School of Graduate studies, Addis Ababa University

Faculty of Computer and Mathematical Sciences

Department of Mathematics

A Project on

RECURRENT NEURAL NETWORK

By Kassahun Tesfaye

Advisor: Dr. Berhanu Bekele

A project Report submitted to the Department of Mathematics in Partial fulfillment of the requirements for the degree of Master of Science in Mathematics.

January, 2011

Addis Ababa, Ethiopia

Declaration

“I declare that this project has been composed by me and that no part of the project has formed the basis for the award of any Degree, Diploma, Associate ship, Fellowship or any other similar to me.

Signature____________________”

Permission

“This is to certify that this project is a record of the research work done by Kassahun Tesfaye in the Department of Mathematics, Addis Ababa University, under my supervision. I hereby also confirm that the project can be submitted for evaluation by examiners and eventual defense.

Advisor Signature______________________”

Acknowledgments

First and foremost I want to praise my God, the Almighty God, who has passed me many unspeakable situations throughout my life from the very beginning until this time and also, knows about my future.

I would like to present my excellent gratitude and grateful respect to my advisor Dr.Berhanu Bekele for his critical reading of the paper and providing me constructive comments, assistance, material support and encouragement.

I would also like to thank Mr. Yonas Abebe for reading the paper and giving me constructive suggestions.

My praise also goes to my elder brother kisses Yaekob Bernabas for his being of my mother in place of my mother, giving me the strength and material support during my stay time at the universities.

Abstract

The purpose of the study was to investigate **Neural Networks** in general and **Recurrent Neural Network** in particular. The investigation was to construct a model that shows the complex data processing of these Neural Networks and their functions in a simple way by using Artificial Neural Networks. Therefore, in order to carry out this task, Perceptron, Feed Forward and Recurrent Neural Networks, and the basic concept of Biological Neural Networks, which are among the contents of this paper, are very important.

From a biological point of view, **Neural Networks** are networks of biological neurons. Basically, they are networks of sensory, motor and associative neurons functionally, and axon, cell body and dendrites structurally. Now, when an external or internal signal reaches the sensory neuron through the receptors, then it passes to the motor neuron through the associative neuron within the central nervous system. Since motor neuron has connection with the brain, then it immediately fires through the command that comes from the brain. This means that it gives response for that particular signal. In this paper we extend these concepts to mathematical model by using the concepts of mathematics.

For instance, a Heaviside function is a simple mathematical model of Neural Network. On the other hand, Recurrent Neural Network consists of the graph $G = (V, E)$ and a family of formal neurons $(X_i, Y_i, \sigma_i, s_i)$, each associated to one of the Vertices $i \in V$. This paper also provides details about the types of Neural Networks present in human being and also provides details about the Artificial Neural Networks. This paper also presents some Applications of Neural Networks.

Table of contents

List of Mathematical Symbols

Mathematical Symbols used in this paper are the following:

Symbols	**Meaning**
1. $\Rightarrow$	Implies
2. $\Leftrightarrow$	If and only if
3. $\therefore$	Therefore
4. $\cup$	Union
5. $\cap$	Intersection
6. $\cdots$	Continues
7. $\subseteq$	is a subset of or equal to
8. $\mathbb{R}$	The set of real numbers.
9. $\in$	Is an element of
10. $\sum$	Summation
11. $\mathbb{Z}$	The set of integers.
12. $-\infty$	Negative Infinity.
13. ∞	Positive Infinity
14. $\subset$	Is a subset of
15. $\mathbb{N}$	The set of natural numbers.
16. $\equiv$	Identical to.
17. $\exists$	There exist.
18. $n \times m$	Matrices n rows with m columns.

List of Abbreviations

Abbreviations	**Meaning**
1. ANNs	Artificial Neural Networks
2. BioNNs	Biological Neural Networks
3. FFNN	Feed Forward Neural Network
4. NNs	Neural Networks
5. WTS	we want to show
6. PL problem	Perceptron learning problem

List of Figures

Figures **Pages**

List of Tables

Introduction

This paper is about Neural Networks in general and Recurrent Neural Network in particular. Neural Networks are networks of biological neurons. This definition is the traditional one, where a Neural Network was considered as a simple calculator. However, modern usage of the term refers to they are networks of Artificial neurons, which are simple to handle. On the other hand, Recurrent Neural Network consists of the graph $G = (V, E)$ and a family of formal neurons $(X_i, Y_i, \sigma_i, s_i)$, each associated to one of the Vertices $i \in V$.

The general interest to study Neural Networks in general and Recurrent Neural Network in particular is that their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques.

This paper contains three chapters. The first chapter is about preliminary concepts, and results. It contains three sections. The first section is about Biological and Artificial Neural Networks. This includes Biological Neural Networks, Artificial Neural Networks, Difference and Similarity between Biological and Artificial Neural Networks. The second section is about perceptron. This includes Formal Neuron, Affine Separation, Separation of Finite Sets and perceptron Learning Algorithm. The third section is about Feed Forward Neural Network. This includes Structure of Feed Forward Neural Network, Feed forward Neural Network, k-layer Feed Forward Neural Network, k-layer σ-perceptrons and Realization by multilayer perceptrons. The second chapter is about Recurrent Neural Network. This part introduces about Finite Automata, Structure and Convergence of Recurrent Neural Network, Asynchronous update, Transient Length and Attractivity. The last chapter is about Applications of Neural Networks.

CHAPTER ONE: PRELIMINARY CONCEPTS

1. BIOLOGICAL AND ARTIFICIAL NEURAL NETWORKS

1.1 Biological Neural Networks

Definition 1.1: Biological Neural Network is a network of biological neurons, and Biological neuron (nerve cell) is a functional unit of the nervous system.

Here, these neurons receive inputs and provide outputs. **For instance,** the neural network in human being is a biological neural network.

Consider the figure shown below:

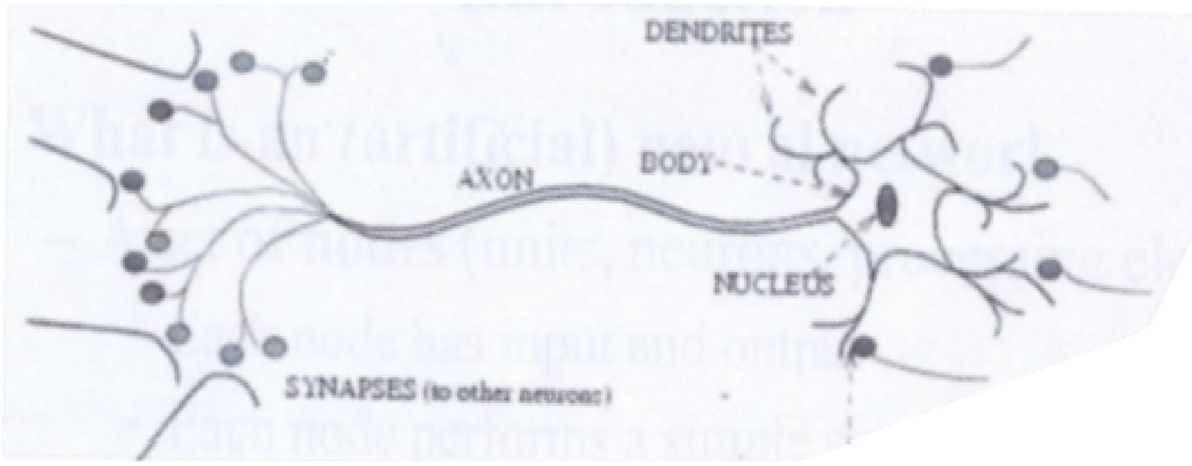

Fig 1.1: The Biological Neural Activity.

- Each neuron structurally has a cell body, an axon, and many dendrites.
    - Can be in one of the two states: firing or rest.
    - Neuron fires if the total incoming stimulus exceeds the threshold.
- **Synapse**: thin gap between axon of one neuron and dendrites of another.
    - Signal exchange (place where neurons communicate).
    - Synaptic strength (efficiency).

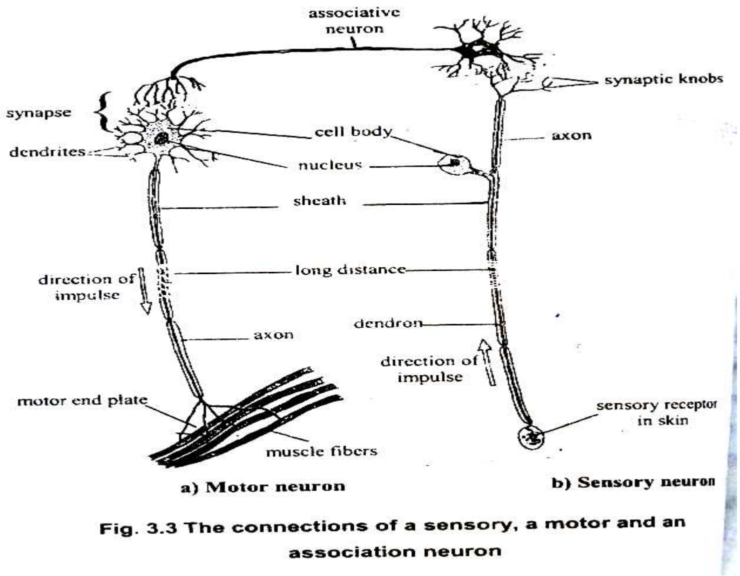

Fig1.2: Biological neural network in human being.

The cell body has a large nucleus embedded in dense granular cytoplasm and is variable in shape. A number of short cytoplasmic strands called **Dendrons** arise from the cell body and branch at the end in to thread – like extensions called **dendrites**. **Dendrites** are parts of the neuron that receive messages from nearby neurons.

The axon is a long cytoplasmic fiber which extends from the cell body and ends in terminal branches as can be seen in the above figure.

1.2 Artificial Neural Networks (ANNs)

Definition 1.2: Artificial Neural Network is a network of artificial neurons**.**

Artificial neuron is a device with many inputs and one output.

In other words, ANN is a set of nodes (units, neurons, processing elements)

- Each node has input and output.
- Each node performs a simple computation by its node function.

- Weighted connection between nodes.
- Connectivity gives the structure/architecture of the net.
- What can be computed by a neural network is primarily determined by the connections and their weights.
- A very much simplified version of networks of neurons is found in animal nerve systems.

1.3 Difference and Similarity between ANNs and BioNNs

DIFFERENCE

ANNs	**BioNNs**
- Nodes	- cell body
- Input	- signal from other neurons
- Output	- firing frequency
- Node function	- firing mechanism
- Connections	- synapses
- Connection strength	- Synaptic strength

Basic similarity between ANNs and BioNNs is that ANNs process information by the way of biological NNs.

Input/output values can be Binary {0, 1} Or Bipolar {-1, 1}.

1.4 Network Layers

The commonest type of artificial neural network consists of three groups, or layers or units: a layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output" units. See the figure below,

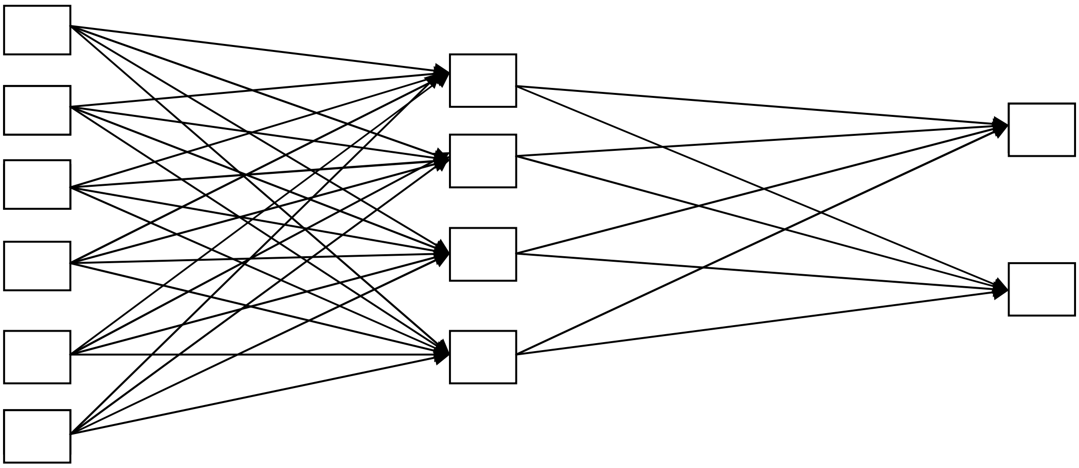

Figure 1.3: an example of a simple feed forward artificial neural network.

2. PERCEPTRON

This section introduces perceptron as a simple model of nerve cell.

Contents of this section are:

- Formal Neurons
- Affine Separation
- Separation of Finite Sets
- Perceptron Learning Algorithm

2.1 Formal Neurons

The structure and dynamics of biological neuron is very complex. Thus, in order to have simple and easily manageable artificial neurons, it is necessary to comprehend the essential functional characteristics of a neuron in a very simplified form. These neurons are known as **Formal Neurons**.

Definition 1.3: For integer $n \geq 1$, the symbol

$$\mathbb{R}^n = \{x | x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} ; \ x_i \in \mathbb{R}, i = 1 \ \cdots n\} .$$

- $\mathbb{R}^{N \times m}$ denotes the set of all $N \times m$ matrices.

Definition 1.4: A Formal Neuron is a data quadruple (X, Y, σ, S), where $X \subseteq \mathbb{R}^n$ for some positive integer 'n', $Y \subseteq \mathbb{R}$ and S and σ are mappings, $S: X \rightarrow \mathbb{R}$ and $\sigma: \mathbb{R} \rightarrow Y$, respectively. We call 'X' the input value set and 'Y' the output value set. The mapping 'S' is called an **activation function**, and σ is called the **output map**. However, sometimes a formal neuron will be identified with its **transfer function**, which is the composed input to output map $f = \sigma o s: X \rightarrow Y$.

Examples:

1. $f(x) = sat\ (-x + 0.5)$ (Transfer function)
2. $g(x) = sat\ (x_1 + x_2 - 1.5)$ (Transfer function) are Formal Neurons.

In 1943, warren McCulloch and Walter Pitts developed a simple but fundamental model, which has the capacity to realize the elementary Logical Functions such as NOT, AND and OR. In particular, all finite Boolean functions can be realized by using appropriate networks of such neurons.

From a biological point of view, the input signals can be viewed to stem from receptors of connected neurons.

