \times 1$ threshold vector.

Let $S_1 = \{i : \xi_i^1 = 1\} \forall i = 1, 2, \dots, N$ and $S_2 = \{i : \xi_i^1 = 0\} \forall i = 1, 2, \dots, N$. The synaptic matrix is constructed using the following formulation. The diagonal elements J_{ii} are given by

$$J_{ii} = \begin{cases} 1 & \text{if } i \in S_1 \\ -1 & \text{if } i \in S_2 \end{cases}$$

The off diagonal elements J_{ij} , $i \neq j$ are

$$J_{ij} = \begin{cases} 1/2 & \text{if } i, j \in S_1 \\ -N^3 & \text{otherwise} \end{cases}$$

The threshold vector θ is given by $\theta_i = -0.5, \forall i = 1, 2, \dots, N$.

Example

Let $\xi^1 = 1011101100$ be a state vector which is to be made as the only stable state of a Hopfield network with $N = 10$ neurons. Then $S_1 = \{1, 3, 4, 5, 7, 8\}$ and $S_2 = \{2, 6, 9, 10\}$. Based on the formulation mentioned above the following synaptic matrix J^ξ and threshold vector θ are constructed.

$$J^{\epsilon} = \begin{pmatrix} 1 & -N^3 & 0.5 & 0.5 & 0.5 & -N^3 & 0.5 & 0.5 & -N^3 & -N^3 \\ -N^3 & -1 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 \\ 0.5 & -N^3 & 1 & 0.5 & 0.5 & -N^3 & 0.5 & 0.5 & -N^3 & -N^3 \\ 0.5 & -N^3 & 0.5 & 1 & 0.5 & -N^3 & 0.5 & 0.5 & -N^3 & -N^3 \\ 0.5 & -N^3 & 0.5 & 0.5 & 1 & -N^3 & 0.5 & 0.5 & -N^3 & -N^3 \\ -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -1 & -N^3 & -N^3 & -N^3 & -N^3 \\ 0.5 & -N^3 & 0.5 & 0.5 & 0.5 & -N^3 & 1 & 0.5 & -N^3 & -N^3 \\ 0.5 & -N^3 & 0.5 & 0.5 & 0.5 & -N^3 & 0.5 & 1 & -N^3 & -N^3 \\ -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -1 & -N^3 \\ -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -1 \end{pmatrix}$$

Threshold vector is given by

$$\theta^T = (-0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5)$$

It can be seen that all the 2^{10} input state vectors converge to **1011101100** as the **stable** state.

Using this formulation it has been experimentally verified with very large number of samples that it is possible to make any of the 2^N state vectors as the only stable state in the **Hopfield** network with N binary neurons. However, state vector with all neuron states as zero is the only state vector that cannot be made the only **stable** state of the Hopfield network with this formulation. But with suitable modification this limitation may be overcome.

4.6 HOPFIELD NETWORK WITH TWO STABLE VECTORS

In this section the concept of making any one state vector as a stable state of Hopfield network is extended to two state vectors. This formulation is based on the concept

mentioned in Section 4.4 that it is possible to construct an energy function so that the hyperplane obtained becomes a support for $\mathcal{H}$ touching it at two vertices only. The issue of Hopfield network having two vectors differing by one bit as stable has been addressed in [Bruck88, Prados89]. The formulation proposed in this section enables to have any two state vectors stable in the resulting Hopfield network, even if the two state vectors differ in only one bit. The proposed formulation is for Hopfield networks with binary neurons. The update mechanism used is asynchronous and maximum local field is used as the basis of selection of neuron to be updated.

4.6.1 Construction of $J^{\xi^1 \xi^2}$ and 0

Let ξ^1 and ξ^2 be two binary vectors of N dimension. The aim is to construct a $N \times N$ synaptic matrix $J^{\xi^1 \xi^2}$ and a threshold vector 0 such that the Hopfield network $(J^{\xi^1 \xi^2}, 0)$ has ξ^1 and ξ^2 as stable state vectors. The construction process is given below.

Let $G(\xi^j) = \{i : \xi_i^j = 1\}$ where $i = 1, 2, \dots, N$ and $j = 1, 2$. Then ,

$$S_1 = G(\xi^1) \cap G(\xi^2)$$

$$s_1 = |S_1|$$

$$S_2 = G(\xi^1) \setminus G(\xi^2)$$

$$s_2 = |S_2|$$

$$S_3 = G(\xi^2) \setminus G(\xi^1)$$

$$s_3 = |S_3|$$

A temporary threshold vector $\tilde{0}$ is constructed as described below.

1. For $i \in S_1$, $\tilde{\theta}_i = \frac{1}{2}(s_1 + s_2 + 1)$.
2. For $i \in S_2$, $\tilde{\theta}_i = \frac{1}{2}(s_1 + s_2 + 1)$.
3. For $i \in S_3$, $\tilde{\theta}_i = \frac{s_2}{2 \times s_3}(s_1 + s_2 + 1)$.

4. For $i \in S_4$, $\theta_i = -N^3$.

The synaptic matrix J^{ξ^1, ξ^2} is constructed as described below. The diagonal elements are given by

$$J_{ii} = \begin{cases} 0.5 - \theta_i & \text{if } i \in G(\xi^1) \\ 0.5 + \frac{s_2}{s_3} - \theta_i & \text{if } i \in S_3 \\ -N^3 & \text{otherwise} \end{cases}$$

and the off-diagonal elements J_{ij} , $i \neq j$ are

$$J_{ij} = \begin{cases} \frac{1}{2} & \text{if } i, j \in G(\xi^1) \\ \frac{s_2}{2s_3} & \text{if } i \in S_1, j \in S_3 \text{ or } i \in S_3, j \in S_1 \\ 1 + \frac{s_2(s_2+1)}{2s_3(s_3-1)} & \text{if } i, j \in S_3 \\ -N^3 & \text{otherwise} \end{cases}$$

The threshold vector θ is given by $\theta_i = -0.5$, $\forall i = 1, 2, \dots, N$.

Using this formulation it has been experimentally verified with large number of samples that the Hopfield network so constructed has only ξ^1 and ξ^2 as the stable states. It can be seen that, the above formulation is valid if $s_3 > 1$.

