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 network. 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 up to 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 \( P \subseteq \mathbb{R}^n \) is a set of points that satisfies a finite number of linear inequalities.

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

**Polytope:** A polyhedron \( P \subseteq \mathbb{R}^n \) is bounded if there exists an \( \omega \in \mathbb{R}^n \) such that
\[ P \subseteq \{ \mathbf{x} \in \mathbb{R}^n : -\omega < x_j < \omega \text{ for } j = 1, 2, \ldots, n \}. \]
A bounded polyhedron is called a polytope.

**Extreme Point:** \( \mathbf{x} \in P \) is an extreme point of \( P \) if there do not exist \( \mathbf{x}^1, \mathbf{x}^2 \in P, \mathbf{x}^1 \neq \mathbf{x}^2 \), such that \( \mathbf{x} \) is convex combination of \( \mathbf{x}^1 \) and \( \mathbf{x}^2 \).

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

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

**Supporting Hyperplane:** A face represented by \( (\pi, \pi_0) \) is nonempty if and only if maximum \( \{ \pi \mathbf{x} : \mathbf{x} \in P \} = \pi_0 \). When \( F \) is nonempty, it can be said that \( (\pi, \pi_0) \) 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, $F(x, \pi_0)$ 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 $\pi x$ in $P$. Similarly, by the above argument, if $(\pi, \pi_0)$ is a supporting hyperplane touching the extreme points $x^1, x^2, \ldots, x^r$ then each of these extreme points is a minimizing point of $\pi x$ in $P$. In other words, $\pi x$ attains its minimum value at $x^1, x^2, \ldots, x^r$.

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 $x^T A x + b^T x$ 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 [Delsat89, 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

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

where $\sigma_i$ 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

$$\sigma_i = 0, 1 \ \forall \ i = 1, 2, \ldots, N$$

By introducing a new variable $\sigma_{ij}$ 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 $\sigma_i$). 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 \( \mathbb{R}^{N+1} \) 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 \( \mathbb{R}^{N+1} \) 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 \( (\xi) \) it is possible to construct a supporting hyperplane for \( \mathcal{H} \) touching it at \( \xi \) 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 \( \xi^1, \xi^2, \) 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 \( \xi^1, \xi^2, \) 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 \( \xi \) it is possible to construct a supporting hyperplane for \( \mathcal{H} \) touching it at \( \xi \) only. The proposed formulation is for neural networks with binary neurons. The update mechanism used is asynchronous and
Chapter 4  POLYHEDRAL COMBINATORICS FOR NEURAL NETWORKS  

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

4.5.1 Construction of $J^t$ and $0$

Let $\xi^1$ be a binary state vector of $N$ dimension. The aim is to construct a Hopfield network $(J^t, \theta)$ such that the state vector $\xi^1$ is the only stable state in the network. $J^t$ is $N \times N$ synaptic matrix and $0$ is $N \times 1$ threshold vector.

Let $S_1 = \{i : \xi_i^1 = 1\} \forall i = 1, 2, \ldots, N$ and $S_2 = \{i : \xi_i^1 = 0\} \forall i = 1, 2, \ldots, N$. 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 $\theta$ is given by $\theta_i = -0.5, \forall i = 1, 2, \ldots, N$.

Example

Let $\xi^1 = 101101100$ 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 $J^t$ and threshold vector $0$ are constructed.
Threshold vector is given by

\[ \theta^T = (-0.5, -0.5, -0.5, -0.5, -0.5, -0.5, -0.5, -0.5) \]

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
Chapter 4  POLYHEDRAL COMBINATORICS FOR NEURAL NETWORKS

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 $J^\xi \xi^2$ and $0$

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

Let $G(\xi_i) = \{i : \xi_i = 1\}$ where $i = 1, 2, \ldots, N$. Then,

\[
S_1 = G(\xi^1) \cap G(\xi^2), \quad s_1 = |S_1|, \\
S_2 = G(\xi^1) \setminus G(\xi^2), \quad s_2 = |S_2|, \\
S_3 = G(\xi^2) \setminus G(\xi^1), \quad s_3 = |S_3|.
\]

A temporary threshold vector $\delta$ is constructed as described below.

1. For $i \notin S_1$, $\delta_i = \frac{1}{2}(s_1 + s_2 + 1)$.
2. For $i \in S_2$, $\delta_i = \frac{1}{2}(s_1 + s_2 + 1)$.
3. For $i \in S_3$, $\delta_i = \frac{s_2}{s_2 + 1}(s_1 + s_2 + 1)$. 

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

The synaptic matrix \( J^* \) is constructed as described below. The diagonal elements are given by

\[
J_{ii} = \begin{cases} 
0.5 - \theta_i & \text{if } i \in G(\xi) \\
0.5 + \frac{3}{2} - \ldots - \frac{3}{2} & \text{if } i \in S_3 \\
-N^3 & \text{otherwise}
\end{cases}
\]

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

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

The threshold vector \( \theta \) is given by \( \theta_i = -0.5, V i = 1, 2, \ldots, N \).

