Chapter 1

Introduction

1.1 Background

Artificial neural networks are computational models that draw inspiration from the nervous systems found in living organisms. The principle working mechanism is motivated by the function of biological neurons. The thesis investigates the stability, synchronization, anti-synchronization, and global dissipativity problems of different neural network models. The dynamics of neural networks play a pivotal role in their practical application to real-world problems, including associative memory, artificial intelligence, secure communication, signal processing, optimization problems, and frequently used in digital communication such as channel identification and equalization, [1, 2, 3, 4], pattern recognition and classification [5], hydrology for real-time forecasting [6]. To illustrate, chaos synchronization in neural networks finds application in secure communication, enhancing the security of signals transmitted from a sender to a receiver. In addressing synchronization issues with the help of a controller in neural networks, control theory is required to stabilize the error systems.

Considering these things, numerous controllers have been designed, including intermittent control [7], linear feedback control, adaptive control [8], and integral sliding mode control [9]. Among these effective controllers, the feedback controller is linear and easy to design and calculate the control cost-effectively, whereas the adaptive controllers deliver effective results in practical applications. These controllers can automatically adjust the coupling strength by designing appropriate adaptive laws. Resulting in reduced control costs, making them more effective , but it is tough to design, and it adopts the system's behaviour and it synchronizes the system even though the parameters of the systems are unknown continuous controllers. This thesis's primary purpose is to study the dynamics of neural networks with different kinds of domains, namely real, complex, and quaternion domains, with different controllers.

The initial part of this introductory chapter begins by outlining the fundamental inspiration behind artificial neural networks, which stem from biological neurons. Let's delve into examining neuron structure, the transmission of neural signals, and how one can create models for the communication among these biological neurons. Subsequently, let's move on to the next section, which defines the definitions for artificial neural networks and introduces some foundational mathematical concepts that will be applied in the subsequent chapters to explore the dynamics of neural networks. In the concluding section of this chapter, the various definitions of synchronization and its distinct forms are introduced.

1.2 Biological neurons

The human brain and the nervous system are composed of an extensive network of interconnected cellular units, including nerve cells or neurons and glial cells. Glial

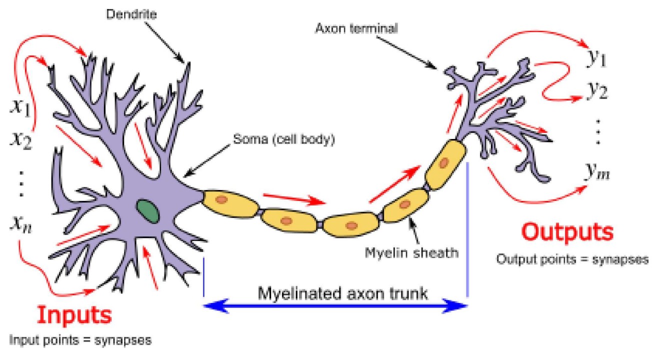


FIGURE 1.1: Schematic diagram of a single neuron.

cells play a supportive role by providing physical and functional support to neurons. While neurons exhibit a diverse array of shapes, sizes, and locations, the majority of them possess uniform structures and operate based on the same fundamental principles for transmitting neural signals. The core characteristics of the brain revolve around the interconnectedness of neurons and the process of transmitting electrochemical signals within these neurons. These fundamental aspects enable the brain to carry out intricate tasks.

Figure 1.1 shows a prototypical neuron. The central portion, known as the cell body or soma, houses the nucleus and various organelles crucial for all cells' functioning. Extending from the cell body, numerous root-like extensions called dendrites and a single tubular fiber known as the axon emerge. At the end of the axon, it branches into several smaller extensions. Dendrites are branch-like extensions originating from the neuron cell body. A typical neuron features multiple dendrites that exhibit extensive branching. The regions that handle signals, impulses, or information (synapses) are situated on the cell body and dendrites. In some neurons, you can find spines on the dendrites, which create specialized receiving sites. Given that one of the fundamental purposes of neurons is to integrate information received from other neurons, the number of inputs that each neuron receives is a crucial factor in shaping its neural function. The axon is an elongated, fiber-like extension stemming from the cell body. The primary role of the axon is to conduct signals, facilitating the transmission of impulses to other neurons or muscle fibers. The process of transmitting electrochemical signals through the axon is known as the action potential. Neurons employ two types of signalling mechanisms: electrical and chemical. Electrical signals predominate within the interior of neurons, while chemical signals are utilized at the synaptic terminals.