Example

Let $\xi^1 = 11111100000$ and $\xi^2 = 11110011100$ be two state vectors to be made as the stable states of a neural network with $N = 11$ binary neurons. The values of S_1, S_2, S_3, S_4 , s_1, s_2, s_3 , and s_4 are given below. The entries are truncated for convenience of representation.

$$S_1 = \{1, 2, 3, 4\}, \quad s_1 = 4$$

$$S_2 = \{5, 6\}, \quad s_2 = 2$$

$$S_3 = \{7, 8, 9\}, \quad s_3 = 3$$

$$S_4 = \{10, 11\}, \quad s_4 = 2$$

The **synaptic** matrix $J^{\xi^1 \xi^2}$ and threshold vector θ constructed using the formulation are given below

$$J^{\xi^1 \xi^2} = \begin{pmatrix} -3.0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.3 & 0.3 & 0.3 & -N^3 & -N^3 \\ 0.5 & -3.0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.3 & 0.3 & 0.3 & -N^3 & -N^3 \\ 0.5 & 0.5 & -3.0 & 0.5 & 0.5 & 0.5 & 0.3 & 0.3 & 0.3 & -N^3 & -N^3 \\ 0.5 & 0.5 & 0.5 & -3.0 & 0.5 & 0.5 & 0.3 & 0.3 & 0.3 & -N^3 & -N^3 \\ 0.5 & 0.5 & 0.5 & 0.5 & -3.0 & 0.5 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & -3.0 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 \\ 0.3 & 0.3 & 0.3 & 0.3 & -N^3 & -N^3 & -4.2 & 1.2 & 1.2 & -N^3 & -N^3 \\ 0.3 & 0.3 & 0.3 & 0.3 & -N^3 & -N^3 & 1.2 & -4.2 & 1.2 & -N^3 & -N^3 \\ 0.3 & 0.3 & 0.3 & 0.3 & -N^3 & -N^3 & 1.2 & 1.2 & -4.2 & -N^3 & -N^3 \\ -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 \\ -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 \end{pmatrix}$$

$$\theta^T = (-0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5)$$

It can be seen that all the 2^{10} input state vectors converge to either 11111100000 or 11110011100 as the stable state. Using this formulation it has been experimentally verified with very large number of samples that it is possible to make any two of the 2^N state vectors as the only stable states in the Hopfield network with N binary neurons.

4.7 ORDER OF VECTORS IN TWO STABLE STATE FORMULATION

In the formulation of Hopfield network proposed in Section 4.6.1 it is observed that the Hopfield network construction is dependent on the sequence in which the vectors are considered. Consider two Hopfield networks constructed using the **formulation given in** Section 6.6. Hopfield network $(J^{\xi^1 \xi^2}, 0)$ constructed by considering ξ^1 as the first **vector**

and ξ^2 as the second vector in the formulation and Hopfield network $(J^{\xi^2 \xi^1}, 6)$ constructed by considering ξ^2 as the first vector and ξ^1 as the second vector in the formulation.

For the example under consideration the the network $(J^{\xi^1 \xi^2}, 0)$ is given in Section 6.6 . The network $(J^{\xi^2 \xi^1}, \theta)$ is given here.

$$J^{\xi^2 \xi^1} = \begin{pmatrix} -3.5 & 0.5 & 0.5 & 0.5 & 0.8 & 0.8 & 0.5 & 0.5 & 0.5 & -N^3 & -N^3 \\ 0.5 & -3.5 & 0.5 & 0.5 & 0.8 & 0.8 & 0.5 & 0.5 & 0.5 & -N^3 & -N^3 \\ 0.5 & 0.5 & -3.5 & 0.5 & 0.8 & 0.8 & 0.5 & 0.5 & 0.5 & -N^3 & -N^3 \\ 0.5 & 0.5 & 0.5 & -3.5 & 0.8 & 0.8 & 0.5 & 0.5 & 0.5 & -N^3 & -N^3 \\ 0.8 & 0.8 & 0.8 & 0.8 & -4.0 & 2.5 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 \\ 0.8 & 0.8 & 0.8 & 0.8 & 2.5 & -4.0 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 \\ 0.5 & 0.5 & 0.5 & 0.5 & -N^3 & -N^3 & -3.5 & 0.5 & 0.5 & -N^3 & -N^3 \\ 0.5 & 0.5 & 0.5 & 0.5 & -N^3 & -N^3 & 0.5 & -5.5 & 0.5 & -N^3 & -N^3 \\ 0.5 & 0.5 & 0.5 & 0.5 & -N^3 & -N^3 & 0.5 & 0.5 & -5.5 & -N^3 & -N^3 \\ -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 \\ -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 \end{pmatrix}$$

It is observed that the two networks have the same pair of state vectors as the only stable states. However, there is slight difference in the dynamics of the two networks.

Observations

Some observations regarding the dynamics of two Hopfield networks have been made by conducting several experiments with asynchronous mode of operation and selection of neuron having maximum local field for updating. For the two networks there is difference in sequence from the input vector to stable vectors for some input vectors. For some input vectors stable state is not reached. The state of the network oscillates between two states, i.e., the network reaches a bi-directionally stable state.

4.7.1 Multiple Weights

The two Hopfield networks $(J^{\xi^1 \xi^2}, 0)$ and $(J^{\xi^2 \xi^1}, 0)$ have the same set of stable states. In this subsection an attempt is made to visualize these two Hopfield networks as belonging to a class of Hopfield networks with same set of stable states. It has been observed that the networks $(J^{\xi^1 \xi^2}, 0)$ and $(J^{\xi^2 \xi^1}, 0)$ are defined by two extreme matrices which can be generated using the expression $J^{\xi \xi} = \lambda J^{\xi^1 \xi^2} + (1 - \lambda) J^{\xi^2 \xi^1}$, where $0 < \lambda < 1$.

THEOREM 4.1 :- If $(J^{\xi^1 \xi^2}, \theta)$ is the Hopfield network memorizing ξ^1 followed by ξ^2 and $(J^{\xi^2 \xi^1}, \theta)$ is the Hopfield network memorizing ξ^2 followed by ξ^1 then, $(J^{\xi \xi}, \theta)$ where $J^{\xi \xi} = \lambda J^{\xi^1 \xi^2} + (1 - \lambda) J^{\xi^2 \xi^1}$ with $0 < \lambda < 1$ is also a Hopfield network memorizing ξ^1 and ξ^2 .