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

Example

Let \( \xi^1 = 11111100000 \) and \( \xi^2 = 1110011100 \) 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.

\[
\begin{align*}
S_1 &= \{1, 2, 3, 4\}, & s_1 &= 4 \\
S_2 &= \{5, 6\}, & s_2 &= 2 \\
S_3 &= \{7, 8, 9\}, & s_3 &= 3 \\
S_4 &= \{10, 11\}, & s_4 &= 2
\end{align*}
\]
Chapter 4 POLYHEDRAL COMBINATORICS FOR NEURAL NETWORKS

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

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

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

It can be seen that all the $2^{10}$ input state vectors converge to either $11111100000$ or $1110011100$ 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 $(J^1, \xi, 0)$ constructed by considering $\xi^1$ as the first vector
and $\mathbf{\xi}^2$ as the second vector in the formulation and

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

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

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

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 \((J^1, \xi^1, 0)\) and \((J^2, \xi^2, 0)\) 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 \((J^3, \xi^3, 0)\) and \((J^4, \xi^4, 0)\) are defined by two extreme matrices which can be generated using the expression \(J^\xi = \lambda J^1 \xi^1 + (1 - \lambda) J^2 \xi^2\), where \(0 < \lambda < 1\).

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

**Proof**: The energy function of the Hopfield network \((J^1, \xi^1, 0)\) is given by

\[ E(J^1, \xi, \theta) = \xi^T J^1 \xi - \theta^T \xi \]

Similarly, the energy function of the Hopfield network \((J^2, \xi^1, 0)\) is given by

\[ E(J^2, \xi, \theta) = \xi^T J^2 \xi - \theta^T \xi \]

\[ E(J^\xi, \theta) = \xi^T J^\xi \xi - \theta^T \xi \]
\[ = \xi^T \left[ \lambda J^1 \xi^1 + (1 - \lambda) J^2 \xi^2 \right] \xi - \theta^T \xi \]
\[ = \lambda \xi^T J^1 \xi^1 \xi + (1 - \lambda) \xi^T J^2 \xi^2 \xi - \theta^T \xi \]
\[ = \lambda \xi^T J^1 \xi^1 \xi - \lambda \theta^T \xi + (1 - \lambda) \xi^T J^2 \xi^2 \xi - \theta^T \xi \]
\[ = \lambda E(J^1, \xi) + (1 - \lambda) E(J^2, \xi) \]
\[ = E(J^\xi, \theta) \]

because \(E(J^1, \xi) = E(J^2, \xi)\). 

By construction of \(J^1\) and \(J^2\), the values of \(E(J^1)\) at \(\xi^1\) and \(\xi^2\) is same as those of \(E(J^2)\) at \(\xi^1\) and \(\xi^2\) respectively. Since \(\xi^1\) and \(\xi^2\) 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 $\xi$ is $> E(J^1)$ at $\xi^1$ and also $\xi^2$. Similarly, $E(J^2)$ at $\xi$ is $> E(J^2)$ at $\xi^1$ and also $\xi^2$.

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

$E(J)$ at $\xi$ is $> E(J)$ at $\xi^1$ and also $\xi^2$. Hence $\xi^1 \xi^2$ are the minimizing points of $E(J)$. Thus the network $(J^\xi, 0)$ also has $\xi^1$ and $\xi^2$ 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 $\xi_1^3 = \xi_2^3 = 1$ then $\xi_3^3 = 1$

2. $\xi^3$ should have two additional bits as 1 where the two vectors $\xi^1$ and $\xi^2$ have 0.

#### 4.8.2 Construction of $J^{\xi_1 \xi_2 \xi_3}$ and $0$

Let $\xi_1$ and $\xi_2$ be two binary vectors of $N$ dimension. A state vector $\xi^3$ is selected considering the restriction mentioned above. The aim is to construct a $-V \times N$ synaptic
matrix $J^{123}$ and a threshold vector $\theta$ such that the Hopfield network with $(J^{12}, J^3, \theta)$ has $\xi^1$, $\xi^2$ and $\xi^3$ as stable vectors. The construction process is given below.

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

\[
\begin{align*}
S_5 &= G(\xi^1) \cap G(\xi^3) \\
S_6 &= |S_1| \\
S_7 &= G(\xi^1) \setminus G(\xi^3) \\
S_8 &= |S_2| \\
S_9 &= G(\xi^3) \setminus G(\xi^1) \\
S_{10} &= |S_3|
\end{align*}
\]

A temporary threshold vector $\hat{\theta}$ is constructed as described below.