We will now elaborate on the process through which an electric current is generated across the axon membrane, how it propagates, and the various mathematical models used to describe it. The neuron membrane comprises two thin layers of lipid molecules. These layers separate the axon's cytoplasm (interior) from the extracellular fluid of the neuron. The membrane exhibits selectivity in allowing the diffusion of specific molecules, and this selectivity can vary over time and along the length of the axon. The selective permeability of membranes is primarily attributed to ion channels, which permit only specific types of ions to traverse the membrane following their concentration and electrochemical gradients. The process within the axon responsible for transmitting signals is called the action potential. Ion concentrations of K^+, Na^+, Cl^-, and Ca^{2+}, and the disparities in these concentrations across the axon membrane, are responsible for generating an electrical potential within the neurons. In the state of equilibrium, the potential difference in the neuron across the axon membrane is typically -70mV, referred to as the resting potential. In the inactive state of a neuron, the distribution of ions across the thin layers of the membrane results in the interior or cytoplasm of the neuron being negatively charged compared to the extracellular fluid. This phenomenon is driven by a biological ion pump that becomes active when the interior of the axon becomes more positively charged. Introducing an electric pulse into the neuron's axon disrupts the resting potential across the membrane. This disruption leads to alterations in ion concentrations across the membrane, resulting in changes to the potential difference. Hence, the axon's membrane exhibits the property of capacitance, involving the separation of charges. Its function is akin to a resistor-capacitor (RC)-type circuit, which will be further explored in the upcoming section. The membrane's permeability permits the entry of Na^+ ions into the neuron's interior via ion channels. This influx raises the voltage across the membrane, and when it surpasses the threshold, the injected current generates a single pulse that propagates through the axon terminals. The pulse signal travelling through the axon terminals comes to a halt at the synaptic ends, thanks to the synaptic gap. It is relayed to the target neuron from the synaptic endings through a unique chemical process known as synaptic transmission. During synaptic transmission, specific neurotransmitters are released from the synaptic ends and interact with the target neurons as they traverse the synaptic gap. Within the target neurons, the signal pulse functions as a current pulse, following the same mechanism as in the pre-synaptic neuron, to impact other neurons.

William James was the first to publish several facts related to brain structure and function [10]. He also stated the correlation between learning and associative memory. Half a century later, McCulloch and Pitts published one of the famous "neural network" papers in which he had derived theorems related to neuronal system models in the early 1940s. Then, Donald Hebb made a pivotal contribution to the journey of neural networks. The first successful neuro-computer was developed between 1957 and 1958 by Frank Rosenblatt, Charles Wightman, and others. Rosenblatt is known as the father of Neurocomputing. 'Neurodynamics' was the early book on Neurocomputing that he wrote. In 1987, the first open conference on neural networks in modern times, the IEEE International Conference on Neural Networks, was held in San Diego, and the International Neural Network Society (INNS) was formed.

The field of neural networks has grown since then. One can find a variety of journals and conferences related to neural networks. The neural networks are today applied in many fields, i.e., in voice recognition, forecasting stock prices [11], geotechnical engineering [12], decision making control in medical applications [13]. Therefore, it becomes an exciting area of research from an application point of view. Also, it will be shown that the differential equations express the neural network models. So, the mathematical analysis of these will interest mathematicians and related people.

1.2.1 Biological model

As previously discussed, neurons in the human brain convey electrical pulses along their axons to communicate with other neurons. Through this transmission of signals from one neuron to another within the network, the brain accomplishes specific tasks of interest. To better understand how the brain functions, it is essential to create mathematical models that represent the biological network of neurons. In this subsection, we will explore the general mathematical model of biological neural networks, which Robert L. Harvey initially developed in his book "Neural Network Principles," published in 1994. Assume that there are n neurons in the network, which is illustrated in Figure 1.2. These neurons are denoted as $v_1, v_2, ..., v_n$. Use a variable $x_i(t)$ to represent the state of the i-th neuron and a variable $Z_{ij}(t)$ to describe the coupling strength between neurons v_i and v_j.

Specifically,

$$x_i(t) = \text{the deviation of the } i\text{-th neuron from the resting potential.}$$

The variable $x_i(t)$ represents the activation level of the i-th neuron, often referred to as axon potential or short-term memory (STM) trace.

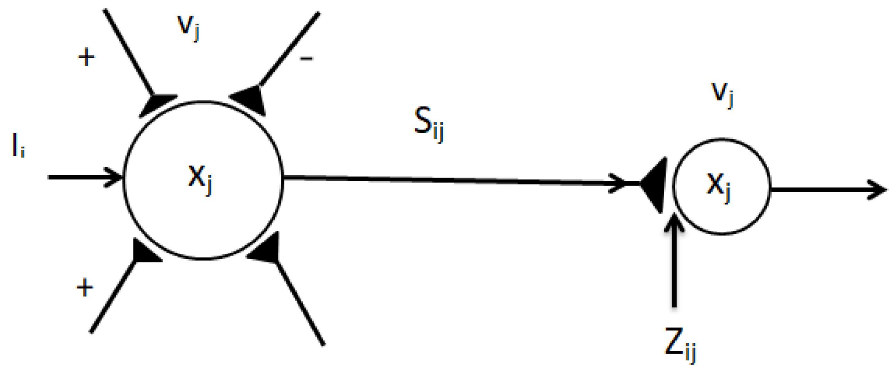


FIGURE 1.2: Diagram illustrating a neuron and a network: Neuron v_i has a potential $x_i(t)$ relative to equilibrium and sends a signal S_{ij} along its axon to a target neuron v_j. This signal impacts the target neuron with a coupling strength Z_{ij}, which can either be excitatory ($Z_{ij} > 0$) or inhibitory ($Z_{ij} < 0$). Furthermore, there could be additional inputs I_i to the neuron, representing external stimuli in the neuron model.

The interaction strength between neurons v_i and v_j is represented by the variable $Z_{ij}(t)$, which can be either negative or positive, depending on whether it signifies the inhibition or excitation of neurons to fire. When $Z_{ij}(t) > 0$, it indicates that the i-th neuron is being excited to send a signal to the j-th neuron. Conversely, if $Z_{ij}(t) < 0$, it implies that the i-th neuron is being inhibited from sending a signal to the j-th neuron.