Proof:- The energy function of the Hopfield network $(J^{\xi^1 \xi^2}, 0)$ is given by

$$E(J^{\xi^1 \xi^2}, \theta) = \xi^T J^{\xi^1 \xi^2} \xi - \theta^T \xi$$

Similarly, the energy function of the Hopfield network $(J^{\xi^2 \xi^1}, \theta)$ is given by

$$E(J^{\xi^2 \xi^1}, \theta) = \xi^T J^{\xi^2 \xi^1} \xi - \theta^T \xi$$

$$\begin{aligned} E(J^{\xi \xi}, \theta) &= \xi^T J^{\xi \xi} \xi - \theta^T \xi \\ &= \xi^T [\lambda J^{\xi^1 \xi^2} + (1 - \lambda) J^{\xi^2 \xi^1}] \xi - \theta^T \xi \\ &= \lambda \xi^T J^{\xi^1 \xi^2} \xi + (1 - \lambda) \xi^T J^{\xi^2 \xi^1} \xi - \theta^T \xi \\ &= \lambda \xi^T J^{\xi^1 \xi^2} \xi - \lambda \theta^T \xi + (1 - \lambda) \xi^T J^{\xi^2 \xi^1} \xi - (1 - \lambda) \theta^T \xi \\ &= \lambda E(J^{\xi^1 \xi^2}) + (1 - \lambda) E(J^{\xi^2 \xi^1}) \\ &= E(J^{\xi^1 \xi^2}) \text{ because } E(J^{\xi^1 \xi^2}) = E(J^{\xi^2 \xi^1}) \end{aligned}$$

□

By construction of J^1 and J^2 , the values of $E(J^1)$ at ξ^1 and ξ^2 is same as those of $E(J^2)$ at ξ^1 and ξ^2 respectively. Since ξ^1 and ξ^2 are the minimizing points of $E(J^1)$ and

also the minimizing point of $E(J^2)$, the following can be inferred. $E(J^1)$ at ξ is $> E(J^1)$ at ξ^1 and also ξ^2 . Similarly, $E(J^2)$ at ξ is $> E(J^2)$ at ξ^1 and also ξ^2 .

Considering the above two observation and also considering the property of convex combinations, it can be concluded that $E(J)$ at ξ^1 and ξ^2 is same as those of $E(J^1)$ and $E(J^2)$.

$E(J)$ at ξ is $> E(J)$ at ξ^1 and also ξ^2 . Hence ξ^1, ξ^2 are the minimizing points of $E(J)$. Thus the network $(J^{\xi}, 0)$ also has ξ^1 and ξ^2 as the only set of stable states.

4.8 HOPFIELD NETWORK WITH MORE STABLE VECTORS

In this section the techniques of polyhedral combinatorics is extended to derive formulation for construction of Hopfield network with three or more candidate state vectors as stable. However, after the selection of two vectors without restriction there are some restrictions on the selection of the third or subsequent candidate state vectors.

4.8.1 Three Stable Vectors

The formulation for making three candidate state vectors stable in the Hopfield network is given in this subsection. The three candidate state vectors satisfying the following conditions can be made stable in the Hopfield network.

1. If $\xi_i^1 = \xi_i^2 = 1$ then $\xi_i^3 = 1$
2. ξ^3 should have two additional bits as 1 where the two vectors ξ^1 and ξ^2 have 0.

4.8.2 Construction of $J^{\xi^1 \xi^2 \xi^3}$ and 0

Let ξ^1 and ξ^2 be two binary vectors of N dimension. A state vector ξ^3 is selected considering the restriction mentioned above. The aim is to construct a $-V \times N$ synaptic

matrix $J^{1,2,3}$ and a threshold vector θ such that the Hopfield network with $(J^{1,2,3}, \theta)$ has ξ^1 , ξ^2 and ξ^3 as stable vectors. The construction process is given below.

The sets $S_1, S_2, S_3, S_4, S_5, S_6, S_7$ and $s_1, s_2, s_3, s_4, s_5, s_6, s_7$ are defined as described in section 4.6.1. Some more sets of indices are defined as follows.

$$\begin{aligned} S_5 &= G(\xi^1) \cap G(\xi^3) \\ s_5 &= |S_5| \\ S_6 &= G(\xi^1) \setminus G(\xi^3) \\ s_6 &= |S_6| \\ S_7 &= G(\xi^3) \setminus G(\xi^1) \\ s_7 &= |S_7| \end{aligned}$$

A temporary threshold vector θ is constructed as described below.

1. For $i \in S_1, \theta_i = \frac{1}{2}(s_1 + s_2 + 1)$.
2. For $i \in S_2, \theta_i = \frac{1}{2}(s_1 + s_2 + 1)$.
3. For $i \in S_3, \theta_i = \frac{s_2}{2 \times s_3}(s_1 + s_2 + 1)$.
4. For $i \in S_7, \theta_i = \frac{s_6}{2 \times s_7}(s_5 + s_6 + 1)$.
5. For $i \in S_4$, and $i \notin S_7, \theta_i = -N^3$.

The synaptic matrix $J^{1,2}$ is constructed as described below. The diagonal elements are given by

$$J_{ii} = \begin{cases} 0.5 - \theta_i & \text{if } i \in G(\xi^1) \\ 0.5 + \frac{s_2}{s_3} - s_3 - \theta_i & \text{if } i \in S_3 \\ 0.5 + \frac{s_6}{s_7} - s_7 - \theta_i & \text{if } i \in S_7 \\ -N^3 & \text{otherwise} \end{cases}$$

and the off-diagonal elements J_{ij} , $i \neq j$ are

$$J_{ij} = \begin{cases} \frac{1}{2} & \text{if } i, j \in G(\xi^1) \\ \frac{s_2}{2s_3} & \text{if } i \in S_1, j \in S_3 \text{ or } i \in S_3, j \in S_1 \\ 1 + \frac{s_2(s_2+1)}{2s_3(s_3-1)} & \text{if } i, j \in S_3 \\ \frac{s_6}{257} & \begin{cases} \text{if } i \in S_5 \text{ and } j \in S_7 \\ \text{or if } i \in S_7 \text{ and } j \in S_5 \end{cases} \\ 1 + \frac{s_6(s_6-1)}{2s_7(s_7-1)} & \text{if } i, j \in S_7 \\ -N^3 & \text{otherwise} \end{cases}$$

The threshold vector θ is given by $\theta_i = -0.5$, $\forall i = 1, 2, \dots, N$.

The Hopfield network so constructed has only ξ^1 , ξ^2 and ξ^3 as the stable states. The update mechanism used is asynchronous and maximum local field is used as the **basis** of selection of neuron to be updated. The above formulation is valid if $s_7 > 1$ and $s_3 > 1$.