Then the signals are modified at the **synapses** and condensed in to a single signal at the **soma**. If the quantity of these signals surpasses a certain threshold,θ, then the neuron fires. The simplest way to model such a neuron therefore uses $Y = \{0, 1\}$ as its output value set. The output is '1' if the neuron fires and '0' otherwise.

Definition: A formal neuron is a neuron whose output is 1 if it fires and 0 otherwise.

The affine linear map $S(x) = S(x_1, \cdots x_n) = \sum_{i=1}^{n} x_i w_i - \theta$ is an activation function. The weights $w_i \in \mathbb{R}^n$ for $i = 1, \cdots, n$ model the influence of the synapses on the signal x_i , $\theta \in \mathbb{R}$ represents the threshold and $x \in \mathbb{R}^n$.

Definition 1.5: Threshold is the smallest detectible sensation.

Examples: External Factors such as Cold, Hot, Being hungry, being thirsty etc are some examples of threshold. They force us to give the corresponding responses. Here, the threshold 'θ' is a determinant factor whether or not the neuron to fire. That is, if $\sum_{i=1}^{n} x_i w_i - \theta \geq 0$, then the neuron fires and not otherwise. It also affects both the output and the synapses.

Definition 1.6: A Formal Neuron that transmits the signal 1 if $\sum_{i=1}^{n} x_i w_i \geq \theta$ and 0 otherwise is called **McCulloch – Pitts or perceptrons**.

Now, let us define a function called the **Heaviside function sat** as follows:

Heaviside function sat: $\mathbb{R} \longrightarrow \{0, 1\}$, which is defined by

$$Sat\ (\mathbb{Z}) = \begin{cases} 1 & if\ \mathbb{Z} \geq 0 \\ 0 & if\ \mathbb{Z} < 0 \end{cases} \quad , \quad \text{here } \mathbb{Z} = \sum_{i=1}^{n} x_i w_i - \theta$$

This function is used as an output map because output map is $\sigma: \mathbb{R} \rightarrow Y = \{0,1\}$. Hence, the Heaviside function sat is a formal neuron as it is a data quadruple (X, Y, σ, S).

Since this formal neuron transmits the signal '1' if and only if $\sum_{i=1}^{n} x_i w_i \geq \theta$ and $'0'$ otherwise. Therefore, it is called **McCulloch – Pitts or perceptrons**. Often, the output map is required to be a **sigmoid function.**

Note: Heaviside function sat is an example of Formal Neuron.

Definition 1.7: Sigmoid function is a function with continuous value set $[0, 1]$ and a monotone function, $\sigma: \mathbb{R} \rightarrow [0, 1]$ with $\lim_{\mathbb{Z} \to -\infty} \sigma(\mathbb{Z}) = 0$ and $\lim_{\mathbb{Z} \to \infty} \sigma(\mathbb{Z}) = 1$.

Common examples for sigmoid function: 1. Heaviside function sat.

2. Fermi function.

1. **Solution**

Since Heaviside function sat: $\mathbb{R} \rightarrow \{0, 1\}$ defined by

$$Sat\ (\mathbb{Z}) = \begin{cases} 1 & if\ \mathbb{Z} \geq 0 \\ 0 & if\ \mathbb{Z} < 0 \end{cases} \qquad (*)$$

Definition (*) shows:

i) It is a function with continuous value set $[0, 1]$. That is, it assumes either '0' or '1' (members of $[0, 1]$).

ii) $\lim_{\mathbb{Z} \to -\infty} \sigma(\mathbb{Z}) = 0$ and $\lim_{\mathbb{Z} \to \infty} \sigma(\mathbb{Z}) = 1$. That is, it is a monotone function. **Hence**, it is a sigmoid function. ◊

2. Fermi function: $\sigma(\mathbb{Z}) = \frac{1}{1+e^{-\mathbb{Z}}}$

Solution

I. $\sigma(\mathbb{Z})$ assumes values between $[0, 1]$.

II. $\lim_{\mathbb{Z} \to -\infty} \sigma(\mathbb{Z}) = 0$ and $\lim_{\mathbb{Z} \to \infty} \sigma(\mathbb{Z}) = 1$.

Hence, it is a sigmoid function. ◊

In summary we arrive at the following definition:

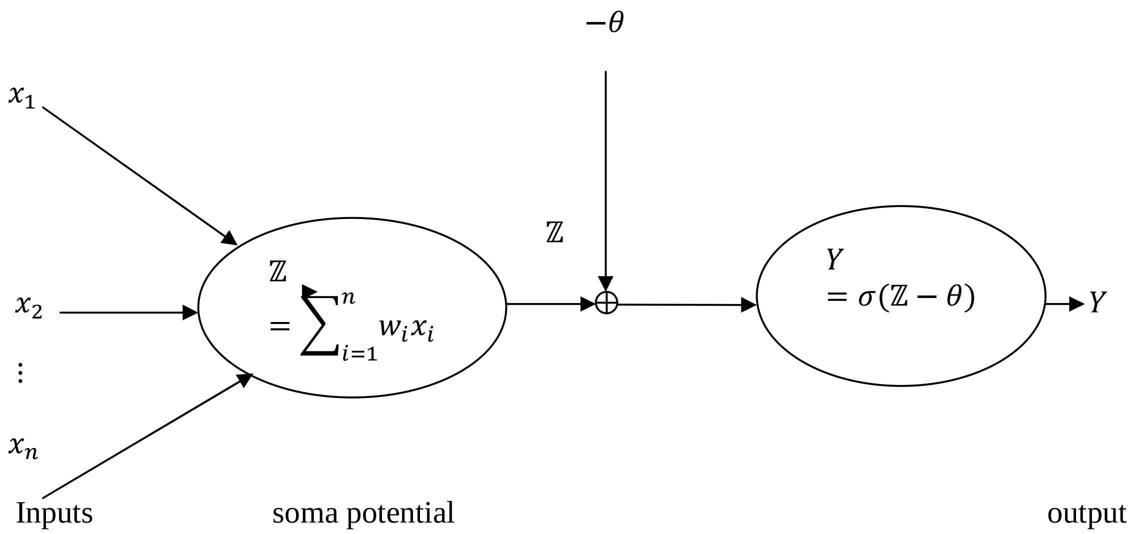

Fig 1.4: Representation of a Formal Neuron by an activation function**.**

Note: For the perceptron, the Heaviside function sat is used as the output function σ.

Definition 1.8: Let $S(x) = S(x_1, \dots, x_n) = \sum_{i=1}^{n} w_i x_i - \theta, w_i \in \mathbb{R}^n, \theta \in \mathbb{R}$.

Then the Formal neuron (X, Y, σ, S) is a σ- perceptron. If σ is equivalent to sat, then it is simply called a **perceptron or McCulloch – Pitts neuron**.

Definition 1.9: Boolean Function is a function whose range is in $\{0, 1\}$.

Examples: Logical Functions such as AND, NOT and OR are Boolean Functions.

Then we put $Y = \{0, 1\}$. Now we first focus on McCulloch – pitts neurons with $X = \{0,1\}^n$. If only binary inputs and outputs are used, then a perceptron's transfer function is a **Boolean function or switching function** $f = \{0,1\}^n \longrightarrow \{0,1\}$ of the form $f(x) = f(x_1, \cdots, x_n) = sat(\sum_{i=1}^{n} w_i x_i - \theta) = sat(\langle w, x\rangle - \theta)$ with fixed weights $w = (w_1, \cdots w_n)T,\ w \in \mathbb{R}^n$ and a fixed threshold $\theta \in \mathbb{R}$. Various switching functions of the form $f = \{0,1\}^n \longrightarrow \{0,1\}$ can be realized by a perceptron. That is, we can find a perceptron whose transfer function equals with 'f'.

Realizing Logical Functions such as NOT, AND

i) Let $f = NOT, f: \{0, 1\} \rightarrow \{0, 1\}$

Table 1.1: Realization of Boolean Function: NOT,

Input	**Desired output**
x	$f(x)$ =negationof x
0	1
1	0

The table is completed by the rule of negation.Then f can be represented by a perceptron for example $f(x) = sat\ (-x + 0.5)$ is a transfer function representing 'f'.

ii. $Let\ f = AND, f: \{0, 1\}^2 \rightarrow \{0, 1\}$

Table 1. 2: Realization of Boolean Function: AND

Input		**Desired output**
$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{f(x) = x_1 \wedge x_2}$
0	0	0
0	1	0
1	0	0
1	1	1

The table is completed by the rule of conjunction.

Then $f(x) = sat\ (x_1 + x_2 - 1.5)$ is an appropriate perceptron representing f.

Now since any Boolean function can be written in disjunctive normal form using only NOT, AND and OR operations. This implies that all Boolean functions can be represented by a **suitable network** consisting of McCulloch – Pitt's neurons or perceptrons. This does not infer, however, that all switching functions can actually be realized by a single perceptron. A counter example is the "exclusive OR" function cannot be represented by a single perceptron.

The Shortcoming of Perceptrons

Lemma 1.1: There is no perceptron that represents the XOR- function.

Proof: Assume that for all $(x_1\ , x_2) \in \{o, 1\}^2$,$x_1 \oplus x_2 = (w_1 x_1 + w_2 x_2 - \theta)$.

i) $0 \oplus 0 = 0 \Rightarrow sat(-\theta) = 0 \Rightarrow -\theta < 0$ (by definition of Heaviside function)

ii) $0 \oplus 1 = 1 \Rightarrow sat(w_2 - \theta) = 1 \Rightarrow w_2 - \theta \geq 0$ (by definition of Heaviside function)

iii) $1 \oplus 0 = 1 \Rightarrow sat(w_1 - \theta\ ) = 1\ \Rightarrow w_1 - \theta\ \geq 0$ (by the same reason)

iv) $1 \oplus 1 = 0 \Rightarrow sat(w_1 \ + w_2 \ - \theta) = 0 \ \Rightarrow w_1 \ + w_2 \ - \theta \ < 0$.(by the same reason)

Now adding (i) and (iv) ⇒ $w_1 \ + w_2 \ \ - 2\theta < \ 0$

Again adding (ii) and (iii) ⇒ $w_1 \ \ + \ w_2 \ - \ 2\theta \ \geq \ 0$, which is a contradiction.

Hence, the proof of the lemma. ◊

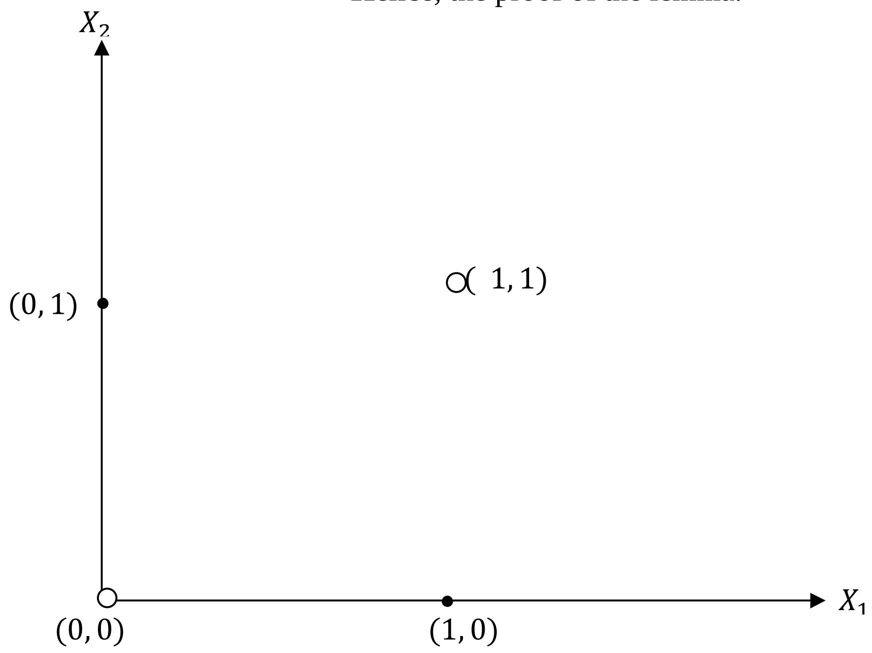

Fig 1.5: The geometrical interpretation of the XOR- problem.

Remark 1.1: The realization of Boolean functions by a perceptron is closely linked to the affine separation of sets.

Definition 1.10: *A* Hyperplane is any set of the form

$$\{x: \langle x, N\rangle = d, d \text{ is a constant}, N \neq 0 \text{ is a normal line } \}.$$

- In $\mathbb{R}^2$, a hyperplane is a line $ax + by \ = d$.
- In $\mathbb{R}^3$, a hyperplane is an ordinary plane $ax + by + zc \ = \ d$.

Example: $x_1 + x_2 - 1 = 0$ in $\mathbb{R}^2$ is a hyperplane.

Definition 1.11: A hyperplane H defines **two closed half-spaces:**

$H_+ = \{x \in \mathbb{R}^n : \langle w, x\rangle \geq \theta\}$ and $H_- = \{y \in \mathbb{R}^n : \langle w, y\rangle \leq \theta\}$ and **two open half-spaces.**

$H_+ = \{x \in \mathbb{R}^n : \langle w, x\rangle > \theta\}$ and $H_- = \{y \in \mathbb{R}^n : \langle w, y\rangle < \theta\}$.

2.2 Affine Separation of Finite Sets

Definition 1.12: A set $A \subset \mathbb{R}^n$ is called **affinely separable** from a set $B \subset \mathbb{R}^n$ if there exists $(w, \theta) \in \mathbb{R}^{n+1}$ with

$$\langle w, x\rangle - \theta \begin{cases} \geq 0 \; for \; x \in A \\ < 0 \; for \; x \in B \end{cases} \tag{1.1}$$

Then $H = \{x \in \mathbb{R}^n : \langle w, x\rangle = \theta\}$ is called a **separating hyperplane**.

If the inequalities in (1.1) above are both **strict,** we say that A and B are **strictly affinely separable** (from each other).

If $\theta = 0$, we say that A is linearly separable from B, or A and B are strictly linearly separable.

Example: The sets $A = \{(1,1)\}$ and $B = \{(0,0), (0,1), (1,0)\}$ are affinely separable.Since $\exists(1,1,1.3) \in \mathbb{R}^3$ with

$$\langle w, x\rangle - \theta \begin{cases} \geq 0 \; for \; all \; x \in A \\ < 0 \quad for \; all \; x \in B \end{cases}$$

Now we formulate a sufficient and necessary condition for $\{0, 1\}$ valued- functions to be realizable by McCulloch – Pitt's neurons.