$Z_{ij}(t)$ = the average release rate of neurotransmitter per unit axonal signal frequency.

This parameter is referred to as the synaptic coupling coefficient or long-term memory (LTM) trace. Assuming there is a deviation in a neuron's potential from equilibrium due to internal and external processes in the neural network, the rate of change in the neuron's potential is described by the following differential equation.

$$\frac{dx_i(t)}{dt} = \left(\frac{dx_i(t)}{dt}\right)_{external} + \left(\frac{dx_i(t)}{dt}\right)_{internal}, \quad \forall i. \tag{1.1}$$

Assume inputs from other neurons and stimuli are additive. Then, we have

$$\frac{dx_i(t)}{dt} = \left(\frac{dx_i(t)}{dt}\right)_{external} + \left(\frac{dx_i(t)}{dt}\right)_{excitatory} - \left(\frac{dx_i(t)}{dt}\right)_{inhibitory} + \left(\frac{dx_i(t)}{dt}\right)_{internal}, \qquad \forall i. \tag{1.2}$$

Furthermore, assuming that the neuron's potential is exponentially decaying toward its equilibrium state in the absence of external processes, we can express this as follows

$$\left(\frac{dx_i(t)}{dt}\right)_{internal} = -\alpha_i(x_i(t))x_i(t), \text{where} \quad \alpha_i(x_i(t)) > 0, \quad \forall i. \tag{1.3}$$

Let's assume that the additive synaptic excitation is directly proportional to the pulse train frequency.

$$\left(\tfrac{dx_i(t)}{dt}\right)_{excitatory} \propto \textstyle\sum_{other\ neurons}(\text{frequency of signal})(\text{synaptic coupling strengths}). \tag{1.4}$$

It can be written as

$$\left(\frac{dx_i(t)}{dt}\right)_{excitatory} = \sum_{\substack{l=1 \\ l \neq i}}^{n} S_{li}(t) Z_{li}(t), \forall i, \tag{1.5}$$

Here, $S_{li}(t)$ represents the average frequency of signals in the axon travelling from neuron v_l to v_i," evaluated at v_i. The average signal frequency $S_{li}(t)$ relies on both the propagation time delay τ_{li} for the signal to reach neuron v_i from v_l and the threshold value Γ_l for the firing of neuron v_l. This relationship can be expressed as follows

$$S_{li}(t) = f_l(x_l(t - \tau_{li}) - \Gamma_l), \tag{1.6}$$

here, $f_l : \mathbb{R} \to [0, \infty)$ represents a specific non-negative function known as the signal function. Various forms of signal functions are commonly employed in neural networks and will be explored in more detail later on.

Substituting (1.6) in equation (1.5), we get

$$\left(\frac{dx_i(t)}{dt}\right)_{excitatory} = \sum_{\substack{l=1 \\ l \neq i}}^{n} Z_{li}(t) f_l(x_l(t - \tau_{li}) - \Gamma_l), \forall i. \tag{1.7}$$

Let's consider that the inhibitory inputs from other neurons are hardwired, meaning the coupling strengths between the inhibited neurons remain constant. In this context, express the following

$$\left(\frac{dx_i(t)}{dt}\right)_{inhibitory} = \sum_{\substack{l=1 \\ l \neq i}}^{n} C_{li}, \forall i, \tag{1.8}$$

where $C_{li} = b_{li} h_l(x_l(t - \tau_{li}) - \Gamma_l)$, where $b_{li} \geq 0$ represents a constant coupling strength. The function h_l is a signal function. Typically, the threshold value Γ_l remains the same for every neuron in the network.

The stimuli refer to external sources that can modify the neuron's potential, and thus,

$$\left(\frac{dx_i(t)}{dt}\right)_{external} = I_i, \forall i. \tag{1.9}$$

Now, substituting the equations (1.3), (1.7), (1.8), and (1.9) in the equation (1.2), we get the so-called additive STM trace equation for $i = 1, 2, ..., n$.

$$\frac{dx_i(t)}{dt} = -\alpha_i(x_i(t))x_i(t) + \sum_{\substack{l=1 \\ l \neq i}}^{n} Z_{li}(t) f_l(x_l(t-\tau_{li}) - \Gamma_l) - \sum_{\substack{l=1 \\ l \neq i}}^{n} b_{li} h_l(x_l(t-\tau_{li}) - \Gamma_l) + I_i. \tag{1.10}$$

Since it has been assumed that the excitatory synaptic coupling varies with time, the following equation based on Hebb's law can be derived

$$\frac{dZ_{ij}(t)}{dt} = -A_{ij}(Z_{ij}(t))Z_{ij}(t) + P_{ij}(t)[x_j(t)]^+, \quad A_{ij}(Z_{ij}(t)) > 0, \forall i, j, \tag{1.11}$$

where $P_{ij}(t) = \beta_{ij} f_i(x_i(t-\tau_{ij}) - \Gamma_i)$, $\beta_{ij} \geq 0$, and

$$[x_j(t)]^+ = \begin{cases} x_j(t), & \text{if } x_j \geq 0, \\ 0, & \text{if } x_j < 0. \end{cases}$$

The second term in equation (1.11) indicates that in order to increase $Z_{ij}(t)$, neuron v_i must transmit a signal $P_{ij}(t)$ to neuron v_j," while simultaneously, neuron v_j must be activated, meaning $x_j(t) > 0$.