Example

Let, $\xi^1 = 11111100000$, $\xi^2 = 11110011100$ and $\xi^3 = 11110000010$ be three state vectors which are to be made stable states of a Hopfield network with $N = 11$ binary neurons.

For this example the following **synaptic** matrix and threshold vector **is constructed** using the formulation given above.

$$J^{\xi^1 \xi^2 \xi^3} = \begin{pmatrix} -3.0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.3 & 0.3 & 0.3 & 1.0 & -N^3 \\ 0.5 & -3.0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.3 & 0.3 & 0.3 & 1.0 & -N^3 \\ 0.5 & 0.5 & -3.0 & 0.5 & 0.5 & 0.5 & 0.3 & 0.3 & 0.3 & 1.0 & -N^3 \\ 0.5 & 0.5 & 0.5 & -3.0 & 0.5 & 0.5 & 0.3 & 0.3 & 0.3 & 1.0 & -N^3 \\ 0.5 & 0.5 & 0.5 & 0.5 & -3.0 & 0.5 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & -3.0 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 \\ 0.3 & 0.3 & 0.3 & 0.3 & -N^3 & -N^3 & -4.2 & 1.2 & 1.2 & -N^3 & -N^3 \\ 0.3 & 0.3 & 0.3 & 0.3 & -N^3 & -N^3 & 1.2 & -4.2 & 1.2 & -N^3 & -N^3 \\ 0.3 & 0.3 & 0.3 & 0.3 & -N^3 & -N^3 & 1.2 & 1.2 & -4.2 & -N^3 & -N^3 \\ -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -5.5 & -N^3 \\ -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 \end{pmatrix}$$

$$\theta^T = (-0.5 \ -0.5 \ -0.5 \ 0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5 \ -0.5)$$

It is observed that all the 2^{10} input state vectors have one of the vectors 11111100000, 11110011100 or 11110000010 as the stable state. Using this formulation it has been experimentally verified with very large number of samples that it is possible to make three vectors out of the 2^N state vectors as the only stable states in the neural network with N binary neurons. The three state vectors should satisfy the conditions mentioned above.

4.8.3 More Than Three Stable Vectors

The above formulation can be generalised to have more stable vectors. For instance, four state vectors can be made stable by designing a **Hopfield** network by having a **synaptic** matrix so that the four stable vectors satisfy the following conditions.

1. If $\xi^1 = 1, \xi^2 = 2, \xi^3 = 3$ then $\xi^4 = 1$.
2. ξ^4 must have at least two additional bits as one where all three vectors ξ^1, ξ^2 , and ξ^3 have zero.

As long as this condition is valid it is possible to generalise the formulation of Hopfield network to have any number of stable vectors as stable. Assume that all the vectors have **atleast** three common bits having one satisfying condition 1. For Hopfield network of N neurons there is a maximum of $\frac{N-3}{2}$ vectors **satisfying** these conditions. Hence a Hopfield network can be designed following the above formulation having $\frac{N-3}{2}$ stable vectors. The capacity of the network in this context can be $\frac{N-3}{2}$.

4.9 STABLE VECTORS WITH SPECIFIC NUMBER OF 1 BITS

The formulations proposed in the **previous** sections can be extended to store all vectors characterized by the number of 1 bits. In this section a formulation to make all vectors with less than or equal to a specific number of 1 bits as stable states of Hopfield network is proposed. That is to make all vectors having less than or equal to L ($0 < L < N$) number of 1 bits stable in Hopfield network. The Hopfield network for this purpose can be constructed as follows.

The diagonal elements of **synaptic** matrix are

$$* = -\frac{2}{L+1}$$

The off-diagonal elements $J_{ij}, i \neq j$ are

$$J_{ij} = -\frac{2}{L(L+1)}$$

$$\theta_i = 0.1 \forall i = 1, 2, \dots, N$$

Example

All vectors **upto** $L = 5$ bits are to be made stable states of a Hopfield network with $N = 10$ binary neurons. For this example following **synaptic** matrix $\mathbf{J}^L$ and threshold vector is constructed.

$$\begin{pmatrix} 0.333 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 \\ -0.067 & 0.333 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 \\ -0.067 & -0.067 & 0.333 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 \\ -0.067 & -0.067 & -0.067 & 0.333 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 \\ -0.067 & -0.067 & -0.067 & -0.067 & 0.333 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 \\ -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & 0.333 & -0.067 & -0.067 & -0.067 & -0.067 \\ -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & 0.333 & -0.067 & -0.067 & -0.067 \\ -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & 0.333 & -0.067 & -0.067 \\ -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & 0.333 & -0.067 \\ -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & 0.333 \end{pmatrix}$$

$$\theta^T = (0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1)$$

It can be seen that all the state vectors with 5 or **less** number of bits **as** 1 are the stable states of the Hopfield network. Using this formulation it **has** been experimentally verified with very large number of **samples** that it is **possible** to **construct a Hopfield** network having all state vectors with L ($0 < L < N$) number of **bits** $a^* 1$, **as** stable states.

4.10 CONCLUSION

The learning rules proposed in **this** chapter **has non-zero** diagonal element* in the **synaptic** matrix. The majority of the popular learning rules have additional **stable** states **besides**

the candidate state vectors. However, the learning rules proposed in this chapter has exactly the specified state vectors as the stable states. No other state vector becomes stable in the network.

In **comparision** to **Hebbian** learning rule the learning rule proposed in Section 6.4 is not commutative. A commutative learning rule can be considered as a rule using which the **Hopfield** network so constructed is not **affected** by the sequence in which the candidate state vectors are considered. However, in reality, the efficiency of learning is dependent on the sequence in which the system learns. Hence, a commutative learning is a desirable property of learning and hence may give a better insight to brain function.

Chapter 5

CONCLUSION

An attempt to study and investigate the dynamics of Hopfield network has been made as part of this research work. This dissertation reports some new theoretical and experimental results of this study. Chapter 2 reports a survey of various capacity estimates of Hopfield class networks. Results of study of dynamics of Hopfield network with self-feedback are given in Chapter 3. Analysis of geometry of stable state vectors of Hopfield network is reported in Chapter 4. Some learning techniques have also been proposed based on the results of this analysis in Chapter 4. The achievements, the limitations and the further research plans are summarized in this chapter.