Theorem 1.2: Let $X \subseteq \mathbb{R}^n$ (be an arbitrary subset of $\mathbb{R}^n$). Then a **function** $f: X \rightarrow \{0, 1\}$ can be represented by a perceptron if and only if X_+ is affinely separable from X_- where $X_+ = f^{-1}(1) \subseteq X$ and $X_- = f^{-1}(0) \subseteq X$.

Proof: (**only the if part**)

($\Rightarrow$) Suppose $f: X \to \{0, 1\}$ can be represented by a perceptron.

Claim: X_+ is affinely separable from X_- .

By the supposition, we have $f(x) = sat\ (< w, x > -\theta)$, where$(w, \theta) \in \mathbb{R}^{n+1}$. But, this holds if and only if

$$\langle w, x\rangle - \theta \begin{cases} \geq 0 \ for\ x \in X_+ \\ < o\ for\ x \in X_- \end{cases}$$

$\Leftrightarrow$ X_+ is affinely separable from X_-. (By definition)

Hence, showing the claim. ◊

Example: Let $f = OR, f: \{0, 1\}^2 \to \{0,1\}$

Table 1. 3: Realization of Boolean Function: OR

Input		**Desired output**
$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{f(x) = x_1 \vee x_2}$
0	0	0
0	1	1
1	0	1
1	1	1

The table is completed by the rule of disjunction.

The Perceptron $f(x) = sat\ (x_1 + x_2 - 0.5)$ represents 'f'.

Now let $X_+ = \{(1,1), (0,1), (1,0)\}$

$X_- = \{(0,0)\}$ Where $X_+ = f^{-1}(1) \subseteq \{0,1\}^2$, $X_- = f^{-1}(0) \subseteq \{0,1\}^2$.

Since 'f' can be represented by a perceptron, $f(x) = f(x_1 + x_2 - 0.5)$.

Hence, X_+ is affinely separable from X_- (by theorem 2.2). However, there are difficulties in separating arbitrary finite disjoint sets. Therefore, the degree of this difficulty in separating arbitrary finite disjoint sets is measured by means of the so-called **capacity**.

Definitions 1.13: A) A real number c is called the **supremum** of a finite set E i) If c is an upper bound of E ii) if c is not an upper bound of E, then $c \leq m$ for any upper bound m of E.If c is the supremum of E, then we write $c = \sup(E)$.

B) A real number d is called the **infimum** of E i) If d is a lower bound of E ii) If d is not a lower bound of E, then $d \geq n$ for any lower bound n of E.If n is the infimum of E, then we write $n = \inf(E)$.

Definition 1.14: Let $A, B \subseteq \mathbb{R}^n$. Then the sets A and B are said to be **strictly affinely separable** if and only if

$$\text{Sup } \{\langle w, x\rangle : x \in B\} < \inf\{\langle w, y\rangle : y \in A\}, w \in \mathbb{R}^n.$$

Example: Consider the sets $A = \{(0,1), (1,0), (1,1)\}$, $B = \{(0,0)\}$.Then they are strictly affinely separable.

Solution

Since for $W = (1,1) \in \mathbb{R}^2$, Sup $\{\langle w, x\rangle : x \in B\} = \text{Sup}\{0\} = 0$ and Inf $\{\langle w, y\rangle : y \in A\} = \inf\{1,2\} = 1$.Which implies Sup $\{\langle w, x\rangle : x \in B\} < \inf\{\langle w, y\rangle : y \in A\}$.

Definition 1.15: A set $A \subset \mathbb{R}^n$ is called a **convex set** if for any $x, y \in A$, the complete line segment between x and y is also contained in A.

That is, $x, y \in A$ and $0 \leq \lambda \leq 1 \Rightarrow \lambda x + (1 - \lambda) y \in A$.

Example: The Set $A = \{(x, 0) : x \in \mathbb{R}\} \cup \{(0,1)\}$ is a convex Set.

Definitions 1.16:

1. A set is **compact** if and only if it is bounded and closed.

Example: $A = [1, 2]$ is a compact set since it is bounded and closed.

2. A point $p \in A$ is an interior point of set A if the open interval containing p belongs to A.

Example: $A = (0,1)$ is an open set.

3: A set $A \subset \mathbb{R}^n$ is said to be **open** if each of its points is an interior point.

4: A set $A \subset \mathbb{R}^n$ is said to be a **closed** set if its complement is an open set in $\mathbb{R}^n$.

Example: $A = [4,8]$ is a closed set. Since $\mathbb{R}^2 - A$ is open in $\mathbb{R}^2$.

Let X be a non – empty set. A collection τ of subsets of X is a topology on X if and only if τ satisfies the following axioms:

i) X and ϕ belong to τ.

ii) The union of any collection of sets in τ belongs to τ.

iii) The intersection of any two sets in τ belongs to τ.

The pair (X,τ) is a topological space.

5: Let X be a topological space.

- A collection ℓ is said to be **an open covering** of X if its elements are open subsets of X . A space X is said to be **a compact space** if every open covering ℓ of X contains a finite sub collection that also covers X.

Remark 1.2: Cauchy – Schwarz inequality: $|\langle u, v\rangle| \leq \|u\|\|v\|$.

In order to decide whether two given finite sets are affinely separable, the concept of convexity plays a decisive role.

The following theorem gives a sufficient condition for separability of convex sets.

Theorem 1.3: Let A and B be two non- empty, disjoint convex subsets of $\mathbb{R}^n$.

a. If B is **open**, then A is affinely separable from B.

b. If one of the sets is **closed** and the other is **compact,** then they are strictly affinely separable.

Proof

a. proof by contradiction

Assume that A is not affinely separable from B. Then for a given vector $(w, \theta) \in \mathbb{R}^{n+1}$,

$$\langle w, x\rangle - \theta \begin{cases} \geq 0 \ for\ x \in A \\ < 0 \ for\ x \in B \end{cases}$$ may not hold for all x.

$\Rightarrow$ There exist $x \in B$ such that $\langle w, x\rangle - \theta \geq 0$

$\Rightarrow B$ Contains boundary points

$\Rightarrow B$ is not open (we arrived at a contradiction).

Hence, the proof of the Theorem2.3 (part a). ◊

b. **Claim 1**: let $A \subset \mathbb{R}^n$ be non-empty, closed, convex set. Then A posse's exactly one element of minimal norm. i.e. there is exactly one $x_o \in A$ such that

$$\|x_o\| = \min_{x \in A}\|x\|.$$

To show this

Let $\{x_n\}_{n=1}^{\infty}$ be a sequence in A with $\|x_n\| \longrightarrow \inf_{x \in A} \|x\| = \gamma$

Now by parallelogram law equation,

$$\frac{1}{4}\|x_n + x_m\|^2 = \frac{1}{2}\|x_n\|^2 + \frac{1}{2}\|x_m\|^2 - \frac{1}{4}\|x_n - x_m\|^2$$

By the minimal property of γ, we have $\|x_n\|^2 \geq \gamma^2, \|x_m\|^2 \geq \gamma^2$. Once again, since A is a convex set, $\frac{1}{2}(x_n + x_m) \in A$.

$$\Rightarrow \frac{1}{4}\|x_n + x_m\|^2 \geq \gamma^2$$

Now for each $\varepsilon > 0$ and sufficiently large $n, m \in \mathbb{N}$,

We have $\|x_n\|^2 \leq \gamma^2 + \varepsilon$, $\|x_m\|^2 \leq \gamma^2 + \varepsilon$, $-\frac{1}{4}\|x_n + x_m\|^2 \leq -\gamma^2$

Then $\frac{1}{4}\left\|x_n - x_m\right\|^2 \leq \frac{1}{2}(\gamma^2 + \varepsilon + \gamma^2 + \varepsilon) - \gamma^2 = \varepsilon$

$\Rightarrow \{x_n\}_{n=1}^{\infty}$ is a Cauchy sequence and therefore it is convergent. i.e. it has a limit point $x_o \in \mathbb{R}^n$.

Since A is a closed set, we have $x_o \in A$. So, we have

$$\gamma = \lim_{n \to \infty} \left\|x_n\right\| = \left\|x_o\right\| = \inf\left\|x\right\| = \min_{x \in A} \left\|x\right\|$$

$$\therefore \left\|x_o\right\| = \min_{x \in A} \left\|x\right\|$$ ◊

Now we prove uniqueness of this minimal norm.

Let x_o , $y_o \in A$ with $\left\|x_o\right\| = \left\|y_o\right\| = \gamma$

Since A is a convex set, $\frac{1}{2}(x_o + y_o) \in A$.

Hence, $\gamma^2 \leq \frac{1}{4}\left\|x_o + y_o\right\|^2 = \frac{1}{2}\left\|x_o\right\|^2 + \frac{1}{2}\left\|y_o\right\|^2 - \frac{1}{4}\left\|x_o - y_o\right\|^2$

$$= \frac{1}{2}\gamma^2 + \frac{1}{2}\gamma^2 - \frac{1}{4}\left\|x_o - y_o\right\|^2 = \gamma^2 - \frac{1}{4}\left\|x_o - y_o\right\|^2$$

$$\Rightarrow \frac{1}{4}\left\|x_o - y_o\right\|^2 \leq 0$$

$$\Rightarrow \left\|x_o - y_o\right\|^2 \leq 0$$

$\Rightarrow \left\|x_o - y_o\right\|^2 = 0$ (Since norm cannot be negative).

$$\Rightarrow x_o = y_o$$

Hence, the element of minimal norm exists and it is unique. ◊

Lemma 1.4: - let $(A \neq \emptyset) \subseteq \mathbb{R}^n$ be a closed, convex set and $0 \notin A$. Then $\{0\}$ and A can be strictly affinely separable. Moreover, if $a \in A$ is the element of minimal norm, then

$$\langle w, x\rangle \geq \left\|a\right\|^2 > 0 \text{ for all } x \in A.$$

Proof:-

By **claim1**, the element of minimal norm exists. Let this element be $a \in A$. Since by the supposition $0 \notin A$ implies $a \neq o$. Then since A is a convex set, we have $\lambda x + (1-\lambda)a = \lambda(x-a) + a \in A$ for $\lambda \in [0,1]$, so by using the minimal property of a, again we have

$$0 \le \|a\|^2 \le \|a + \lambda(x-a)\|^2 = \langle a + \lambda(x-a), a + \lambda(x-a) \rangle$$

$$= \langle a, a + \lambda(x-a)\rangle + \langle \lambda(x-a), a + \lambda(x-a)\rangle$$

$$= \langle a, a \rangle + \langle a, \lambda(x-a) \rangle + \langle \lambda(x-a), a \rangle + \langle \lambda(x-a), \lambda(x-a)\rangle$$

$$= \|a\|\|a\| + \langle a, \lambda x - \lambda a \rangle + \langle \lambda x - \lambda a, a \rangle + \lambda\bar{\lambda}\langle x-a, x-a \rangle$$

$$= \|a\|^2 + \langle a, \lambda x \rangle + \langle a, -\lambda a \rangle + \langle \lambda x, a \rangle + \langle -\lambda a, a \rangle + \lambda\lambda[\|x-a\|\|x-a\|]$$

$$= \|a\|^2 + \bar{\lambda}\langle a, x \rangle - \bar{\lambda}\langle a, a \rangle + \lambda\langle x, a \rangle - \lambda\langle a, a \rangle + \lambda^2\|x-a\|^2$$

$$= \|a\|^2 + \lambda\langle a, x \rangle - \lambda\langle a, a \rangle + \lambda\langle x, a \rangle - \lambda\langle a, a \rangle + \lambda^2\|x-a\|^2$$

$$= \|a\|^2 + \lambda\langle a, x \rangle + \lambda\langle a, x \rangle - \lambda\langle a, a \rangle - \lambda\langle a, a \rangle + \lambda^2\|x-a\|^2$$

$$= \|a\|^2 + 2\lambda\langle a, x \rangle - 2\lambda\langle a, a \rangle + \lambda^2\|x-a\|^2$$

$$\Rightarrow \|a\|^2 \le \|a\|^2 + 2\lambda\langle a, x \rangle - 2\lambda\langle a, a \rangle + \lambda^2\|x-a\|^2$$

$$\Rightarrow 0 \le 2\lambda\langle a, x \rangle - 2\lambda\langle a, a \rangle + \lambda^2\|x-a\|^2$$

$$\Rightarrow 0 \le 2\lambda\langle a, x \rangle - 2\lambda\|a\|^2 + \lambda^2\|x-a\|^2$$

$$\Rightarrow 2\lambda\|a\|^2 - \lambda^2\|x-a\|^2 \le 2\lambda\langle a, x \rangle$$

$$\Rightarrow \|a\|^2 - \tfrac{\lambda}{2}\|x-a\|^2 \le \langle a, x \rangle \quad \text{for } \lambda > 0$$

In particular, if $\lambda \to 0$, then we have $0 < \|a\|^2 \le \langle a, x \rangle$.

Now $\langle a, 0 \rangle = 0 < \|a\|^2 \le \langle a, x \rangle$ for all $x \in A$ this implies

$$0 = \sup_{y=0} \langle a, y \rangle < \inf_{x \in A} \langle a, x \rangle = \|a\|^2$$

Hence, the result. ◊

Now suppose $Z \notin A$. (***)

Claim 2: $\{Z\}$ and A can be strictly affinely separable.

Proof:

Let $Z \notin A, C = A - \{Z\}$. Clearly C is a closed set. Since $A - \{Z\}$ is open in $\mathbb{R}^n$.

Here, $C = A - \{Z\} = \{a - z : a \in A, z \in \{Z\}\}$.

Let $\propto$ be the element of minimal norm of C. Then there exists $a \in A$ such that

$$\propto := a - z$$

Obviously, $0 \notin C$ otherwise $Z \in A$, this is a contradiction to (***). We can apply the above theorem with $\propto = a - z$ and $x = a^` - z$.

Then we have $\langle \propto, x \rangle = \langle \propto, a^` - z \rangle \geq \| \propto \|^2 > 0$ for $x \in A$. That is, this yields

$$\langle \propto, a^` \rangle \geq \langle \propto, z \rangle + \| \propto \|^2 > \langle \propto, z \rangle \text{ for all } \propto \in A.$$

This implies

$$\langle \propto, a^` \rangle > \langle \propto, z \rangle \text{ for all } a^` \in A$$

$$\Rightarrow \inf_{a^` \in A} \langle \propto, a^` \rangle > \langle \propto, z \rangle = \sup_{y = z} \langle \propto, y \rangle$$

Hence, $\{z\}$ and A are strictly affinely separable. (By definition above) ◊

From the above we have $\langle a, x \rangle - \|a\|^2 > 0$ for all $x \in A$.