1. For $i \in S_1$, $\hat{\theta}_i = \frac{1}{2}(s_1 + s_2 + 1)$.
2. For $i \in S_2$, $\hat{\theta}_i = i(s_1 + s_2 + 1)$.
3. For $i \in S_3$, $\hat{\theta}_i = \frac{s_1}{2x_{23}}(s_1 + s_2 + 1)$.
4. For $i \in S_7$, $\hat{\theta}_i = \frac{s_2}{2x_{37}}(s_5 + s_6 + 1)$.
5. For $i \in S_4$, and $i \notin S_7$, $\hat{\theta}_i = -N^3$.

The synaptic matrix $J^{123}$ is constructed as described below. The diagonal elements are given by

\[
J_{ii} = \begin{cases} 
0.5 - \theta & \text{if } i \in G(\xi^1) \\
0.5 + \frac{s_1}{s_2} - s_3 - \theta & \text{if } i \in S_3 \\
0.5 + \frac{s_3}{s_2} - s_7 - \hat{\theta} & \text{if } i \in S_7 \\
-N^3 & \text{otherwise}
\end{cases}
\]
and the off-diagonal elements \( J_{ij}, i \neq j \) are

\[
J_{ij} = \begin{cases} 
\frac{1}{2} & \text{if } i, j \in G(\xi^1) \\
\frac{m}{2s_3} & \text{if } i \in S_4, j \in S_3 \text{ or } i \in S_3, j \in S_4 \\
1 + \frac{\alpha(s_3+1)}{2s_4(s_3-1)} & \text{if } i, j \in S_4 \\
\frac{m}{2s_7} & \left\{ \begin{array}{l}
\text{if } i \in S_5 \text{ and } j \in S_7 \\
\text{or if } i \in S_7 \text{ and } j \in S_5
\end{array} \right.
\end{cases}
\]

\[
1 + \frac{\alpha(s_7-1)}{2s_4(s_7-1)} & \text{if } i, j \in S_7 \\
-N^3 & \text{otherwise}
\]

The threshold vector \( \theta_i = -0.5 \), \( \forall i = 1, 2, \ldots, N \).

The Hopfield network so constructed has only \( \xi^1, \xi^2 \) and \( \xi^3 \) 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.
\[ J^{\dagger} J = \begin{pmatrix}
-3.0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.3 & 0.3 & 1.0 & -N^3 \\
0.5 & -3.0 & 0.5 & 0.5 & 0.5 & 0.3 & 0.3 & 1.0 & -N^3 \\
0.5 & 0.5 & -3.0 & 0.5 & 0.5 & 0.3 & 0.3 & 1.0 & -N^3 \\
0.5 & 0.5 & 0.5 & -3.0 & 0.5 & 0.3 & 0.3 & 1.0 & -N^3 \\
0.5 & 0.5 & 0.5 & 0.5 & -3.0 & 0.5 & 0.3 & 1.0 & -N^3 \\
0.5 & 0.5 & 0.5 & 0.5 & 0.5 & -3.0 & 0.5 & -N^3 & -N^3 \\
0.3 & 0.3 & 0.3 & 0.3 & -N^3 & -N^3 & -N^3 & -N^3 & -N^3 \\
0.3 & 0.3 & 0.3 & 0.3 & -N^3 & -N^3 & 1.2 & -4.2 & 1.2 & -N^3 & -N^3 \\
0.3 & 0.3 & 0.3 & 0.3 & -N^3 & -N^3 & 1.2 & -4.2 & 1.2 & -N^3 & -N^3 \\
\end{pmatrix} \]

\[ \theta^T = (-0.5 - 0.5 - 0.5 \cdots 0.5 - 0.5 - 0.5 - 0.5 - 0.5 - 0.5 - 0.5) \]

It is observed that all the 2^10 input state vectors have one of the vectors 1111100000, 1111011100 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.
Chapter 4  POLYHEDRAL COMBINATORICS FOR NEURAL NETWORKS

1. If \( \xi^1 = 1, \xi^2 = 2, \xi^3 = 3 \) then \( \xi^4 = 1 \).

2. \( \xi^4 \) must have at least two additional bits as one where all three vectors \( \xi^1, \xi^2, \) and \( \xi^3 \) 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 at least 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

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

\( \theta_i = 0.1 \forall i = 1, 2, \ldots, N \)
Example

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

\[
J^L = \begin{pmatrix}
0.333 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 \\
-0.067 & 0.333 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 \\
-0.067 & -0.067 & 0.333 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 \\
-0.067 & -0.067 & -0.067 & 0.333 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 \\
-0.067 & -0.067 & -0.067 & -0.067 & 0.333 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 \\
-0.067 & -0.067 & -0.067 & -0.067 & -0.067 & 0.333 & -0.067 & -0.067 & -0.067 & -0.067 \\
-0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & 0.333 & -0.067 & -0.067 & -0.067 \\
-0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & 0.333 & -0.067 & -0.067 \\
-0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & 0.333 & -0.067 \\
-0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & -0.067 & 0.333 \\
\end{pmatrix}
\]

\[
\theta^T = (0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1)
\]

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 as 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 comparison 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.