5.1 SURVEY ON CAPACITY

In the recent years many models of neural networks have been proposed. A model may perfectly suit the purpose for which it is designed, but in the presence of many models, a need for comparative study was felt. Many research articles related to capacity of neural networks have been reported on several diversified disciplines. Chapter 2 reports the various models that have been proposed to attain a better **performance** of Hopfield network. The diversification in the research approaches can also be observed in **that** chapter. Large variations in nature of these approaches **necessitates** a comparative **study** to evaluate each approach. This survey has been carried out **considering capacity** as a common criteria for evaluating the performance of various models **belonging to Hopfield** class networks.

The survey as reported in chapter 2 considers only one of the several models of neural networks, namely **Hopfield** network and there are very large number of research articles reported in literature concentrating only on this single model. Only those research articles which were fitting in the framework have been considered for this survey. The complicated nature of this task restricted the expansion of the scope of this survey within limited time and hence, may not be termed as a complete survey. Some aspects of capacity of Hopfield network have not been included due to the limited scope of the study. Even for the aspects considered, it was not possible to access all possible related research literature. As a future work an attempt to include more aspects of capacity of Hopfield network can be made. The study can also be extended to include other models of neural networks. A survey of all neural network models with a generalized framework can be attempted to give a better insight into the dynamics of neural networks. This survey is a humble attempt towards the need for a unified and universal criteria for evaluation of various neural networks. Such criteria if devised, will serve as an important tool for the design, development and application of neural networks. An attempt in this area will be of significant use in the area of neural networks.

5.2 STUDY OF SELF-FEEDBACK IN HOPFIELD NETWORKS

The concept of self-feedback in Hopfield network has been selected for a **detailed** study. Some experimental observations and theoretical conclusions of this study are reported in Chapter 3. An attempt has been made to **understand** the impact of **diagonal** and threshold element changes on set of stable states of Hopfield network. Some critical values of the relationship has been determined. It is concluded that the changes in diagonal elements and threshold elements can be used for tuning Hopfield networks to obtain require performance. These results will help in **maintaining** a balance **between**

information storage and equivocation. These results may also be helpful in design of Hopfield networks with high selectivity.

A further study to determine exact tuning mechanism can be attempted. Such an attempt will **atleast** partially help in determining the type of diagonal element **changes** required to have a particular set of candidate state vectors stable, to eliminate **specific** or all spurious states and to determine the basins of attraction of a stable state.

5.3 POLYHEDRAL COMBINATORICS

Analysis of the geometry of stable state vectors using the polyhedral combinatorics techniques is reported in Chapter 4. Based on this analysis some learning techniques have been proposed for Hopfield network. The learning rules proposed have non-zero diagonal elements in the **synaptic** matrix. These **learnig** rule have exactly the specified state vectors as the stable states. The set of candidate state vectors and the set of stable states of network **constructed** using these learning rules are same. However, there are **restrictions** on the selection of candidate state vectors.

As a future work this study can be extended to design learning rules for higher order and multivalued neural networks.

Bibliography

- [Abumostafa85] Abu-Mostafa, Y. S., & St.Jacques, J. (1985). Information capacity of the Hopfield Model. *IEEE Transactions on Information Theory*, 31, 461-464.
- [Ackley85] Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A Learning Algorithm for Boltzmann machines. *Cognitive Science* 9, 147-169.
- [Amari77] Amari, S. (1977). Neural theory of association and concept formation. *Biological Cybernetics*, 26, 175-185.
- [Amit85a] Amit, D. J., Gutfreud, H., & Sompolinsky, H. (1985). Spin-glass model of Neural networks. *Physical Review A*, 32, 1007-1018.
- [Amit85b] Amit, D. J., Gutfreud, H., & Sompolinsky, H. (1985). Storing infinite numbers of patterns in a Spin Glass model of neural networks. *Physical Review Letters*, 55, 1530-1533.
- [Amit87] Amit, D. J., Gutfreud, H., & Sompolinsky, H. (1987). Information storage in neural networks with low level activity. *Physical Review A*, 35, 2293-2303.
- [Amit89] Amit, D. J. (1989). *Modelling Brain Function*. Cambridge:Cambridge University Press.
- [Anderson72] Anderson, J. A. (1972). A simple neural network generating an interactive memory. *Mathematical Biosciences*, 14, 197-200.
- [Anderson88] Anderson, J. A., & Rosenfeld, E. (Eds.), (1988). *Neurocomputing: Foundations of Research*. Cambridge:MIT press.
- [Anderson92] Anderson, J.A. (1992). Foreword. In B. Kosko *Neural Networks and fuzzy systems: A dynamical systems approach to machine intelligence*. New Jersey:Printice-Hall Inc.
- [Baldi87] Baldi, P., & Venkatesh, S. S. (1987). Number of stable points for spin-glasses and neural networks of higher orders. *Physical Review A*, 38, 913-916.

- [Baram91] Baram, Y. (1991). On the capacity of ternary **Hebbian** networks. *IEEE Transactions of Information Theory*, 37, 528-534.
- [Bolle91] Bolle, D., Dupont, P., & Mourik, J. V. (1991). Stability properties of Potts neural network with biased patterns and low loading. *Journal of Physics A*, 24, 1065-1081.
- [Bolle92] Bolle, D., Dupont, P., & Vinck, B. (1992). On the overlap **dynamics** of **multi-state** neural networks with a finite **number** of patterns. *Journal of Physics A*, 25, 2859-2872.
- [Braham88] Braham, R., & Hamblen, J. O. (1988). On the behaviour of some associative neural networks. *Biological Cybernetics*, 60, 145-151.
- [Bruck88] Bruck, J., & Sanz, J. (1988). A study of neural networks. *International Journal of Intelligent Systems*, 3, 59-75.
- [Budinich91] Budinich, M., & Milotti, E. (1991). Feed-forward neural networks: A geometrical perspective. *Journal of Physics A*, 24, 881-888.
- [Buhmann89] Buhmann, J., Divko, R., & Schulten, K. (1989). Associative memory with high information content. *Physical Review A*, 39, 2689-2692.
- [Carpenter90] Carpenter, R. H. S. (1990). *Neurophysiology*. London: **Edward** Arnold.
- [Chandru93] Chandru, V. & Vinay, V. (1993). Constructing highly attractive recursive neural networks. In M. Vidyasagar (Ed). *Proceedings of the International conference on intelligent Robotics* (pp. 307-315). New Delhi: Tata McGraw-Hill Publishing Company Ltd..
- [Chen86] Chen, H. H., Lee, Y. C., Sun, G. Z., Lee, H. Y., Maxwell, T., & Giles, C. L. (1986). Higher order correlation model for associative memory. *AIP Conference Proceedings*, 151, 86-99.
- [Cook89] Cook, J. (1989). The **mean-field** theory of a **Q-state** neural model. *Journal of Physics A*, 22, 2057-2067.
- [Cooper73] Cooper, L. N. (1973). A possible organization of animal memory and learning. In B. Lundquist & S. Lundquist (Eds.), *Proceedings of the Nobel Symposium on Collective Properties* (pp. 252-264). New York: Academic Press Inc..
- [Crisanti88] Crisanti, A., & Sompolinsky, H. (1988). Dynamics of **spin systems** with randomly asymmetric bonds, **Ising spins** and Glauber dynamics. *Physical Review A*, 37, 4865-4874.