It's important to note that the equations for short-term memory (STM) and long-term memory (LTM) traces, as expressed in equations (1.10) and (1.11) respectively, cannot be solved without the knowledge of the coefficients $\alpha_i, A_{ij}, C_{li}, S_{li}, P_{ij}$ as well as the external stimuli I_i.

1.3 Artificial Neural Network

An artificial neural network (ANN) is a computational model created to imitate the human brain's cognitive processes while executing a certain activity or function. These networks are typically constructed using electronic components or simulated within the software on a digital computer. Numerous designers of artificial intelligence systems have drawn inspiration from experimental results related to the brain's analog responses. They have constructed machines that mimic brain regions or functions, wherein nodes represent neurons or groups of neurons and connections between these nodes simulate the synapses of real neurons. The definition of an artificial neural network, as provided by Simon Haykin in his book [14] "Neural Network: A Comprehensive Foundation," given as follows

Definition 1.3.1. *A neural network is a massively parallel distributed processor comprising simple processing units characterized by its inherent ability to acquire experiential knowledge and make it accessible for various tasks. It shares similarities with the human brain in two key aspects*

(i) Neural networks acquire knowledge from their environment through a learning process.

(ii) The acquired knowledge is stored in the form of interconnection strengths, referred to as synaptic weights.

The following definition, derived from the DARPA study in 1988 [15], is described as

Definition 1.3.2. *A neural network is a system composed of numerous simple processing elements that work in parallel. Its functionality is dictated by the network's* *structure, connection strengths, and the processing carried out at these computational elements. Neural network architectures draw inspiration from the design of biological nervous systems, which operate in parallel to achieve high computational efficiency. The neural network's exceptional computing power is primarily attributed to its massively parallel distributed structure and its capacity to learn and, as a result, generalize. Generalization signifies the neural network's capability to generate sensible outputs for inputs that were not encountered during the learning phase. A neural network excels at solving complex problems through an integrated approach, and its components are not effective when operating individually.*

Parameters	Artificial neural networks	Biological neural networks
Structure	input hidden layer weight output	dendrites synapse axon cell body
Processor	complex high speed one or a few	simple low speed large number
Memory	separate from a processor localized non-content addressable	integrated into processor distributed content-addressable
Computing	centralized sequential stored programs	distributed parallel self-learning

TABLE 1.1: Comparison between artificial and biological neural network.

A fundamental diagram of an artificial neuron is given in Figure 1.3. The artificial neuron is composed of seven essential components, which include

(a) The input signals $(x_1, ..., x_p, .., x_n)$ are the data or samples derived from the external environment, and they represent the values associated with specific variables in a given application. Normalizing the input signals to improve the computational efficiency of learning algorithms is a common practice.

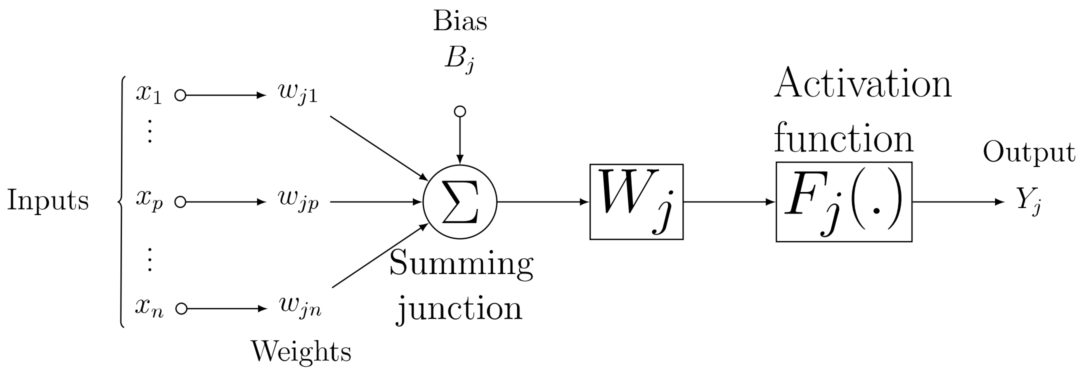


FIGURE 1.3: Fundamental diagram of an artificial neuron of the network.

(b) The synaptic weights $(w_{j1}, ..., w_{jp}, ..., w_{jn})$ are numerical values assigned to each of the input variables. These weights quantify the importance or relevance of each input variable in relation to the neuron's functionality.

(c) The linear aggregator $(\sum)$ combines all the input signals, each weighted by its corresponding synaptic weight, to generate an activation voltage.

(d) The activation threshold or bias (B_j) is a variable employed to define the appropriate threshold that the result generated by the linear aggregator should meet to produce a trigger value for the neuron's output.

(e) The activation potential (W_j) is the outcome of the difference between the linear aggregator and the activation threshold. If this value is positive, meaning $W_j > B_j$, the neuron generates an excitatory potential; otherwise, it will result in an inhibitory potential.

(f) The activation function (F_j) is designed to restrict the neuron's output within a reasonable range of values corresponding to the neuron's intended functional behaviour.

(g) The output signal (Y_j) is the ultimate value generated by the neuron based on a specific set of input signals. This output signal can also serve as input for other sequentially interconnected neurons.