Thus, put $w = a(\neq 0) \in \mathbb{R}^n$ and $\theta = \|a\|^2$. Then there exist $(w, \theta) \in \mathbb{R}^{n+1}$ such that

$$\langle w, x \rangle - \theta \begin{cases} \geq 0 \; for \;\; all & x \in A \\ < 0 \; for \;\; all & x \in \{0\} \end{cases}$$

Hence, $\{0\}$ and A are strictly affinely separable. ◊

Now assume the set B is closed and A is compact such that $A \cap B \neq \emptyset$.

Claim 3: A and B are strictly affinely separable.

Let $C = B - A$.

Since $A \cap B \neq \emptyset$ and $B - A$ is open in $\mathbb{R}^n$, C is a closed set.

Further, $B \neq A, 0 \notin C$. Therefore, by claim 2 A and B are strictly affinely separable. ◊

Remark 1.3: For arbitrary set A, it is useful to consider their convex supersets, the smallest of which is called the convex hull of A.

Definition 1.17: Let $A \subset \mathbb{R}^n$ be an arbitrary set. The convex hull $\text{Conv}(A)$ of A is defined as the intersection of all convex subsets of $\mathbb{R}^n$ that contain A. Thus, $\text{Conv}(A)$ is the smallest convex subsets of $\mathbb{R}^n$ that contain A.

That is $\text{Conv}(A) = \left\{\sum_{i=1}^{k} \lambda_i x_i : k \in \mathbb{N}, x_i \in A, \lambda_i \geq 0, \sum_{i=1}^{k} \lambda_i = 1\right\}$

, where $\text{Conv}(A)$ is the convex hull of A

Remark 1.4: 1. Convex hull of an open set in $\mathbb{R}^n$ is again open.

2. Convex hull of a compact set in $\mathbb{R}^n$ is again compact.

3. Convex hull of a closed set in $\mathbb{R}^n$ is not necessarily closed.

4. Closure of a convex set in $\mathbb{R}^n$ is again convex.

Definition 1.18: closure of a set:

Let X be a topological space. Let A be a subset of X. Then the closure, $\overline{A}$ of A is the intersection of all closed subsets of A.

Lemma 1.5: The sets $H_+ = \{x \in \mathbb{R}^n : \langle w, x\rangle \geq \theta\}$ and $H_- = \{ y \in \mathbb{R}^n : \langle w, y\rangle < \theta\}$ are convex sets.

Claim 1: $H_+ = \{x \in \mathbb{R}^n : \langle w, x\rangle \geq \theta\}$ is a convex set. Let $x, y \in H_+$. Then

<u>**WTS**</u>: $\langle w, \lambda x + (1 - \lambda)y \rangle \in H_+$

$= \langle w, \lambda x + y - \lambda y \rangle = \langle w, \lambda x - \lambda y + y \rangle$

$= \langle w, \lambda x\rangle + \langle w, -\lambda y\rangle + \langle w, y \rangle$ Since $\langle\ \rangle$ is liner with respect to the 1st variable

$= \overline{\lambda}\langle w, x \rangle - \overline{\lambda}\langle w, y \rangle + \langle w, y \rangle$

$= \overline{\lambda}\ \theta - \overline{\lambda}\theta + \ \theta$ Since by taking the minimum values.

$\geq \theta$

Hence, $\langle w, \lambda x\ +\ (1\ -\ \lambda)y\ \rangle \in H_+$

$\therefore H_+\ =\ \{x \in \mathbb{R}^n : \langle\ w, x\ \rangle \geq \theta\}$ is a convex set ◊

Similarly, $H_- = \{\ y \in \mathbb{R}^n : \langle w, y\rangle < \theta\}$ is a convex set. ◊

The following corollary gives a sufficient and necessary criterion for separability of non-convex sets.

Corollary 1.6:

1. Let $A, B \subset\ \mathbb{R}^n$ be non – empty, and let B be open. Then, A is affinely separable from B if and only if $conv(A) \cap conv(B) =\ \phi$.
2. Let $A, B \subset\ \mathbb{R}^n$ be non – empty, with A closed and B compact sets.

   Let $\overline{cov(A)}$ denote the closure of $conv(A)$. Then A and B are strictly affinely separable if and only if $\overline{cov(A)} \cap Conv\ (B) =\ \emptyset$.

Proof:

1. $(\Rightarrow)$ Suppose A is affinely separable from B.

   Claim: $conv(A)\ \cap conv\ (B) =\ \emptyset$

   Since by supposition A is affinely separable from B, we have the half-spaces $H_+\ =\ \{x \in \mathbb{R}^n : \langle\ w, x\quad \rangle \geq \theta\}$ and $H_- = \{\ y \in \mathbb{R}^n : \langle w, y\rangle < \theta\}$ with $A \subset H_+$ and $B \subset H_-$ respectively.

Since H_+ and H_- are convex sets (Lemma 2.3), we have $conv(A) \subset H_+$ and $conv\ (B) \subset\ H_-$

$\Rightarrow Conv(A)$ and $conv(B)$ Being disjoint

$\Rightarrow Conv(A)\ \cap Conv(B)\ = \emptyset$

Hence, the claim ◊

(⇐) Suppose $conv\ (A)\ \cap\ conv\ (B)\ = \emptyset$

Claim: A is affinely separable from B.

Since $conv(A) \cap conv(B) = \emptyset$ and $conv(A)$ and $conv(B)$ are convex sets. On the other hand, since B is an open set and by the remark 2.2.11, convex hull of an open set in $\mathbb{R}^n$ is again open. Then $conv(B)$ is open.

$\Rightarrow conv(A)$ and $conv(B)$ are affinely separable (by theorem 2.3 (a))

Since $conv\ (A)$ is the smallest convex set containing A and similarly, $conv(B)$ is the smallest convex set containing B. Then we have $A \subset\ conv(A)$ and $B \subset\ conv\ (B)$.

Since $Conv\ (A)$ is affinely separable from $conv(B)$.

$\Rightarrow A$ is affinely separable from B (since $A \subset\ conv(A)$ and $B \subset\ conv\ (B)$).

Hence, then claim ◊

2. (⇒) Suppose A and B are strictly affinely separable.

Claim:$\overline{cov(A)} \cap conv(B)\ =\ \phi$

By the assumption, we have a closed half- space H_+ with $A \subset H_+$ and an open half space H_- with $B \subset H_-$. Since A is a closed set, we have $conv(A)$ is a closed set and H_+ is a closed convex set containing $conv(A)$. That is, $conv(A) \subset H_+$. This is because of $conv(A)$ is the smallest. Since B is a compact set, we have $conv(B)$ is a compact set (by remark 2.4). H_- is also a compact set containing $conv(B)$. That is, $conv(B) \subset H_-$.

Now let $\overline{cov(A)}$ denote the closure of $conv(A)$ and $\overline{H_+}$ is closure of H_+ . Then $\overline{cov(A)} \subset \overline{H_+} = H_+$ and hence, $\overline{cov(A)} \cap conv(B)\ =\ \phi$

Hence, the claim ◊

(⇐) Suppose $\overline{cov(A)} \cap conv\ (B)\ =\ \phi$

Claim: A and B are strictly affinely separable.

Since $conv(B)$ is open, we have $\overline{cov(A)}$ is affinely separable from $conv(B)$(by theorem 2.3 (a) and by the assumption. Now by remark 2.4, the convex hull of a compact set is compact and the closure of a convex set is convex, we have $A \subset \overline{cov(A)}$ and $B \subset conv(B)$.

$\Rightarrow A$ is strictly affinely separable from B. ◊

Example: Consider $A = [0, 1]$, $B = (8, 12)$ in $\mathbb{R}^2$. Now B is an open set, since any $a \in B$ is an interior point of B. Hence, A is affinely separable from B (by theorem 2.3 (a).

Once again, A is a closed set since $\mathbb{R}^2 - A$ is open in $\mathbb{R}^2$. .

Let $B = [8, 12]$. Then B is compact since it is closed and bounded.

Hence, A and B are strictly affinely separable (by theorem 2.3 (b)).

The Limitation of Affine Separation

There exists no affine separation hyperplane $\langle w, s\rangle - \theta = 0$ that separates $\{(1,0), (0,1)\}$ from $\{(0,0), (1,1)\}$. i.e There exists no $(w, \theta) \in \mathbb{R}^{n+1}$ such that

$$\langle w, s\rangle - \theta \begin{cases} \geq 0 \; for \; s \in A \\ < 0 \; for \; s \in B \end{cases}$$

Where $A = \{(1,0), (0,1)\}$

$B = \{(0,0), (1,1)\}$

2.3 Separation of Finite Sets

Remark 1.5: Finite sets and their convex hulls are compact sets.

Consider the mapping $f: X \rightarrow \{\pm 1\}$, where $X = \{ x_1, \cdots, x_N\} \subset \mathbb{R}^n$.

Let $x \in X_{\pm} = f^{-1}(\pm 1)$

Suppose X_+ and X_- are both non-empty.

Let $Y_i = f(x_i)$ for $1 \leq i \leq N$

The sets X_+ and X_- are (strictly) affinely separable if and only if $Y_i\,(\langle w, y_i\rangle - \theta\,) > 0$ Where $1 \le i \le N$ and $w \in \mathbb{R}^2, \theta \in \mathbb{R}$.

Equivalently, $f(x) = sign(\,\langle w, y_i\rangle - \theta\,)$ for all $y_i \in X$.That is, the function 'f' is realizable by a perceptron with the modified Heaviside or sign function,

$$sign(\mathbb{Z}) = \begin{cases} 1\ if\ \mathbb{Z} \ge 0 \\ -1\ if\ \mathbb{Z} < 0 \end{cases}$$

Example: Consider the logical function AND, $f\colon \{-1, 1\}^{\,2} \rightarrow \{-1, 1\}$

Table 1.4:

Input		Desired output
$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$f(x) = \boldsymbol{x_1} \wedge \boldsymbol{x_2}$
-1	-1	-1
-1	1	-1
1	-1	-1
1	1	1

Hence, $f(x) = sign\,(\,\boldsymbol{x_1} + \boldsymbol{x_2} - 1)$ is a transfer function since it is the combination of input and output. Once again, $\boldsymbol{x_1} + \boldsymbol{x_2} - 1$ is equivalent to $\sum_{i=1}^{2} w_i x_i - \theta,\ x_i, w_i \in \mathbb{R}^2, \theta \in \mathbb{R}$.

$$\Rightarrow w_1x_1 + w_2x_2 - \theta = x_1 + x_2 - 1$$ and $w_1 = 1$, $w_2 = 1$, $\theta = 1$.

Let $X_+ = \{(1,1)\}$

$X_- = \{\,(-1,-1), (-1,1\,), (\,1,-1)\}$ *where* $X_+ = f^{-1}(1) \subseteq X$

$X_- = f^{-1}(1) \subseteq X$

Hence, X_+ and X_- are (strictly) affinely separable. ◊

2.4 Perceptron Learning Algorithms

If the finite sets, X_+ and X_- are to be separated, we are interested in finding a concrete separating hyperplane. A method of constructing such a hyperplane is given by the perceptron Learning Algorithm.

The strict affine separability of the sets, X_+ and X_- is assumed throughout this section.

PL problem: Two finite sets, $X_+, X_- \subset \mathbb{R}^n$ with $conv(X_+) \cap conv(X_-) = \phi$ shall be affinely separated by a perceptron.

In other words, find $(w, \theta) \in \mathbb{R}^{n+1}$ such that

$$sign(\langle w, x\rangle - \theta) = \begin{cases} 1\ if\ \mathbb{Z} \geq 0 \\ -1\ if\ \mathbb{Z} < 0 \end{cases} \quad \text{Where } \mathbb{Z} = \langle w, x\rangle - \theta$$

It is convenient to eliminate the threshold θ: Replace (x, w) by $(\hat{x}, \hat{w})$, where

$\hat{x} = \begin{bmatrix} x \\ 1 \end{bmatrix}$ and $\hat{w} = \begin{bmatrix} w \\ -\theta \end{bmatrix}$

Hence, the new input domain is $\mathbb{R}^n \times \{1\}$ and $sign(\langle \hat{w}, \hat{x}\rangle) = sign(\langle w, x\rangle)$

Instead of the affine separation of X_+ and X_- in $\mathbb{R}^n$, we now face the problem of linear separation of $\hat{X}^+$ and $\hat{X}^-$ in $\mathbb{R}^{n+1}$.

Let further m= n+1 and $\hat{X}^+ \cup \{-x : x \in \hat{X}^-\} \subset \mathbb{R}^n$

Reformulated PL problem: find a vector $\hat{w} \in \mathbb{R}^m$ with $\langle w, \boldsymbol{\xi}\rangle > 0$ for all $\boldsymbol{\xi} \in \hat{\boldsymbol{X}}$.

If $\hat{\boldsymbol{X}} = \{\boldsymbol{\xi_1} \ldots, \boldsymbol{\xi_N}\}$ define $A = [\boldsymbol{\xi_1} \quad \ldots \quad \boldsymbol{\xi_N}]^T \in \mathbb{R}^{N\times M}$.Then $A\hat{W} > 0$, where the inequality is to be understood component wise.

Definition 1.19: Let $\hat{x} \subset \mathbb{R}^m$ be a finite set. Then a sequence $\nu: \mathbb{N} \longrightarrow \hat{x}$ is a training sequence for $\hat{x}$ if for all $\boldsymbol{\xi} \in \hat{x}$ and all $k_o \in \mathbb{N}$ there is a $k \geq k_o$ with $\upsilon(k) = \boldsymbol{\xi}$. That is, each pattern $\boldsymbol{\xi} \in \hat{x}$ appears infinitely often in a training sequence ν for $\hat{x}$.

Procedure of perceptron Learning algorithm:-

Let $A = [\boldsymbol{\xi_1} \quad \ldots \quad \boldsymbol{\xi_N}]^T \in \mathbb{R}^{N\times M}$ be given, let $\nu: \mathbb{N} \longrightarrow \hat{x}$ be a training sequence for $\{\boldsymbol{\xi_1}, \cdots, \boldsymbol{\xi_N}\}$. And let $\varphi: \mathbb{N} \longrightarrow \mathbb{R}$ be a sequence with $0 < \inf \varphi(k) \leq \sup \varphi(k) < \infty$.

Step 0: Choose $w(0) \in \mathbb{R}^m$ and put $k = 0$.