- [Delsatre89] Delsatre, P., & Kamp, Y. (1989). Low rank matrices with given **sign** patterns. *SIAM Journal of Discrete Mathematics*, 2, 51-63.
- [Der92] Der, R., Dotsenko, V. S., & Tirozzi. (1992). Modified **pseudo-inverse** neural network storing correlated patterns. *Journal of Physics A*, 25, 2843-2857.
- [Derrida87] Derrida, B., Gardner, E., & Zippelius, A. (1987). An exactly solvable asymmetric neural network model. *Europhysics Letters*, 4, 167-173.
- [Derrida89] Derrida, B. (1989). Distribution of the activities in a diluted neural network. *Journal of Physics A*, 22, 2069-2080.
- [Dotsenko91] Dotsenko, V. S., Yarunin, N. D., & Dorotheyev, E. A. (1991). Statistical mechanics of **Hopfield-like** neural networks with modified interactions. *Journal of Physics A*, 24, 2419-2429.
- [Eckmiller90] Eckmiller, R. (1990). The design of intelligent robots as a federation of geometric machine. In S. F. Zornetzer, J. L. Davis, C. Lau (Eds), *An introduction to neural and Electronic networks* (pp. 109-128). San Diego: Academic Press Inc..
- [Elderfield83] Elderfield, D., & Sherrington, D. (1983). **Spinglass**, ferromagnetic and mixed phases in the disordered Potts model. *Journal of Physics A*, 16, L971-L977.
- [Evans89] Evans, M. R. (1989). Random dilution in a neural network for biased patterns. *Journal of Physics A*, 22, 2103-2118.
- [Farrell90] Farrell, J. A., & Michel, A. N. (1990). A **synthesis procedure** for Hopfield's continuous-time associative memory. *IEEE Transactions on Circuits and Systems*, 37, 877-884.
- [Feigelman86] Feigelman, M. V., & Ioffe, L. B. (1986). The statistical properties of the **Hopfield** model of memory. *Europhysics Letters*, 1, 197-201.
- [Fontanari88] Fontanari, J. F., & Koberle, R. (1988). **Enhancing** the learning of a finite number of patterns in neural networks. *Journal of Physics A*, 21, L253-L257.
- [Fukai92] Fukai, T., & Shuno, M. (1990). Comparative study of **spurious-state** distribution in analogue neural networks and **Boltzmann machine**. *Journal of Physics A*, 25, 2873-2887.

- [Garces92] **Garces, R., Kuhlmann, P., & Eissfeller, H.** (1992). In search of an **optimal** dilution algorithm for feed-forward networks. *Journal of Physics A*, 25, **L1335-L1342**.
- [Gardner88a] Gardner, E. (1988). The space of interactions in neural network models. *Journal of Physics A*, 21, 257-270.
- [Gardner88b] Gardner, E., & Derrida, B. (1988). Optimal storage properties of neural network models. *Journal of Physics A*, 21, 271-284.
- [Gardner89a] Gardner, E. (1989). Optimal basins of attraction in randomly **sparse** neural network models. *Journal of Physics A*, 22, **1969-1974**.
- [Gardner89b] Gardner, E., & Derrida, B. (1989). The probability distribution of the partition of the random energy model. *Journal of Physics A*, 22, 1975-1981.
- [Gardner89c] Gardner, E., Derrida, B. (1989). Three unfinished works on the optimal storage capacity of networks. *Journal of Physics A*, 22, 1983-1994.
- [Gardner89d] Gardner, E., Gutfreund, H., & Yekutieli, I. (1989). The phase space of interactions in neural networks with definite symmetry, *Journal of Physics A*, 22, 1995-2008.
- [Gardner89e] Gardner, E., Mertens, S., & Zippelius, A. (1989). Retrieval properties of a neural network with **asymmetrical** learning rule. *Journal of Physics A*, 22, 2009-2018.
- [Gardner89f] Gardner, E., Storoud, N., & Wallace, D. J. (1989). Training with noise and the storage of correlated patterns in a neural network model, *Journal of Physics A*, 22, 2019-2030.
- [Gindi88] Gindi, G. R., Gmitro, A. F., & Parthasarathy, K. (1988). Hopfield model associative memory with **nonzero-diagonal** terms in memory **ma-**trix. *Applied Optics*, 27, 129-134.
- [Grossberg80] Grossberg, S. (1980). How **does** a brain build a cognitive code? *Psychological Review*, 87, 1-51.
- [Gutfreund88] Gutfreund, H. (1988). Neural network with **hierarchically correlated** patterns, *Physical Review A*, 37, **570-577**.
- [Hao91] Hao, J., Tan, S., & Vanderwalle, J. (1991). A new **approach** to **design** of Hopfield associative memory. *Proceedings of International Joint Conference on Neural Network*, 1705-1710.