The input signals, denoted as $x_1, ..., x_p, ..., x_n$, are connected to the j-th neuron via synaptic weights, represented as $w_{j1}, ..., w_{jp}, ..., w_{jn}$. These synaptic weights measure the strength of connections, and their positive or negative sign depends on whether the signal is excitatory or inhibitory, respectively. It's important to note the notation used for the subscripts in synaptic weights, such as w_{jp}. The first subscript indicates the neuron that receives the signals, and the second subscript refers to the input terminals of the synapse to which the weight is associated. Unlike biological neurons, the synaptic weights in artificial neural networks can take values within intervals of real numbers. The summing junction performs as a linear combiner of inputs, where the input signal x_p is multiplied by its corresponding synaptic weight w_{jp}. The bias B_j represents an external stimulus that can either decrease or increase the input to the activation function $F_j(.)$, depending on whether the bias is negative or positive, respectively. The following two expressions summarize the outcome generated by the artificial neuron as proposed by McCulloch and Pitts [16], in mathematical expression for the linear combination of input signals is as follows.

$$W_k = \sum_{p=1}^{n} x_p w_{jp} + B_j. \tag{1.12}$$

The output of the linear combiner u_k in equation (1.12) serves as an input value or an induced local field for the activation function. An activation function's primary role is to achieve a neuron's desired output. It is often referred to as a "squashing function" because it compresses the amplitude of the neuron's output to a finite range.

In Figure 1.3, the output of a neuron, following the application of the activation function, is as follows.

$$Y_j = F_j(W_j) = F_k(\sum_{p=1}^{n} x_p w_{jp} + B_j). \tag{1.13}$$

Hence, the operation of an artificial neuron can be summarized through the following steps

(i) Present a set of values to the neuron, representing the input variables.

(ii) The multiplication of each input of the neuron by its corresponding synaptic weight is carried out.

(iii) Obtain the activation potential by calculating the weighted sum of the input signals and then subtracting the activation threshold.

(iv) The implementation of an appropriate activation function is necessary to restrict a neuron's output.

(v) Compute the output by applying the neural activation function to the activation potential.

1.3.1 Activation function

Activation functions are fundamental in shaping neural networks' behaviour and learning capacity. They introduce non-linearity, allowing networks to model complex relationships in data, and their choice can significantly impact the training and performance of the network. It is applied to the weighted sum of inputs at each neuron (or node) in a neural network layer and helps the network learn complex patterns and representations. Activation functions are essential for the network's ability to model and approximate a wide range of functions.

Here are some Common Activation Functions are

(a) **Sigmoid function.** The sigmoid function has the following properties It is strictly increasing, meaning that for any two real numbers x and y, where $x < y,\ F(x) < F(y).$, it is smooth; that is, its derivative is continuous for all real numbers, it is bounded, and its range is the interval $[0, 1]$. A logistic function is one of the examples of a sigmoid function, defined by

$$F(u) = \frac{1}{1 + exp(-\beta u)}, \tag{1.14}$$

The symbol β denotes the amplification factor, which serves as the slope parameter for the curve. By manipulating the parameter β, it is possible to observe variations in the slopes of the curve, as depicted in Figure 1.4(a). It is evident from the mathematical expression in equation (1.14) that the sigmoid function tends to approach the values of 1 and 0 when the parameter β approaches positive infinity and negative infinity, respectively. The sigmoid function converges to a threshold function as the slope parameter β tends towards infinity. It is worth noting that a threshold function is not smooth, whereas a sigmoid function is a smooth function. The smoothness of the sigmoid function is an important property from an application perspective. Let's consider random variables for the firing threshold of an all-or-non neuron with a Gaussian normal distribution function. The expected output signal's value is a sigmoid function of activity [17]. The model of this type of neuron is called the stochastic model. For this and other reasons, a sigmoid activation function has become increasingly popular in designing artificial neural network models. Other popular examples of sigmoid activation function are the inverse tangent function and tangent hyperbolic function.

(b) **hyperbolic tangent activation function** The resulting output, in contrast to the logistic function, will consistently take on real values within the range of -1 to 1, as described by the following mathematical expression

$$F(u) = \tanh(.), \tag{1.15}$$

(c) **Piecewise linear function.** The following function defines the piecewise linear function shown in Figure 1.4(c).

$$F(u) = \begin{cases} 0, & \text{if } u \leq -\frac{1}{\beta}, \\ u + \frac{1}{\beta}, & \text{if } -\frac{1}{\beta} < u < \frac{1}{\beta}, \\ 1, & \text{if } \ u \geq \frac{1}{\beta}, \end{cases} \tag{1.16}$$

where β is called the amplification factor or the neural gain. Such a type of activation function has been widely implemented in cellular neural networks [18, 19]. The piecewise linear function plotted in Figure 1.4(c) is for $\beta = 2$. It can be observed from the function 1.16 that if the amplification factor β approaches infinity, then the piecewise linear function reduces to a threshold function.

(d) **Threshold function (step function).** For the activation function depicted in Figure 1.4(d), the output of a neuron in Figure 1.1 is determined by the following expression

$$Y_k = F_k(u_k) = \begin{cases} 1 & \text{if } u_k \geq 0, \\ \\ 0 & \text{if } u_k < 0. \end{cases} \tag{1.17}$$

This particular activation function is commonly referred to as the McCulloch-Pitts model [16], named in recognition of the pioneering work conducted by McCulloch and Pitts in 1943. In the McCulloch-Pitts model, the fundamental idea is that a neuron emits a signal indicating "yes" (represented by "1") if the induced local field of that neuron is non-negative; otherwise, it outputs "0". The activation function succinctly captures the all-or-none property inherent in the McCulloch-Pitts model, where the neuron either fully activates or remains inactive based on the sign of the induced local field.