Step 1: Set $w(k+1) = \begin{cases} w(k) \;\; if \;\; \upsilon(k)^T w(k) > 0 \\ w(k) + \varphi(k)\upsilon(k) \;\; if \;\; \upsilon(k)^T w(k) < 0 \end{cases}$

Step 2: If $Aw(k+1) > 0$ applies component wise, then an admissible separation has been found and the algorithm stops. Otherwise, augment k by one and go to step 1.

Example: - An affine separation of the sets $X_+ = \{(0,0),(0,1),(1,0),(-1,1)\}$ and $X_- = \{(-2,1),(-2,0),(-1,0),(-1,-1)\}$ is sought.

Solution

After eliminating the thresholds and reflecting the set X_- at the origin, we obtain the set,

$\hat{x}^+ = \begin{bmatrix} x \\ 1 \end{bmatrix}, \hat{x}^- = \begin{bmatrix} -x \\ -1 \end{bmatrix}$

$$\hat{x} = \hat{x}^+ \cup \{-x : x \in \hat{X}^-\}$$
$$= \{(0,0,1),(0,1,1),(1,0,1),(-1,1,1),(2,-1,-1),(2,0,-1),(1,0,-1),(1,1,-1)\}$$

Now a weight vector $w \in \mathbb{R}^3$ is sought with $wx > o$ for $x \in \hat{x}$.For this, the perceptron Learning Algorithm is started with the randomly chosen weight vector $w(0) = (0,2,1)$ and for simplicity, let$\varphi: \mathbb{N} \rightarrow \mathbb{R}$ with $\varphi(k) = 1$. The hyperplane $H = \{x \in \mathbb{R}^3 : \langle w(0), x\rangle = 0\}$, which is orthogonal to $w(0)$, separates the correctly classified points from the falsely classified points, $(2,-1,-1), (2,0,-1)$ and $(1,0,-1)$.

If a falsely classified point emerges in the training sequence $\nu: \mathbb{N} \longrightarrow \hat{x}$ for example $\nu(0) = (2,0,-1)$ the weight vector is corrected accordingly,

$w(1) = w(0) + \nu(0) = (0,2,1) + (2,0,-1) = (0+2, 2+0, 1+-1) = (2,2,0)$

(By component wise addition). The algorithm only stops if a "w" is found with $\langle w, x\rangle > 0$ for all $x \in \hat{x}$. similarly as $(w(1))$ continuing we obtain the following: since $w(1) = (2,2,0)$ and

$\hat{x} = \{(0,0,1),(0,1,1),(1,0,1),(-1,1,1),(2,-1,-1),(2,0,-1),(1,0,-1),$
$(1, 1,-1)\}$. Now $w(1)\hat{x} > 0$ for all except (0, 0, 1), (-1, 1, 1).

$W(2) = w(1) + \nu(1) = (2,2,0) + (0,0,1) = (2+0, 2+0, 0+1) = (2,2,1)$. Once again $w(2)\hat{x} > 0$ for all elements of $\hat{x}$.

Hence, $W(2) = (2,2,1) \in \mathbb{R}^3$ is the required vector.

∴The corresponding affine separation of X_+ and X_- is $2x_1 + 2x_2 + 1 = 0$ and the corresponding perceptron has the weights $w = (2,2)^T$ and the threshold $\theta = -1$ since $2x_1 + 2x_2 + 1 = 0$ is equivalent to $w_1 x_1 + w_2 x_2 - \theta = 0$. ◊

In this chapter, the perceptron tries to model the complex data processing of nerve cell by using simple models such as Heaviside, Sign, Sigmoid and Transfer functions. But, there are difficulties in the case of the XOR function.

In general, in this chapter we have seen about simple perceptron and some methods such as formal neuron, affine separation, separation of finite sets and perceptron learning algorithm. The target of the perceptron was to be model of nerve cell. However, the perceptron faced a problem. This problem was that it could not able to model the XOR function. This is because of it is a single layer and single layer cannot model this function. Similarly, those methods introduced in the chapter assumed to solve the problem also could not able to change the situation. Therefore, we need to introduce another layer called Hidden layer in the next chapter.

3. FEED FORWARD NEURAL NETWORK

The efficiency of perceptron is quite limited. Since the exclusive or function cannot be realized by using a single layer perceptron. However, the construction of multilayer Neural Networks that will be introduced in this chapter, which has the capacity to realize all switching functions, remedies the matter.

Contents of this section are:

- Structure of Feed Forward Neural Network
- Feed Forward Neural Network
- k- Layer Feed Forward Neural Network
- k – Layer σ - perceptron
- Realization by Multilayer Perceptrons

3. 1 Structure of Feed Forward Neural Network

Definition 1.20:

a. A (finite, simple, directed) graph $G = (V, E)$ is composed of a non-empty set of vertices or nodes V and a set of edges $E \subseteq V \times V$.

b. $P(i) := \{j \in V : (j\ ,\ i) \in E\}$ is the set of all direct predecessors of the node $i \in V$.
$S(i) = \{\, j \in V : (i\ ,\ j) \in E\}$ is the set of direct successors of the node $i \in V$.

c. A vertex i with $P(i)\ =\ \phi$ is called a **source.**
A node i with $S(i)\ =\ \phi$ is said to be a **sink**.

d. For $i_o,\ \cdots, i_l\ \in V$, let $e_j :=\ (\ i_{j-1}\ ,\ i) \in\ E$ apply for all $j = 1, \dots, l$. The corresponding sequence of edges $(e_1, \dots, e_l)$ is called **a path** from $i_o\ to\ i_l$ and the number of edges l is its length.

e. A path whose first and last nodes coincide is called a **cycle**, and a graph that is devoid of cycles is called **acyclic graph**.

Lemma 1.7: An acyclic graph contains a source and a sink.

Proof:

Suppose that $G = (V, E)$ contains no sink, then any vertex posses a direct successor. This implies that there exist paths of arbitrary length.

Consider a path $(e_1, \cdots, e_l)$ whose length exceeds the finite number of edges of the graph. However, the edges cannot be distinct. That is, we must have $e_i = e_j$ for some $1 \leq i < j \leq 1$. But, $(e_1, \cdots, e_l)$ has to be a cycle. The dual statement on sources is proven by reversing the orientation of each edge of the graph.

Let $G = (V, E)$ be an acyclic graph. Then there exists an integer $k \geq -1$ and a natural partition of V, $V = V_o \cup, \cdots, \cup V_{k+1}$ (1.2)

, which is constructed as follows: V_o contains all sources.

Remove from G all vertices in V_o and all edges that start in one of them. The resulting graph is again acyclic. Let V_1 be the set of all sources in the new graph, etc.

Thus, $i \in V_j \Rightarrow P(i) \subseteq V_o \cup \ldots \cup V_{j-1}$ and $S(i) \subseteq V_{j+1} \cup \ldots \cup V_{k+1}$. The elements of V_{k+1} are sinks. ◊

Definition 1.21: Let $G = (V, E)$ be an acyclic graph then a F**eed Forward Neural Network** $\boldsymbol{\mathcal{F}}$ is composed of the graph G and a family of formal neurons$(X_i, Y_i, \sigma_i, S_i)$, each associated to one of the non-source vertices $i \in V \setminus V_o$, where $x_i \in \mathbb{R}^{n_i}$, $n_i = |P(i)|$. The transfer functions of these neurons are (f_i) $i \in V \setminus V_o$ with $f_i = \sigma_i \, o S_i : X_i \rightarrow Y_i$.

Example: Simple feed forward neural network.

The compatibility requirement $\prod_{j \in P(i)} y_j \subseteq X_i$ will be satisfied if we set $X_i = \mathbb{Q}^{n_i}$, $Y_i = \mathbb{Q}$ for all i and for some set $\mathbb{Q} \subseteq \mathbb{R}$.

For simplicity, we make this assumption in the following, and then each node $i \in V$ has a state $q_i \in \mathbb{Q}$. For $i \in V_o$, the states will have to be imposed by some additional initial condition. For all $i \in V \setminus V_o$, the state is determined by the states of the direct predecessors of i, via,

$$q_i = f_i \left((q_j) j \in P(i) \right) .$$

The nodes in V_o , $V \backslash V_o \cup V_{k+1}$ and V_{k+1} are called **input**, **hidden** and **output nodes**, respectively. Similarly, the associated neurons are called **input**, **hidden** and **output neurons**.

Now, the transfer function is the mapping:

$f: \mathbb{Q}^{|V_o|} \rightarrow \mathbb{Q}^{|V_{k+1}|}$, which maps the vector of input states to the vector of output states.

Feed forward neural networks allow signals to travel one way only, from input to output. There is **no feedback**. The output of any layer does not affect that same layer. Feed forward neural networks tend to be straight forward networks that associate inputs with outputs. They are extensively used in **pattern recognition**, also referred to as bottom-up or top – down.

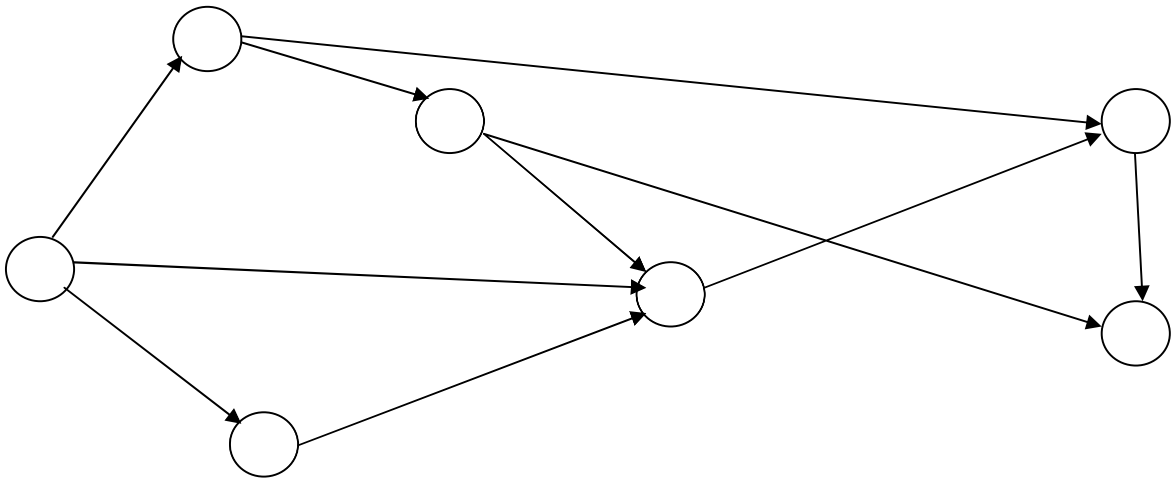

Fig 1.6: Graph of a Feed Forward Neural Network without a Layer Structure.

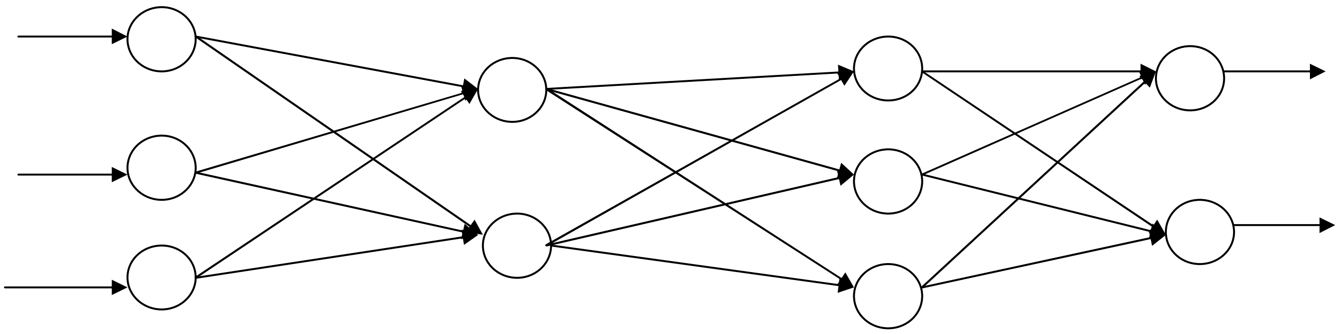

Fig 1. 7: Graph of a 2- layer feed forward Neural Network.

A connection is allowed from a node in layer i only to nodes in layer $i+1$. Most widely used architecture.

Definition 1.22: A feed forward neural network $\mathcal{F}$ with $V = V_o \cup, \cdots, \cup V_{k+1}$ according to (1.2) is called a ***k*-layer feed forward neural network** if $i \in V_j \Rightarrow P(i) \subseteq V_{j-1}$ and $S(i) \subseteq V_{j+1}$ for $j = 0, \ldots k+1$ (putting $V_{-1} = V_{k+1} = \phi$).

We shall restrict to the case where the layers are fully interconnected, that is,
$i \in V_j \Rightarrow P(i) = V_{j-1}$ and $S(i) = V_{j+1}$ for $j = 0, \ldots k+1$. Then V_o is the set of all sources (by construction),V_{k+1} is the set of all sinks.
We write $V_o = V_I$ and $V_{k+1} = V_o$, where the subscripts refer to input and output respectively.

Definition 1.23: A k-layer feed forward neural network whose all neurons are $(\sigma-)$ perceptrons, is called **a *k*-layer** $(\sigma-)$ **perceptrons.**
The transfer functions of a k- layer $\sigma-$ perception is a composition:

$$f = f^{(k+1)} \circ \ldots \circ f^{(1)}: \mathbb{Q}^{|V_I|} \to \mathbb{Q}^{|V_o|}$$

Here, of $k+1$ is mappings of the form

$$f^{(i)}: \mathbb{Q}^{|V_{i-1}|} \to \mathbb{Q}^{|V_i|}$$

$$q \to \underline{\sigma}(\mathrm{w}^{(\mathrm{i})} q - \theta^{(i)})$$

Where $\mathrm{w}^{(\mathrm{i})} \in \mathbb{R}^{|V_{i-1}| \times |V_i|}$ is a fixed weight matrix, $\theta^{(i)} \in \mathbb{R}^{|V_i|}$ is a threshold vector, and

$\underline{\sigma}$ is a vector – valued sigmoid function defined by component wise application of $\sigma: \mathbb{R} \to \mathbb{Q}$. That is, for any integer $n > 0$

$$\underline{\sigma}\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \sigma(x_1) \\ \vdots \\ \sigma(x_n) \end{pmatrix} \quad (1.3)$$

For $i = 1, \cdots, k$, the mapping $f^{(i)}$ is called **transfer function** of i^{th} hidden layer and $f^{(k+1)}$ is referred to as **transfer function** of the output layer. Unfortunately, the counting of layers for multilayer neural networks is not specified in the literature.