- [Hebb49] **Hebb, D.** (1949) *The Organization of Behavior*. New York: Wiley.
- [Hecht-Nielsen91] **Hecht-Nielsen, R.** (1991). *Neuro computing*. Addison-Wesley Publishing Company.
- [Hendrich91] **Hendrich, N.** (1991). Associative memory in damaged neural networks. *Journal of Physics A*, 24, 2877-2887.
- [Hinton84] **Hinton, G. E., Sejnowski, T. J., & Ackley, D. H.** (1984). **Boltzmann** machines: Constraint satisfaction networks that learn. *Carnegie Mellon University Technical Report Number CMU-CS-84-119*, **Caranegie Mellon University**, USA.
- [Hopfield82] **Hopfield, J. J.** (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA*, 79, 2554-2558.
- [Hopfield84] **Hopfield, J. J.** (1984). Neurons with graded response having collective computational properties like those of **two-state** neurons, *Proceedings of the National Academy of Sciences, USA*, 81, 3088-3092.
- [Hopfield86] **Hopfield, J. J., & Tank, D. W.** (1986). Computing with neural circuits: A model. *Science*, 233, 625.
- [Houselander90] **Houselander, P., & Taylor, J. T.** (1990). Calculating the upper and lower bounds on the capacity of two modified **Hebbian** trained Hopfield networks, *Electronics Letters*, 16, 1266-1267.
- [Kamp91] **Kamp, Y., & Hasler, M.** (1991). *Recursive neural networks as associative memory*. New York: John Wiley & Sons Ltd.
- [Kanter87] **Kanter, J., & Sompolinsky, H.** (1987). Associative recall of memory without errors. *Physical Review A*, 35, 380-392.
- [Kanter88] **Kanter, J. D.** (1988). Potts-glass model of neural networks. *Physical Review A*, 37, 2739-2742.
- [Karayiannis93] **Karayiannis, N. B., & Venetsanopoulos, A. N.** (1993). *Artificial neural networks: Learning algorithms, performance evaluation and applications*. **Boston:Kluwer Academic Publishers**.
- [Katz66] **Katz, B.** (1966). *Nerve, Muscle and synapses*. New York: McGraw-Hill Inc.

- [Kohonen72] Kohonen, T. (1972). Correlation matrix memories. *IEEE Transactions on Computers*, 21, 353-359.
- [Kohonen84] Kohonen, T. (1984). *Self-organization and associative memory*. Berlin:Springer-Verlag.
- [Kohonen88] Kohonen, T. (1988). An introduction to neural computing. *Neural Networks*, 1, 3-16.
- [Kosko92] Kosko, B. (1992). *Neural networks and fuzzy systems: A dynamical systems approach to machine intelligence*. New Jersey: **Printice-Hall** Inc.
- [Kree92] Kree, R., Widmaier, D., & Zippelius, A. (1992). Spin-glass phase in a neural network with asymmetric couplings. *Journal of Physics A*, 21, L1181-L1186.
- [Kuhlmann92] Kuhlmann, P., Garces, R., & Eissfeller, H. (1992). A dilution **algorithm** for neural networks. *Journal of Physics A*, 25, L593-L598.
- [Lewenstein92] Lewenstein, M., & Tarkowski, W. (1992). Optimal storage of correlated patterns in **neural-network** memories. *Physical Review A*, 46, 2136-2142.
- [Little75] Little, W. A., & Shaw, G. L. (1975). A Statistical theory of Short and Long Term Memory. *Behaviour Biology*, 14, 115-133.
- [Marcus90] Marcus, C. M., & Westervert, R. M. (1990). Stability and convergence of analog neural networks with **multiple-time-step** parallel dynamics. *Physical Review A*, 42, 2410-2417.
- [McCulloch43] McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biosciences*, 5, 115-133.
- [McEliece87] McEliece, R. L., Posner, E. C., Rodemich, E. R., & Venkatesh, S. S. (1987). The capacity of the **Hopfield** associative memory. *IEEE Transactions on Information Theory*, 33, 461-482.
- [Minsky69] Minsky, M., & Papert, S. (1969). *Perceptrons*. Cambridge, MA: **MIT** Press.
- [Miyashita88] Miyashita, Y. (1988). **Neuronal** correlate of **visual** associative long **term** memory in the primate temporal cortex. *Nature*, 335, 817-819.

- [Montgomery86] Montgomery, B. L., & Vijayakumar, B. V. K. (1986). Evaluation of the use of the **Hopfield** neural network model in nearest-neighbour algorithm. *Applied Optics*, 25, 3759-3766.
- [Mori89] **Mori, Y., Davis, P., & Nara, S.** (1989). Pattern retrieval in an asymmetric neural network with embedded limit cycle. *Journal of Physics A*, 22, 2525-2532.
- [Nadal86] **Nadal, J. P.** (1986). Networks of formal neurons and memory palimpsests. *Europhysics Letters*, 1, 535-542.
- [Nemhauser88] Nemhauser, G. L., & Wolsey, L. A. (1988). *Integer and Combinatorial Optimization*. New York: John Wiley & Sons Ltd.
- [Newell58] Newell, A., Shaw, J. C., & Simon, H. A., (1958). Elements of the theory of human problem solving, *Physiological Review*, 65, 151-166.
- [Nicolis90] **Nicolis, S.** (1990). Retrieval properties of neural networks with infinitely many marked patterns. *Europhysics Letters*, 12, 583-587.
- [Nobre86] **Nobre, F. D., & Sherrington, D.** (1986). The infinite-range clock spin glass model : An introduction of the relevance of reflection symmetry. *Journal of Physics C*, 19, L181-L188.
- [Noest88a] **Noest, A. J.** (1988). Discrete-state **phasor** neural networks. *Physical Review A*, 38, 2196-2199.
- [Noest88b] Noest, A. J. (1988). Associative memory in sparse phasor neural network. *Europhysics Letters*, 6, 469-474.
- [Palm90] Palm, G. (1990). Local learning rules and sparse coding in **neural** networks, In **R. Eckmiller** (Ed.) *Advanced neural computers*, (pp 145-150). Amsterdam: **Elsevier** Science Publishers.
- [Parisi86a] Parisi, G. (1986). A memory that forgets. *Journal of Physics A*, 19, 2617-2620.
- [Parisi86b] Parisi, G. (1986). Neural networks and the process of learning. *Journal of Physics A*, 19, L675-L680.
- [Patrick90] Patrick, A. E., & Zagrebnov, V. A. (1990). **Parallel dynamics** for **extremely** diluted neural network, *Journal of Physics A*, 23, L1323-L1329.