(e) **RELU function** RELU stands for rectified linear unit. This nonlinear activation function is widely used in neural networks. It can be defined mathematically as $F(x) = max(0, x)$. The advantage of using the RELU function is faster training speed and decreased saturation problems. The updated versions of the RELU functions include the leaky RELU, parametric RELU, and randomized RELU. These functions overcome the property of the RELU function of retaining very little or very high information in the storage, resulting in an explosion or the dealt of the neural network [20].

In summary, activation functions are fundamental in shaping neural networks' behaviour and learning capacity. They introduce non-linearity, allowing networks to model complex relationships in data, and their choice can significantly impact the training and performance of the network. This can be done by adopting a nonlinear activation function. Although predefined activation functions are used, the research on the trainable activation function is on the way [21]. The predefined activation functions fall under the category of fixed-shape activation functions.

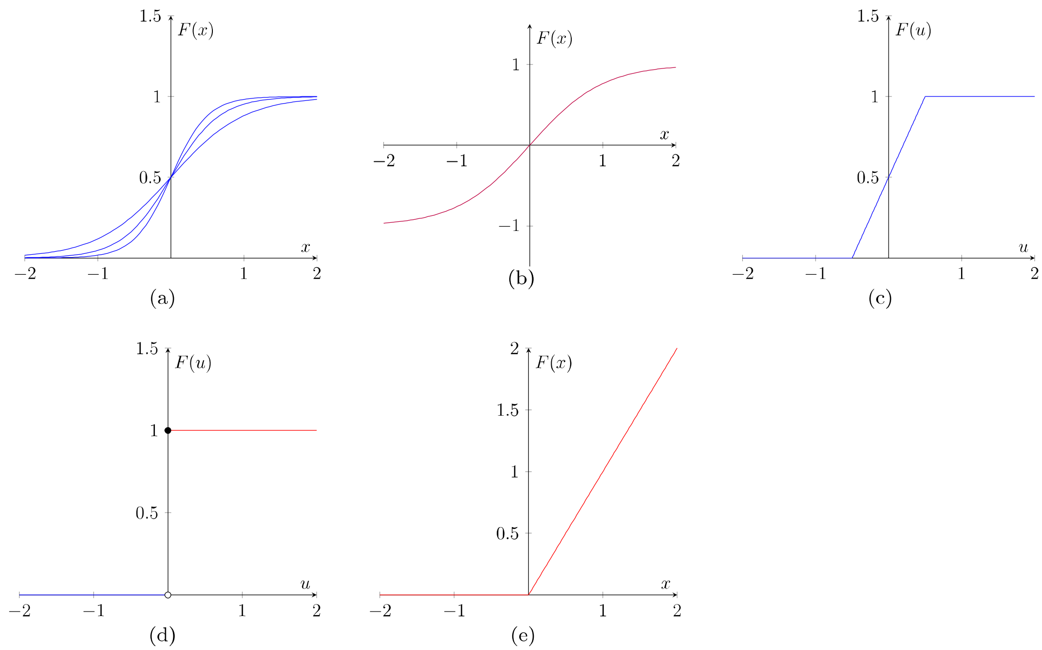


FIGURE 1.4: (a) Sigmoidal function, (b) tanh function, (c) Piecewise activation function, (d) Threshold function, (e) RELU function

1.3.2 Major Architectures of Artificial Neural Networks.

(a) **Input Layer.** This layer is tasked with receiving information, which can be data, signals, features, or measurements, from the external environment. The inputs (samples or patterns) are often normalized within the activation function's limit values. This normalization enhances numerical precision for the mathematical operations conducted by the network.

(b) **Intermediate Layers.** These layers consist of neurons that are responsible for extracting patterns related to the process or system under analysis. They carry out the majority of the internal processing within a network.

(b) **Output Layer.** This layer, also consisting of neurons, is responsible for generating and presenting the final outputs of the network. These outputs result from the processing performed by the neurons in the preceding layers.

The primary architectures of artificial neural networks, taking into account the arrangement of neurons, their interconnections, and the composition of layers, can be categorized as follows: (i) single-layer feedforward artificial neural networks, (ii) multi-layer feedforward artificial neural networks, (iii) recurrent artificial neural networks, and (iv) mesh artificial neural networks.

1.3.3 Architecture of a Single-Layer Feedforward Network

The single-layer feedforward architecture of an artificial neural network consists of only one input layer and a single neural layer, which also serves as the output layer. In this architecture, illustrated in Figure 1.5, there are n inputs and m outputs. The flow of information is unidirectional, always moving from the input layer to the output layer. As depicted in Figure 1.5, the number of network outputs corresponds to the number of neurons. Following this architecture, networks are commonly used in pattern classification and linear filtering problems.