Here, we will stick to counting the hidden layers only.

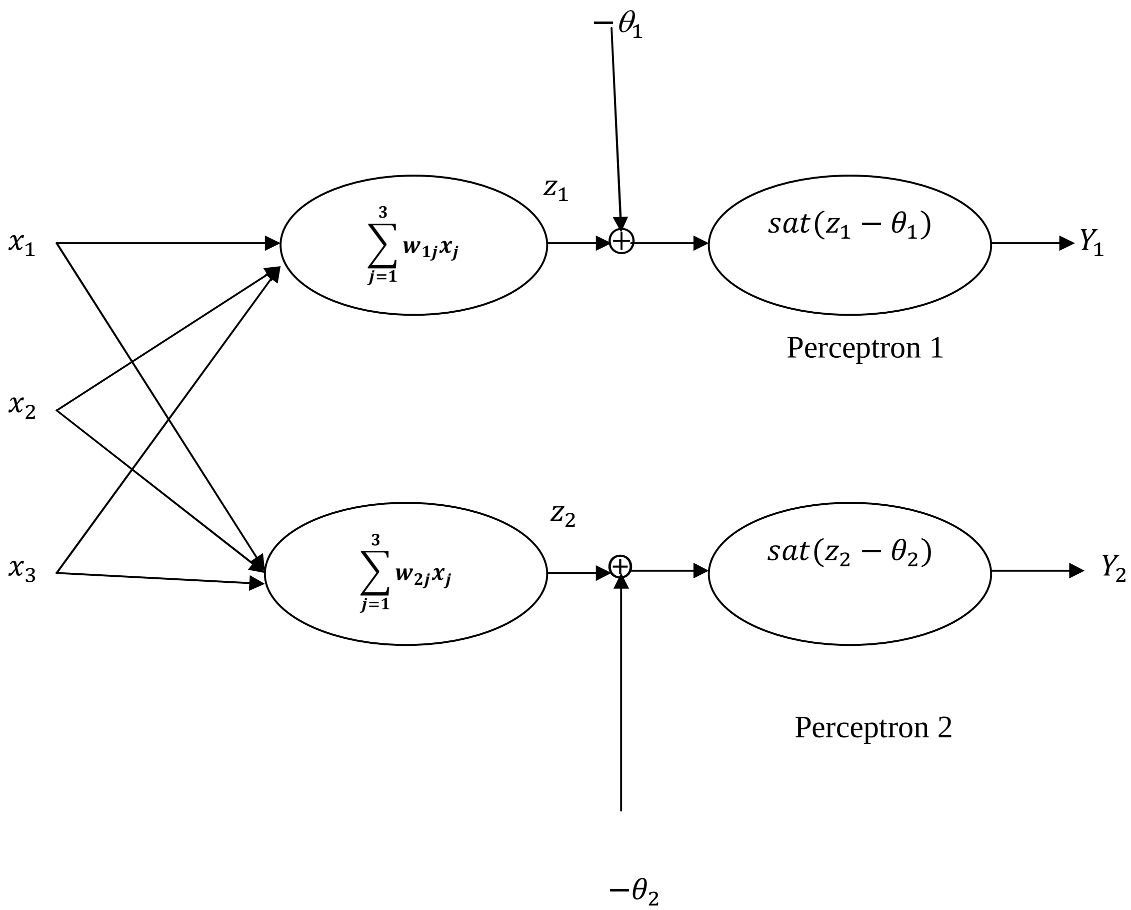

Fig.1.8: A layer of two perceptrons in a multilayer perceptrons.

3.2 Realization by Multilayer perceptrons: Realization of Boolean Functions.

As the XOR- problem in lemma 2.1 shows, the realization of arbitrary switching functions by a single perceptron is impossible. However, this situation is different when a multilayer perceptron is used.

Here, the set of possible states of neurons will be $\mathbb{Q} = \{0,1\}$ and the transfer functions considered will have the form $f: \{0,1\}^n \rightarrow \{0,1\}^p$. That is, we consider feed forward neural networks with 'n' input nodes and 'p' output neurons.

A simple observation is the fact, as long as the number of hidden neurons is unconstrained, $f = (f_1, \cdots, f_p)$ can be realized by such a feed forward neural network if and only if all its components $f_i: \{0,1\}^n \rightarrow \{0,1\}$ can be realized by a feed forward neural network (with 'n' input nodes and one output neuron).

Theorem 1.8: Any Boolean function $f: \{0,1\}^n \rightarrow \{0,1\}^p$ can be realized by 1 – layer perceptron.

Proof:

Let $p = 1$ without loss of generality, let $X = \{0,1\}^n$, $X_- = f^{-1}(0) \subseteq X$, $X_+ = f^{-1}(1) \subseteq X = \{ x_1, \cdots, x_k \}$, and $X_i = \{x_i\}$ for $i = 1, \cdots, k$. Then $\text{Conv}(X_i) = X_i$ and $\text{Conv}(X_i) \cap \text{Conv}(X / X_i) = \phi$. This follows from the fact that x_i is an extreme point of the unit cube $\text{Conv}(X)$. This signifies that $\text{Conv}(X) \setminus X_i$ is still Convex, and therefore, $\text{Conv}(X \setminus X_i) \subseteq \text{Conv}(X) \setminus X_i$.

Thus, X_i and $X \setminus X_i$ are affinely separable. (Since $\text{Conv}(X_i) \cap \text{Conv}\left(X/_{X_i}\right) = \emptyset$).

For $i = 1, \cdots, k$, let $\langle w^{(i)}, x \rangle = \theta^{(i)}$ be a hyperplane that separates X_i from $X \setminus X_i$.

i.e for $1 \leq i \leq k, X_i \subset H_i := \{x \in \mathbb{R}^n: \langle w^{(i)}, x \rangle \geq \theta^{(i)}\}$ and $X_i = X \cap H_i$.

Define $g^{(1)}(x) = y = (y_1, \cdots, y_k)^T$ by $y_i = sat(\langle w^{(i)}, x \rangle - \theta^{(i)})$ for $i = 1, \ldots, k$. Then with $g^{(2)}(y) = z = sat\,(y_1 + \cdots + y_k - 0.5)$ we have constructed a function $g = g^{(2)} o g^{(1)}$ that coincides with the given function f due to.

$z = g(x) = 1 \Leftrightarrow \exists\, i \in \{1, \cdots, k\}$ with $y_i = 1$

$\Leftrightarrow \exists\ i \in \{1, \cdots, k\}$ with $\langle w^{(i)}, x \rangle \geq \theta^{(i)}$

$\Leftrightarrow \exists\ i \in \{1, \cdots, k\}$ With $x \in X_i$

$\Leftrightarrow x \in X_i$ (Since the output is 1)

$\Leftrightarrow x \in f^{-1}(1) \Leftrightarrow f(x) = 1$

Hence, the proof of the theorem. ◊

Example: The XOR- function shall be realized by a perceptron with **one hidden layer** (3.4).

Solution:

For the XOR- problem with $X_+ = \{(0,1), (1,0)\}$ and $X_- = \{(0,0), (1,1)\}$, let $X_1 = \{(0,1)\}$ ad $X_2 = \{(1,0)\}$.

Hence, we have the half spaces,

$H_1 = \{(x_1, x_2) \in \mathbb{R}^2 : x_2 - x_1 \geq 0.5 \}$ satisfying X_1.

$H_2 = \{(x_1, x_2) \in \mathbb{R}^2 : x_1 - x_2 \geq 0.5 \}$ satisfying X_2.

And thus,

$$y_1 = sat\left(x_2 - x_1 - 0.5\right)$$

$$y_2 = sat\ (x_1 - x_2 - 0.5)$$

$$z = sat\ (y_1 + y_2 - 0.5)$$

This can be summarized in the following table:

x_1	x_2	y_1	y_2	z
0	0	0	0	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

Let us draw the graph of H_1 and H_2 as can be seen below:

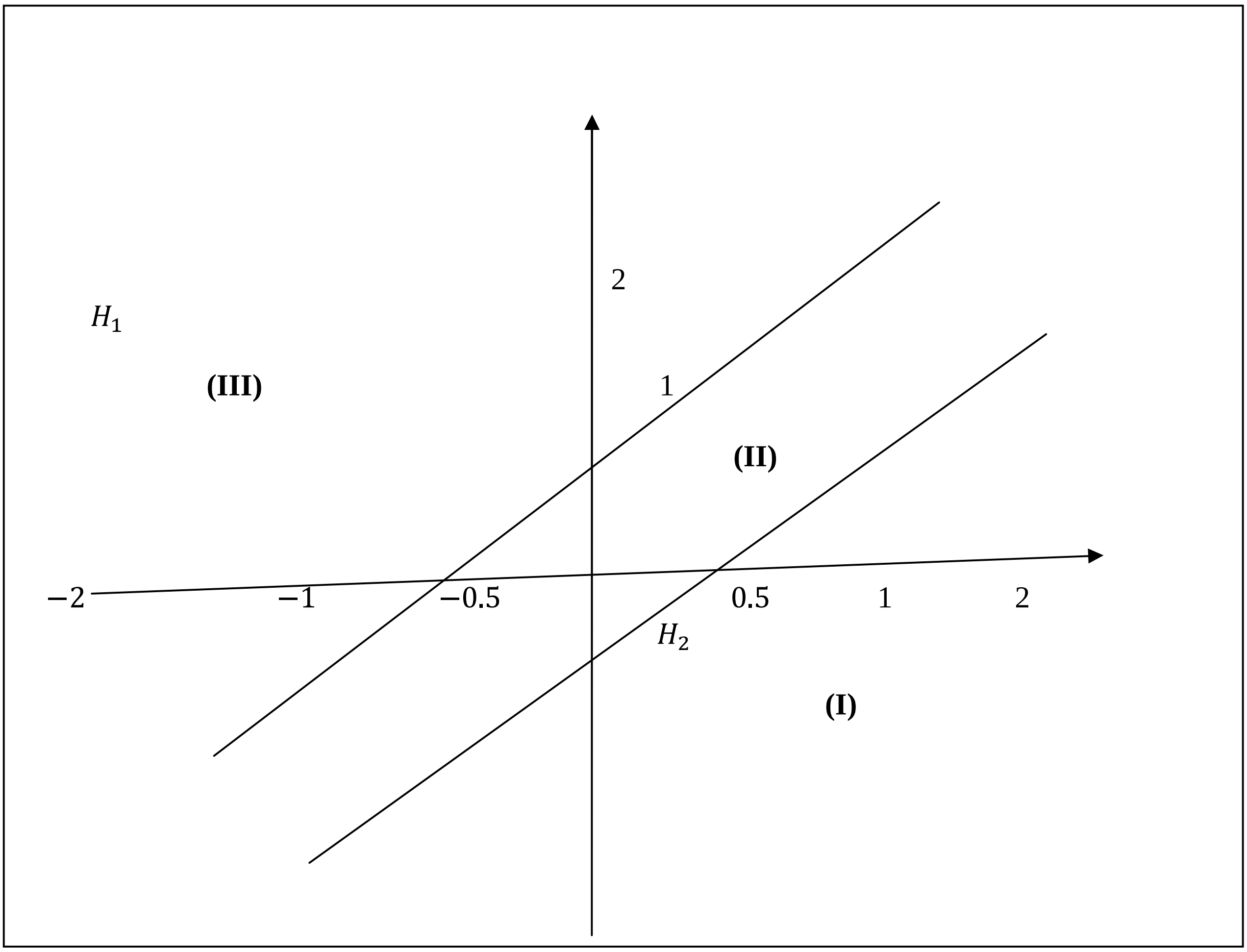

Fig 1.9: Realization of the XOR – function by a perceptron with one hidden layer (in region **I** neuron 1 fires; In region **II** no neuron fires; In region **III** neuron 2 fires)

Limitation of Feed Forward Neural Network

Since Feed forward Neural Networks allow signals to travel one way only. There is no feedback connection, self-loops (the neuron in the network cannot use its own output as an input). But, all data cannot be processed in this manner. For instance, in human being data processing type is not unidirectional. Therefore, we need to introduce a bidirectional data processing type called Recurrent Neural Network in the following chapter.

CHAPTER TWO: RECURRENT NEURAL NETWORK

A Recurrent Neural Network consists of formal neurons whose interconnection structure is give by a graph. In contrast to the previous section, **no assumption** is made about the acyclicity of this underling network graph. Therefore, there is no well defined direction of information propagation, and thus, recurrent neural networks are **modeled a**s **dynamical systems**.

We start by reviewing some basic facts from the theory of discrete dynamical systems, and then we study the structural properties of **Hopfield neural networks** with respect to convergence to **fixed points**, **transient length**, and **attractivity**. In general, contents of this chapter are the following:

- Finite Automata
- Structure and Convergence of Recurrent Neural Network
- Asynchronous update
- Transient Length and Attractivity

2.1 Finite Automata

Let X be a set and let $f: \mathbb{Z} \times X \longrightarrow X$ be a function.

Consider the equation,

$$X(t+1) = f(t, x(t)) \qquad (2.1)$$

Where, $x \in X$ and $t \in \mathbb{Z}$. Together with an initial condition $x(t_o) = x_o$,

Where $t_o \in \mathbb{Z}$ and $x_o \in X$, this recursively defines a **unique** sequence,

$x = (x(t_o), x(t_o+1), x(t_o+2), \ldots)$ of elements of X.

Such sequences are called **trajectories** of (2.1), and the set of all trajectories is called a **discrete dynamical system**. We call $x(t)$ the state of the system at time t (when starting in state x_o at time t_o). If X is finite, such a system is called a **finite automaton**.

A point $\overline{x} \in X$ is called a **fixed point** of (2.1) if there exists a $t_o \in \mathbb{Z}$ such that $x(t_o) = \overline{x}$ implies that $x(t+1) = x(t)$ for all $t_o \leq t \in \mathbb{Z}$. That is, $x(t) = \overline{x}$ for all $t \geq t_o$. This is equivalent to saying that there exists $t_o \in \mathbb{Z}$ with

$$f(t,\overline{x}) = \overline{x} \text{ for } all\ t_o \leq t \in \mathbb{Z} \text{ (by 2.1)}$$

For an integer $k \geq 1$, a vector $(\overline{x_o}, \dots, \overline{x_{k-1}}) \in x^k$ is called a **cycle** of (2.1) of length k if there exists a $t_o \in \mathbb{Z}$ such that $x(t_o) = \overline{x}_o$ implies $x(t_o + l) = \overline{x}_l$ for $l = 0, \dots, k-1$ and $x(t+k) = x(t)$ for all $t_o \leq t \in \mathbb{Z}$.