- [Patrick91a] Patrick, A. E., & Zagrebnov, V. A. (1991). A probabilistic **approach** to parallel dynamics for the **Little-Hopfield** model. *Journal of Physics A*, 24, 3413-3426.
- [Patrick91b] Patrick, A. E., Picco, P., Ruiz, J., & Zagrebnov, V. A. (1991). Main overlap dynamics for **multistate** neural networks. *Journal of Physics A*, 24, L637-L647.
- [Patrick92] Patrick, A. E., & Zagrebnov, V. A. (1992). Parallel dynamics for an extremely diluted network, *Journal of Physics A*, 25, **1009-1011**.
- [Penna90] Penna, T. J. P., & de Oliveira, P. M. C. (1990). Enhancing retrieval of low activity patterns in neural networks. *Europhysics Letters*, **11**, 191-194.
- [Personnaz85] Personnaz, L., Guyon, I., & Dreyfus, G. (1985). *Journal of Physics (Paris) Letters*, 46, 359.
- [Prados89] Prados, D. L., & Kak, S. C. (1989). Neural network capacity using Delta Rule. *Electronic Letters*, 25, 197-199.
- [Prados] Prados, D., & Kak, S. Shift invariant associative memory. *Technical Report, Louisiana State University, USA*.
- [Pujari83] Pujari, A. K., Mittal, A. K., & Gupta, S. K. (1983). A convex **polytope** of diameter one. *Discrete and Applied Mathematics*, 5, 241-242.
- [Rau92] Rau, A., Wong, K. Y. M., & Sherrington, D. (1992). Pattern selectivity in optimized neural networks. *Europhysics Letters*, 17, 649-654.
- [Rieger90] Rieger, H. (1990). Storing an extensive number of gray-toned patterns in a neural network using multistate neurons. *Journal of Physics A*, 23, **L1273-L1279**.
- [Rochester56] Rochester, N., Holland, J. H., Haibt, L. H., & Duda, W. L. (1956). Tests on a cell assembly theory of action of the brain, using a large digital computer. *IRE Transactions on Information Theory*, 2, **80-93**.
- [Rosenblatt58] Rosenblatt, F. (1958). The **Perceptron**: A **probabilistic model** for information storage and organization in the Brain. *Psychological Review*, 65, **386-408**.

- [Rumelhart86] Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1986). A General Framework for Parallel Distributed Processing. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 1, (pp. 45-76). Cambridge, MA:MIT Press.
- [Ryle49] Ryle, G. (1949). *The Concept of Mind*. London:Hutcheson.
- [Sezan90] Sezan, M. I., Stark, H., & Yeh, S. (1990). Projection method formulation of Hopfield-type associative memory neural networks. *Applied Optics*, 29, 2616-2622.
- [Shonkwiler93] Shonkwiler, R. (1993). Separating the vertices of N-cubes by hyperplanes and its applications to artificial neural networks. *IEEE Transactions on Neural Networks*, 4, 343-347.
- [Sompolinsky86] Sompolinsky, H. (1986). Neural Networks with nonlinear synapses and a static noise. *Physical Review A*, 34, 2571-2574.
- [Soucek89] Soucek, B. (1989). *Neural and concurrent real-time systems: The sixth generation*. New York: John Wiley & Sons Ltd.
- [Stark90] Stark, J., & Bressloff, P. (1990). Two-state representations of three-state neural networks. *Journal of Physics A*, 22, 1633-1644.
- [Stiefvater92] Stiefvater, Thomas., Muller, & Klaus-Robert. (1992). A finite size scaling investigation for Q-state Hopfield models: storage and capacity and basin of attraction. *Journal of Physics A*, 25, 5919-5929.
- [Sussmann89] Sussmann, H. J. (1989). On the number of memories that can be perfectly stored in a neural net with Hebb weights. *IEEE Transactions on Information Theory*, 35, 174-178.
- [Tarkowski92] Tarkowski, W., & Lewenstein, M. (1992). Estimates of optimal storage conditions in neural network memories based on random matrix theory. *Journal of Physics A*, 25, 6251-6264.
- [Tarkowski93] Tarkowski, W., & Lewenstein, M. (1993). Storage of sets of correlated data in neural network memories. *Journal of Physics A*, 26, 2453-2469.
- [Thompson85] Thompson, R. F. (1985). *The Brain: An introduction to neuroscience*. New York:W.H.Freeman & Company.
- [Thornton92] Thornton, C. J. (1992). *Techniques in Computational Learning: An Introduction*. London:Chapman & Hall Computing.

- [Toulouse86] Toulouse, G., Dehaene, S., & Changeux, J. P. (1986). Spin glass model of learning by selection. *Proceedings of National Academy of Sciences, USA*, 83, 1695-1698.
- [Tsodyks88] Tsodyks, M. V., & Feigelman, M. V. (1988). The enhanced storage capacity in neural networks with low activity level. *Europhysics Letters*, 6, 101-105.
- [Viswanathan93] Viswanathan, R. Raju. (1993). Neural networks with biased bipolar
• synapses and biased patterns. *Journal of Physics A*, 26, 873-881.
- [Vogt92] Vogt, H., & Zippelius, A. (1992). Invariant recognition in Potts-glass neural networks. *Journal of Physics A*, 25, 2209-2226.
- [Wendemuth93] Wendemuth, A., Oppen, M., & Kinzel, W. (1993). The effect of correlations in neural networks, *Journal of Physics A*, 26, 3165-3185.
- [Widrow60] Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits. In 1960 IRE WESCON Convention Record (pp. 96-104). New York.
- [Wittgenstein53] Wittgenstein, L. (1953). *Philosophical Investigations*, (English Third Edition). New York:Macmillan.
- [Wong91] • Wong, K. Y. M., Kohn, P. E., & Sherrington, D. (1991). A neural network model of working memory exhibiting primacy and recency. *Journal of Physics A*, 24, 1119-1135.
- [Yanai90] Yanai, H., & Sawada, Y. (1990). Associative memory composed of neurons with hysteretic property, *Neural Networks*, 3, 223-228.
- [Yedidia89] Yedidia, J. S. (1989). Neural networks that use three state neurons. *Journal of Physics A*, 22, 2265-2273.
- [Youn89] Youn, C. H., & Kak, S. C. (1989). Continuous unlearning in neural networks. *Electronic Letters*, 25, 202-203.
- [Zornetzer90] Zornetzer, S. F., et al. (Eds.), (1990). *An Introduction to Neural and Electronic Networks* Academic Press.

RESEARCH PAPERS

1. Sharma, R. & Pujari, A. K. (1990). Text Retrieval using Neural Network. In V.P. Bhatkar and K.M. Rege [Eds.], Frontiers in Knowledge-Based Computing. (187-196), New Delhi: Narosa Publishing House.
2. Pujari, A. K., & Sharma, R. (1990). Modified Hopfield Model of Neural Network, in A. Pedar et.al.[Eds.], Real Time System. (348-352), SGSITS, Indore.
3. Pujari, A. K., & Sharma, R. (1990). Hopfield Model of Neural Networks with Unrestricted Self-Feedback, Proceedings of International Neural Network Conference INNC-90. (872), Paris.