1.4 Architectures of Multi-Layer Feedforward Networks

In contrast to single-layer network architecture, multi-layer feedforward networks consist of one or more hidden neural layers see Figure 1.6. These networks are

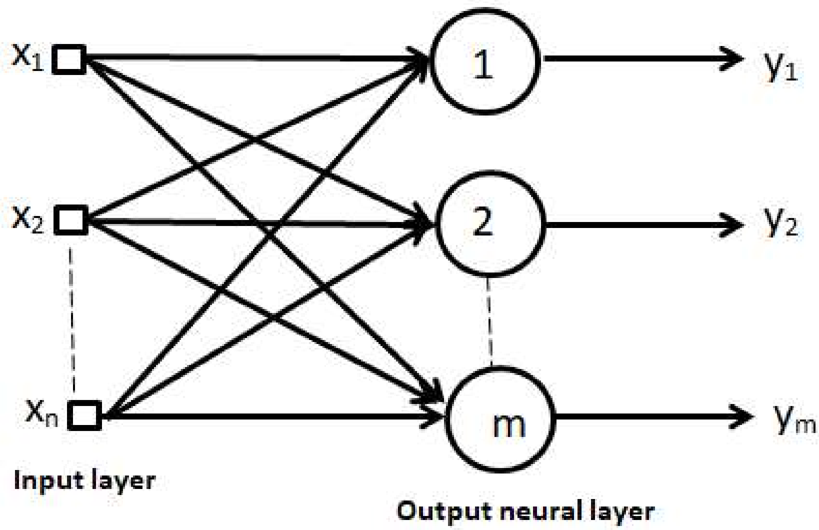


FIGURE 1.5: Single-layer feedforward network

utilized to address a variety of problems, including function approximation, pattern classification, system identification, process control, optimization, robotics, and more. Figure 1.6 depicts a multi-layer feedforward network with the following composition: one input layer containing n sample signals, two hidden neural layers comprising n_1 and n_2 neurons, and a final output neural layer consisting of m neurons representing the corresponding output values of the analyzed problem. From Figure 1.6, it is evident that the number of neurons in the first hidden layer is typically different from the number of signals in the network's input layer. The number of hidden layers and their respective number of neurons depend on the nature and complexity of the problem being addressed by the network and the quantity and quality of available data related to the problem. However, similar to single-layer feedforward networks, the number of output signals always matches the number of neurons in the respective output layer.

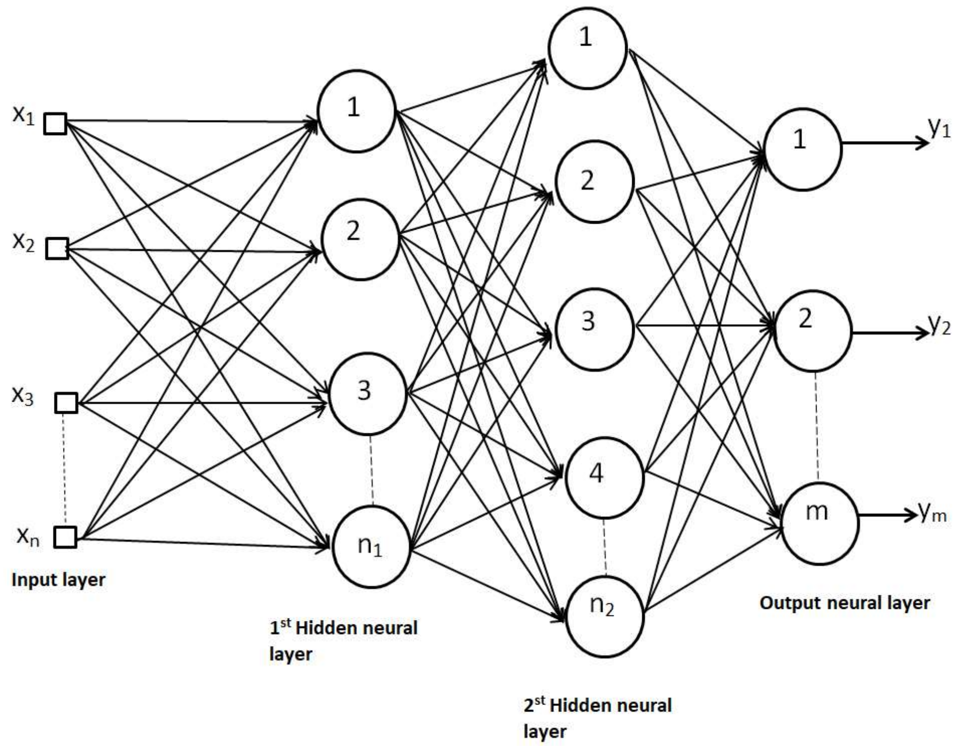


FIGURE 1.6: Multi-layer feedforward network

1.4.1 Architecture of Recurrent or Feedback Networks

In recurrent or feedback networks, the outputs of the neurons serve as feedback inputs for other neurons. The incorporation of feedback makes these networks suitable for dynamic information processing, allowing them to be applied to time-variant systems. Examples of applications include time series prediction, system identification, optimization, process control, and more. Among the primary feedback networks are the Hopfield and Perceptron, with feedback between neurons from distinct layers. The learning algorithms employed in their training processes are based on energy function minimization and the generalized delta rule. Figure 1.7 illustrates an example of a Perceptron network with feedback, where one of its output signals is fed back to the middle layer. Through the feedback process, networks with this architecture generate current outputs while also considering the previous output values.