Remark 2.1: - A fixed point is precisely a cycle of length one.

If $f : X \longrightarrow X$ doesn't depend on time t, we write

$$x(t+1) = f(x(t)) \tag{2.2}$$

We call such an equation **autonomous**. That is, equation (2.2) doesn't depend on t. Then it suffices to consider the initial time $t_o = 0$. That is, trajectories $x = (x(0), x(1), x(2), \dots)$, where $x(t) = f^t(x(o))$ for all $t \in \mathbb{N}$.

Here, f^t denotes the t-fold composition of f with itself. The fixed points of (2.2) are characterized by the equation $f(x) = \overline{x}$.

A cycle of (2.2) corresponds to a vector $(\overline{x_o}, \dots, \overline{x}_{k-1})$ of length k with

$$f(\overline{x_o}) = \overline{x_1}, \dots, f(\overline{x_{k-2}}) = \overline{x}_{k-1}, \quad f(\overline{x_{k-1}}) = \overline{x_o}.$$

- A function $E : X \longrightarrow \mathbb{R}$ is called a **Lyapunov function** for f if $E(f(x)) \leq E(x)$ for all $x \in X$.

Let $(x(0), x(1), x(2), \dots)$ be a trajectory of (2.2). Then a Lyapunov function E satisfies $E(x(t+1)) \leq E(x(t))$ for all $t \in \mathbb{N}$. That is, E decreases along the trajectories of (2.2). If we even have $x \neq f(x)$ implies $E(f(x)) < E(x)$, then we say that E is **a strict Lyapunov function** for f. This signifies that along the trajectory x of (2.2), E decreases strictly whenever the state changes:

$$x(t) \neq x(t+1) \Rightarrow E\big(x(t+1)\big) < E\big(x(t)\big).$$

Theorem 2.1:- If X is finite, the existence of a strict Lyapunov function for f implies that any trajectory of (2.2) reaches a fixed point.

Proof: -

Suppose X is a finite set and a strict Lyapunov function E for f exists.

Then E can only take finitely many values. This implies that there can be only a finite number of changes in the trajectory $x = (x(0), x(1), x(2), \ldots)$ i.e., there are only finitely many times steps t for which $x(t) \neq x(t+1)$.

Let T be the largest integer for which $x(T) \neq x(T+1)$.

Then $x(T+1) = \bar{x}$ must be a fixed point of f.

Hence, the proof of Theorem 4.1. ◊

Definition 2.1: - Let x be a trajectory of (2.2). The number of times steps t with $x(t) \neq x(t+1)$ is called **transient length** of x.

Remark 2.2:- A trajectory reaches a fixed point if and only if its transient length is finite.

Definition 2.2: - A metric d on a set X is a function $d: X \times X \to \mathbb{R}$ having the following properties:

1. $d(x, y) \geq 0$ for all $x, y \in X$; $d(x, y) = 0 \Leftrightarrow x = y$
2. $d(x, y) = d(y, x)$ for all $x, y \in X$. (symmetry)
3. $d(x, z) \leq d(x, y) + d(y, z)$ (triangular inequality)

Example: $d(x, y) = \begin{cases} 1 \; if \; x \neq y \\ 0 \quad Otherwise \end{cases}$ is a metric since it satisfies the above three properties.

Remark 2.3:- A metric on a set X sometimes called **distance function** on the set X.

Now let X be equipped with a **metric** $d: X \times X \longrightarrow \mathbb{R}$. For $\bar{x} \in X$ and $r > 0$, let

$$B_r(\bar{x}) = \{x \in X : d(x, \bar{x}) \leq r\}.$$

We say that B is a **neighborhood** of $\bar{x}$ if $\bar{x} \in B$. But, $B \neq \{\bar{x}\}$.

Now let $\bar{x}$ be a fixed point of (2.1). We say that $\bar{x}$ is **attractive** if there exists $t_o \in \mathbb{Z}$ and a real number $r > 0$ such that

i. $B_r(\bar{x})$ is a neighborhood of $\bar{x}$

ii. $x_o \in B_r(\bar{x}) \Rightarrow \lim_{t \to \infty} x(t) = \bar{x}$.

Where,$x(t)$ is the state at time t when starting in state x_o at time t_o.

- In the autonomous case, a fixed point $\overline{x}$ is attractive if there exists r as above with

$$x_o \in \mathrm{B_r}(\overline{x}) \Rightarrow \lim_{t\to\infty} f^t(x_o) = \overline{x} \qquad (\propto)$$

If X is additionally finite, ($\propto$) simplifies to

$$x_o \in \mathrm{B_r}(\overline{x}) \Rightarrow \text{ there exist } t \in \mathbb{N} : \; f^t(x) = \overline{x} \qquad (\beta)$$

- The largest $r \in$ image of d ($im(d)$) with (β) property is called the **radius of attraction** of $\overline{x}$.
- Similarly, the largest r as (above) with

$x \in B_r(\overline{x}) \Rightarrow f(x) = \overline{x}$ is called the **radius of direct attraction** of $\overline{x}$,

and the set

$\{x \in X : f(x) = \overline{x}\}$ is called the **domain of direct attraction** of $\overline{x}$.

2.2 Structure and Convergence of Recurrent Neural Network

Definition 2.3:- A **recurrent neural network** $\mathcal{R}$ consists of a (finite, directed, simple) graph $G = (V, E)$ and a family of formal neurons $(X_i, Y_i, \sigma_i, S_i)$, each associated to one of the vertices $i \in V$.

- We suppose that for some set $\mathbb{Q} \subseteq \mathbb{R}$, $X_i = \mathbb{Q}^{n_i}$ and $Y_i = \mathbb{Q}$ for all i.

  Where, $n_i = |p(i)|$. Then each neuron has a transfer function,

$$f_i : X_i \longrightarrow Y_i$$

$$f_i : \mathbb{Q}^{n_i} \longrightarrow \mathbb{Q}$$

Now let $n = |v|$. For $t \in \mathbb{N}$, and $i = 1, \dots, n$, we set

$$q_i(t+1) = f_i\Big(\big(q_j(t)\big) j \in p(i)\Big).$$

And we call $q_i(t) \in Q$ the **state** of neuron i at time t. Accordingly, the vector $q(t) = (q_1(t), \ldots, q_n(t)$ is called the **state of the network** at time t.

Definition 2.4:- A recurrent neural network is called a **Hopfield neural network** if the graph G is fully interconnected. i.e. $E = V \times V$, and if all the neurons are **perceptrons.** Then $\mathbb{Q} = \{0, 1\}$ and $q_i(t+1) = f_i(q_1(t), \ldots, q_n(t))$, which can be written in the concise form (putting $f = (f_1 \ldots f_n)$)

$$q(t+1) = f(q(t))$$

Where, $f(x) = \underline{sat}\,(Wx - \theta)$ for some weight matrix $W \in \mathbb{R}^{nxn}$ and some threshold vector $\theta \in \mathbb{R}^n$.

In the language of the previous section, the Hopfield neural network equation $q(t+1) = \underline{sat}\,(wq(t) - \theta)$ defines an **autonomous finite automaton** with $X = \{0,1\}^n$.

A point $\overline{x} \in X$ is a fixed point of this equation if $\overline{x} = \underline{sat}(w\,\overline{x} - \theta)$.

In the following, we will tacitly assume that w and θ are non-zero for all $x \in X = \{0,1\}^n$. This can always be achieved by a modification of the threshold vector, without changing the values of $f(x) = sat\,(wx - \theta)$ on X. Another assumption frequently made below is that W is a symmetric matrix. This means that the strength of interconnection between neuron i and neuron j equals the strength of interconnection between j and i.

Theorem 2.2: Let W be a symmetric matrix. Then each trajectory of a Hopfield neural network reaches a fixed point or a cycle of length 2.

Proof:-

Define $F : X \longrightarrow \mathbb{R}$ by $F(x) = -f(x)^T W x + (f(x)^T + x^T)\theta$

Then $F\big(f(x)\big) = -f\big(f(x)\big)^T W f(x) + (f\big(f(x)\big)^T + f(x)^T)\theta$

Using the symmetry of W, this yields:

$$F\big(f(x)\big) - F(x) = -f\big(f(x)\big)^T W f(x) + (f\big(f(x)\big)^T + f(x)^T)\theta$$

$$- [- f(x)^T W x + (f(x)^T + x^T)\theta]$$

$$= -f\big(f(x)\big)^T W f(x) + \Big(f\big(f(x)\big)^T + f(x)^T\Big)\theta + f(x)^T W x$$

$$-f(x)^T\theta - x^T\theta$$

$$= -f\big(f(x)\big)^T Wf(x) + f\big(f(x)\big)^T\theta + f(x)^T Wx - x^T\theta$$

$$= -f\big(f(x)\big)^T Wf(x) + f(f(x)^T\theta + x^T Wf(x) - x^T\theta$$

$$= -f\big(f(x)\big)^T (Wf(x) - \theta) + x^T(wf(x) - \theta)$$

$$= (x^T - f\big(f(x)\big)^T)(Wf(x) - \theta)$$

$$= \Big(x - f\big(f(x)\big)\Big)^T (Wf(x) - \theta)$$ Since $A^T - B^T = (A-B)^T$

$$= \langle x - f\big(f(x)\big), wf(x) - \theta\rangle$$

$$= \sum_{i=1}^{n} (x - f(f(x)))_i \, (wf(x) - \theta)_i$$

$$= \sum_{i=1}^{n} (x_i - f_i(f(x))) \, (wf(x) - \theta)_i$$

Now for each of these summands, there are four cases, summarized in the following table:

x_i	$f_i(f(x))$	$(x_i - f_i(f(x))$	$(Wf(x) - \theta)_i$
0	0	0	< 0
1	0	1	< 0
0	1	-1	> 0
1	1	0	> 0

The last column is obtained by nothing that $f(f(x)) = \underline{\boldsymbol{sat}}\,(Wf(x) - \theta)$

Thus, $f_i(f(x)) = \boldsymbol{sat}(wf(x) - \theta)_i$.

We conclude that $x \neq f(f(x)) \Rightarrow F(f(x)) < F(x)$. Since F can only take finitely many values, there exist, for every trajectory $x = (x(0), x(1), x(2), \dots)$ only finitely many times steps t with $x(t+2) \neq x(t)$.

Thus, there exists a $t_o \in \mathbb{N}$ such that $x(t+2) = x(t)$ for all $t \geq t_o$. ◊

Note: $f(x)^T W x = [\ x^T W^T f(x)\ ]^T$

Theorem 2.3: Let W be a symmetric matrix and assume that $x^T W x \geq 0$ for all x in $\{-1, 0, 1\}^n$. Then $E(x) = -\frac{1}{2}x^T W x + x^T \theta$ is a strict Lyapunov function for the Hopfield neural network. According to Theorem 2.1, this implies that each trajectory reaches a fixed point.

Proof:-

Since $E(x) = -\frac{1}{2}x^T W x + x^T \theta$, we have $E(f(x)) = -\frac{1}{2}\big(f(x)\big)^T W f(x) + (f(x))^T \theta$.
Then using the symmetry of W, this implies

$$E(f(x)) - E(x) = -\tfrac{1}{2}(x^T - f(x)^T) W \big(x - f(x)\big) + (x^T - f(x)^T)(Wx - \theta).$$

Since $x, f(x) \in \{0,1\}^n$ we have $x - f(x) \in \{-1, 0, 1\}^n$ and therefore the first summands is non-positive due to our assumption on W. For the second summand, we consider again the possible cases for a particular component:

x_i	$f_i(x)$	$x_i - f_i(x)$	$(Wx - \theta)_i$
0	0	0	< 0
1	0	1	< 0
0	1	-1	> 0
1	1	0	> 0

The last column comes again from $f(x) = \underline{sat}(Wx - \theta)$

That is, $f_i(x) = sat(wx - \theta)_i$

Thus, we obtain

$x \neq f(x) \Rightarrow E(f(x)) < E(x)$ as desired. ◊

2.3 Asynchronous update

Definition 2.5: Let m be an integer. Then integers a and b are said to be congruent to one another $modulom$, written $a \equiv bmodm$ if and only if $a - b = mq$ for some $q \in \mathbb{Z}$ and b is called a residue of modulo$m$.

Example: $35 \equiv 8mod27$. Since $35 - 8 = 27q$. Then $q = 1 \in \mathbb{Z}$.

The sufficient condition of Theorem 2.3 is often too restrictive. One way out is to compute the **new state vector** $q(t+1) = f(q(t))$ component for component, and to use the already updated components,

$q_1(t+1)$, ⋯, $q_{i-1}(t+1)$ for the calculation of $q_i(t+1)$.

The resulting computation procedure is:

$$q_1(t+1) = f_1\big(q_1(t), q_2(t), \cdots, q_n(t)\big)$$

$$q_2(t+1) = f_2\big(q_1(t+1), q_2(t), \cdots, q_n(t)\big) \qquad (2.3)$$

$$\vdots$$

$$q_n(t+1) = f_n(q_1(t+1), q_2(t), \cdots\ q_{n-1}(t+1), q_n(t))$$

We want to recast these equations in the form discussed in section 2.1. For this, we set $h = \frac{1}{n}$ and $x(t) = q(t)$,

$$x(t+h) = \begin{bmatrix} f_1(x(t)) \\ q_2(t) \\ . \\ . \\ . \\ q_n(t) \end{bmatrix}, \dots, x(t+(n-1)h) = \begin{bmatrix} f_1\big(x(t)\big) \\ . \\ . \\ . \\ f_{n-1(x(t+(n-2)h))} \\ q_n(t) \end{bmatrix}$$

and $x(t+nh) = x(t+1) = q(t+1)$. After re-scaling the time axis, we have

$$x_i(t+1) = \begin{cases} f_i\big(x(t)\big) if i \equiv t+1 mod n \\ x_i(t) Otherwise \end{cases} \quad (2.4)$$

Where $i = 1, \ldots, n$

Model (2.4) is equivalent to (2.3), and it has the form discussed in section 2.1. However, note that the equation is not autonomous. We call (2.4) a sequential updating scheme. More generally, for a sequence $h: N \longrightarrow \{1, \ldots, n\}$ we call

$$x_i(t+1) = \begin{cases} f_i\big(x(t)\big) if\ i = h(t) \\ x_i(t) Otherwise \end{cases}$$

An **asynchronous updating scheme**.

Usually, one requires that h is such that for all $i \in \{1, \ldots, n\}$, and for all $k_o \in \mathbb{N}$ there exists $k \geq k_o$ such that $h(k) = i$. This guarantees that each component is updated infinitely many times. This notion should be compared with the concept of a training sequence, used in section 2. The sequential updating scheme corresponds to the special case where $h = (1, \ldots, n, 1, \ldots, n, \ldots)$. For practical purposes it suffices to consider the sequential update. Consider the Hopfield neural network equation with sequential update, that is

$$x_i(t+1) = \begin{cases} f_i\big(x(t)\big) \\ x_i(t)\ otherwise \end{cases} \text{if } i \equiv t+1 mod n \quad (2.5)$$

First, we note that the notion of fixed points does not depend on the choice of the updating mode.

Lemma 2.4: A point $\overline{x} \in X = \{0,1\}^n$ is a fixed point of (2.5) if and only if $\overline{x} = \underline{sat}(W\,\overline{x} - \theta)$.

Proof:

(⇒) Suppose a point $\overline{x} \in X = \{0,1\}^n$ is a fixed point of (4.5) .

Claim: $\overline{x} = \underline{sat}(W\overline{x} - \theta)$.

$Now\ x_i(t+1) = sat(W\,x(t) - \theta)_i = \overline{x}$ (by the supposition).

$$\Rightarrow \underline{sat}(Wx(t) - \theta) = \overline{x}$$

$$\Rightarrow sat(Wx(t) - \theta)_i = \overline{x}$$

$$\Rightarrow\ x_i(t) = \overline{x}$$

$$\Rightarrow\ sat(W\overline{x} - \theta) = \overline{x}$$

Hence, the required claim. ◊

(⇐) Suppose $\overline{x} = \underline{sat}(W\overline{x} - \theta)$.

Claim: A point $\overline{x} \in X = \{0,1\}^n$ is a fixed point of (2.5)

Since fixed points do not depend on the choice of the updating mode, we have

$x_i(t+1) = sat(W\,x(t) - \theta)_i$ from (2.5). Again further, we have

$x_i(t+1) = sat(W\,x(t) - \theta)_i = \overline{x}$ (by the supposition).

$$\Rightarrow\ x_i(t+1) = \overline{x} = x_i(t)$$

∴ A point $\overline{x} \in X = \{0,1\}^n$ is a fixed point of (2.5). ◊

Theorem 2.5: Let W be a symmetric matrix, with $W_{ii} \geq 0$ for all i. Then every trajectory of (2.5) reaches a fixed point.

Proof:-

Consider again the energy function,

$$E(x) = -\tfrac{1}{2}x^T W x + x^T \theta$$

Let x be a trajectory of (2.5) and suppose that $x(t) \neq x(t+1)$.

As usual, it suffices to show that $E\big(x(t+1)\big) < E\big(x(t)\big)$.

Since $E(x) = -\tfrac{1}{2}x^T W x + x^T \theta$

$$\Rightarrow E\big(x(t+1)\big) = -\tfrac{1}{2}x(t+1)^T W x(t+1) + x(t+1)^T\theta$$

and $E\big(x(t)\big) = -\tfrac{1}{2}x(t)^T W x(t) + x(t)^T\theta$.

Consider $E\big(x(t+1)\big) - E\big(x(t)\big) = -\tfrac{1}{2}(x(t)^T - x(t+1)^T)W\big(x(t) - x(t+1)\big) + (x(t)^T - x(t+1)^T)(Wx(t) - \theta)$ (2.6)

Due to the asynchronous update, $x(t)$ and $x(t+1)$ differ only in one component, say $x(t) - x(t+1) = +e_i$, where e_i is the i^{th} natural basis vector of $\mathbb{R}^n$. Therefore, $E(x(t+1)) - E\big(x(t)\big) = -\tfrac{1}{2}W_{ii} \pm (Wx(t) - \theta)_i$

Now, if $x(t) - x(t+1) = +e_i$ then $x_i(t) = 1$ and $x_{i+1}(t) = 0$ and thus, $(Wx(t) - \theta)_i < 0$.

The case $x(t) - x(t+1) = -e_i$ is analogous.

In either cases, $E\big(x(t+1)\big) < E\big(x(t)\big)$.

Hence, the proof of the Theorem. ◊

Remark 2.4:

1. Note that $x^T W x \geq 0$ for all $x \in \{-1,0,1\}^n$. Therefore, the sufficient condition of Theorem 2.3 is strictly stronger than that of Theorem 2.5 showing that asynchronous update is really an advantage.
2. The condition $W_{ii} \geq 0$ is met by the "classical" Hopfield neural network, which even requires $W_{ii} = 0$. This means that no neuron in the network uses its own output as an input, which corresponds to a graph $G = (v, E)$ without self-loops, that is, $(i, i) \notin E$ for $i \in V$.

2.4 Transient Length and Attractivity

For the following discussion, it is convenient to replace $X = \{0, 1\}^n$ by $X = \{+1, -1\}^n$. And correspondingly, we replace the Heaviside function by sign function. All the results obtained so far remain valid.

Theorem 2.6: Let W be a symmetric matrix with $W_{ii} \geq 0$ for all i. According to Theorem 2.3, all trajectories of the Hopfield neural network with sequential update (2.5) reach a fixed point. Moreover, the transient length T satisfies

$$T \leq \frac{\sum_{i=1}^{n} \sum_{j>i} |W_{ij}| + \sum_{i=1}^{n} |\theta_i|}{\omega}$$

Where $\omega = \min W_{ii} \geq 0$ and $\varepsilon > 0$ is such that $|(Wx-\theta)_i| \geq \varepsilon$ for all $x \in X = \{\pm 1\}^n$ and all $i = 1, \dots, n$.

Proof:

Consider once more

$$E(x) = -\tfrac{1}{2} x^T W x + x^T \theta$$

And let $\Delta = \max_{x \in X} E(x) - \min_{x \in X} E(x)$

If $\delta > 0$ is such that for any trajectory $x = (x(0), x(1), x(2), \dots)$
$x(t+1) \neq x(t) \Rightarrow |E(x(t+1)) - E(x(t))| \geq \delta$.
Then, the number of times steps t with $x(t+1) \neq x(t)$ is bounded by $\frac{\Delta}{\delta}$.
It suffices to find an upper bound for Δ. For this, note that for $x \in X$.

$$E(x) = -\frac{1}{2} \sum_{i=1}^{n} W_{ii} x_i^2 - \sum_{i=1}^{n} \sum_{j>i} W_{ij} x_i x_j + x^T \theta$$

$$= -\frac{1}{2} \sum_{i=1}^{n} W_{ii} - \sum_{i=1}^{n} \sum_{j>i} W_{ij} x_i x_j + x^T \theta$$

and thus, setting $\tilde{E}(x) = E(x) + \frac{1}{2} \sum_{i=1}^{n} W_{ii}$, $\Delta = \max_{x \in X} \tilde{E}(x) - \min_{x \in X} \tilde{E}(x)$
One obtains $\Delta = 2(\sum_{i=1}^{n} \sum_{j>i} |W_{ij}| + \sum_{i=1}^{n} |\theta_i|)$.
On the other hand, when $x(t+1) \neq x(t)$, say $x(t) - x(t+1) = \pm 2e_i$ we get from (2.6)

$$E\big(x(t+1)\big) - E\big(x(t)\big) = -2W_{ii} \pm 2(Wx(t) - \theta)_i.$$

And thus

$$\left|E\big(x(t+1)\big)-E\big(x(t)\big)\right| = 2(W_{ii}+|(Wx(t)-\theta)_i|)\geq 2(\omega+\varepsilon) =: \delta$$

This yields the result. ◊

For discussing attractivity, we need to introduce a metric on X.

Definition 2.6: For $x, y \in X = \{\pm 1\}^n$, $d(x,y) = |\{i: x_i \neq y_i, i = 1, \dots, n\}|$ is called the **Hamming distance** between x and y.

Lemma 2.7: d is a metric on X.

Proof:

Claim: d is a metric on X.

i) $d(x,y) = 0$ if and only if x, y agree in all coordinates and this happens if and only if $x = y$. Otherwise, $d(x,y) \neq 0$.
ii) The number of coordinates in which x differs from y is equal to the number of coordinates in which y differs from x. That is, $d(x,y) = d(y,x)$(symmetry).
iii) $d(x,y)$ is equal to the minimum number of coordinate changes necessary to get from x to y. In its turn, $d(y,z)$ is equal to the minimum number of coordinate changes necessary to get from y to Z. So, $d(x,y) + d(y,z)$ changes will get us from x to z.

Hence,$d(x,y) + d(y,z) \geq d(x,z)$, which is the minimum number of coordinate changes necessary to get from x to z.

Hence, the claim. ◊

Let $\bar{x}$ be a fixed point of the Hopfield neural network. That is $\bar{x} = sign(w\bar{x} - \theta)$. Let $r_1(\bar{x})$ denote the radius of direct attraction of $\bar{x}$, which is defined to be the largest integer $r > 1$ with

$$x_o \epsilon B_r(\bar{x}) \Rightarrow sign(w\bar{x} - \theta) = \bar{x}$$

In that case, if we start in $x_o \epsilon B_r(\bar{x})$ and use a sequential update, the trajectory will reach $\bar{x}$ after at most n times steps.

Note: Recall that n sequential times steps correspond to one synchronous time step.

Theorem 2.8: Let $\overline{x}$ be a fixed point of the Hopfield neural network, and assume that $w \neq 0$. Let $\varepsilon > 0$ be such that

$$|W\,\overline{x} - \theta| \geq \varepsilon \text{ for } i = 1, \dots, n. \text{Then } r_1(\overline{x}) \geq \left[\frac{\varepsilon}{2\nu}\right]$$

Where $\nu = \max\left|W_{ij}\right|$, and $[x]$ denotes the greatest integer less than or equal to a real number x.

Proof:

For any $x, \overline{x} \in X$ and any $i = 1, \dots, n$, we have

$$\begin{aligned} |(Wx - \theta)_i - (W\,\overline{x} - \theta)_i| &= |[W(x - \overline{x}) - \theta + \theta)]_i| \\ &= |[W(x - \overline{x})]_i| \\ &= \left|\sum_{j=1}^{n} W_{ij}(x_j - \overline{x_j})\right| \\ &\leq \sum_{j=1}^{n} \left|W_{ij}\right|\left|x_j - \overline{x}_j\right| \\ &\leq 2\nu d(x, \overline{x}) \end{aligned}$$

If $d(x, \overline{x}) \leq \frac{\varepsilon}{2\nu}$, then

$$|(Wx - \theta)_i - (W\,\overline{x} - \theta)_i| \leq 2\nu \frac{\varepsilon}{2\nu} \leq \varepsilon$$

Which implies that $(Wx - \theta)_i$ and $(W\,\overline{x} - \theta)_i$ have the same sign and hence

$$\underline{sat}(Wx - \theta) = \underline{sat}(W\,\overline{x} - \theta) = \overline{x}. \qquad \Diamond$$

CHAPTER THREE: APPLICATIONS OF NEURAL NETWORKS

- **Neural Networks in practice**

  Neural Networks have broad applicability to real world business problems. In fact, they have already been successfully applied in many industries.

  Since Neural Networks are best at identifying patterns or trends in data, they are well suited for prediction or forecasting needs including:

  - Sales forecasting
  - Industrial process control etc

- **Neural Networks in medicine**

  ANNs are currently a 'hot' research area in medicine and it is believed that they will receive extensive applications to biomedical systems in the next few years.

- **Modeling and Diagnosing the cardiovascular system**.

  Neural Networks are used experimentally to model the human cardiovascular system. Diagnosis can be achieved by building a model of the cardiovascular system of an individual and comparing it with the real time physiological measurements taken from the patients. If this routine is carried out regularly, potential harmful medical conditions can be detected at an early stage and thus make the process of combating the disease much easier.

- **Electronic noses**

  ANNs are used experimentally to implement electronic noses. Electronic noses have several potential applications in telemedicine. Telemedicine is the practice of medicine over long distances via a communication link. The electronic noses would identify odours in the remote surgical environment. These identified odours would then be electronically transmitted to another site where an door generation system would recreate them.

- **Instant physician**

An application developed in the mid-1980s called the “Instant physician” trained an autoassociative memory neural network to store a large number of medical records, each of which includes information on symptoms, diagnosis and treatment for a particular case. After training, the net can be presented with input consisting of a set of symptoms; it will then find the full stored pattern that represents the “best” diagnosis and treatment.

- **Neural Network in Business**

  Business is a diverted field with several general areas of specialization such as accounting or financial analysis. There is some potential for using neural networks for business purposes including resource allocation and scheduling.

Conclusions and Future Works

The computing world has a lot to gain from neural networks. Their ability to learn by example makes them very flexible and powerful. Furthermore, there is no need to devise an algorithm in order to perform a specific task. i.e. there is no need to understand the internal mechanisms of that task. They are also very well suited for real time systems because of their fast response and computational times which are due to their parallel architecture.

Neural networks also contribute to other areas of research such as neurology and psychology. They are regularly used to model parts of living organisms and to investigate the internal mechanisms of the brain.

Perhaps the most exciting aspect of neural networks is the possibility that some day ‘conscious’ networks might be produced. There is a number of scientists arguing that consciousness is a mechanical property and that ‘conscious’ neural networks are a realistic possibility.

Finally, I would like to state that even though neural networks have a huge potential we will only get the best of them when they are integrated with some other subjects.

The topic Recurrent Neural Network is fundamental as it derives meaning from complex or imprecise data.

Therefore, my interest towards it is great. So; I need to undertake further study on the topic in the future.

Bibliography

[1] Otto Brestscher, **Linear Algebra with application**, prentice-Hill, Inc, 2001.

[2] Frank Uhlig, **Transformation Algebra**, Prentice-Hill, Inc, 2002.

[3] Arthur Mattuck, **Introduction to Analysis**, Prentice-Hill, Inc, 1999.

[4] W.Rudin, **Functional Analysis**, McGraw-Hill, Inc, 1973.

[5] Seymour Lipschutz, **General Topology**, McGraw-Hill, Inc, 1965.

[6] Eva Zerz, et al. **Mathematical Theory of Neural Networks**, Fachbereich Mathematic Universtat Kaiserslautern.