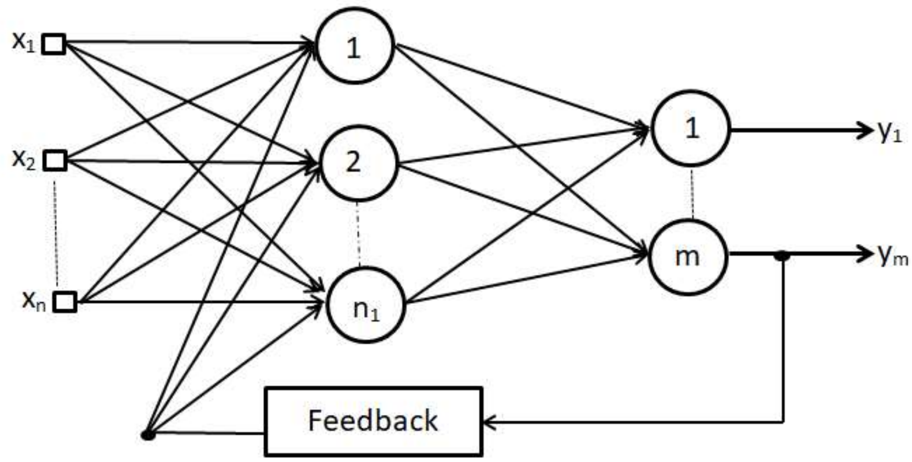


FIGURE 1.7: Recurrent feedforward network

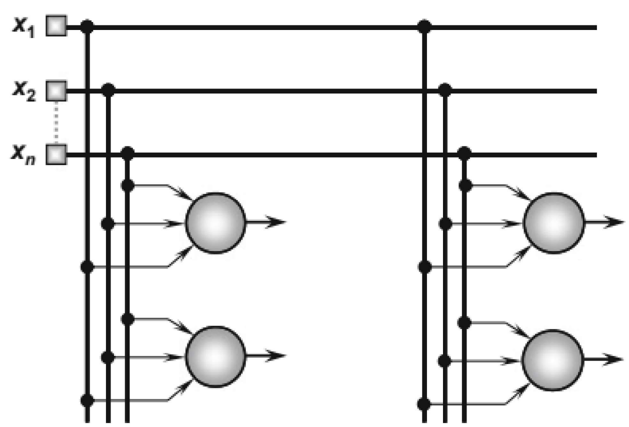


FIGURE 1.8: Structure of a Mesh network

1.4.2 Mesh Architectures

The distinctive characteristic of networks with mesh structures lies in their consideration of the spatial arrangement of neurons for pattern extraction purposes. In these networks, the spatial localization of neurons is directly linked to the adjustment of their synaptic weights and thresholds. Mesh architectures find applications in various domains, including data clustering, pattern recognition, system optimization, and graph-related tasks.

Based on the preceding discussions, it can be concluded that the following crucial factors influence the performance of a neural network.

(1) External inputs.

(2) Internal decay rates.

(3) Synaptic weights.

(4) Activation functions.

(5) Propagation delays.

(6) Connections topology/Network architecture.

In deterministic models, a network with a particular connection topology operates dynamically. Considering initial activation levels and synaptic coupling coefficients and treating other relevant factors as parameters makes it feasible to compute the subsequent activation levels and synaptic coupling coefficients. This phenomenon is referred to as joint activation-weight dynamics. However, in practical applications, there is often a separation between activation dynamics and weight dynamics. Several methods are available to dynamically decide the synaptic weights of a network to achieve certain objectives, such as pattern recognition or generating desired network outputs from a given set of inputs. Typically, these schemes involve defining a discrete dynamical system (a system of difference equations) or a continuous dynamical system (a system of differential equations) within the space of matrices representing synaptic coupling coefficients. This aspect is commonly known as weight dynamics. After establishing the connection topology and synaptic weights, along with treating other factors as parameters, predicting future activation levels entails specifying the initial activation levels.

Now, this completes the discussions about the basic properties of artificial neural networks and proceeds further to delve into the introduction of famous neural network models such as the Hopfield and Cohen-Grossberg models. These are important for the current thesis. In order to introduce the Hopfield model, the first thing is to understand the RC circuit.

1.4.3 Basics of electrical circuit

The study of electrical circuits is crucial for comprehending the mechanisms involved in signal processing among neurons in the human brain. As previously explained, a biological neuron's membrane functions like a capacitor, creating a separation of ion concentrations. This charge separation leads to the generation of an electric pulse that travels along the neuron's axon to influence the target neuron. A resistor-capacitor (RC) circuit can effectively model this behaviour, as depicted in Figure 1.9. In the electrical circuit depicted in Figure 1.9, various components, including a

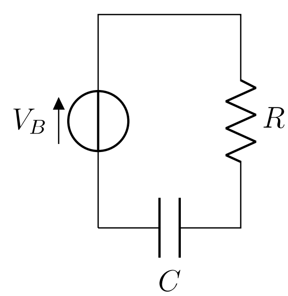


FIGURE 1.9: RC-circuit with a source of voltage.

resistor (R), a capacitor (C), and a voltage source (battery), are arranged in series. This configuration serves as a model for understanding the dynamics of electrical signal transmission. Specifically, the battery in the circuit represents the force arising from differences in ion concentrations inside and outside the cell membrane. Analogously, the resistor simulates ion channels, dictating the membrane's permeability to specific ions. The modeling of this circuit relies on Kirchhoff's laws, which govern the conservation of current and voltage in electrical circuits.

(a) *Kirchhoff's Current Law:* The total current entering a junction is equal to the total current leaving the junction.

(b) *Kirchhoff's Voltage Law:* The algebraic sum of the voltage differences around any closed loop in a circuit is zero.

According to Kirchhoff's voltage law, the voltage drop across the capacitor is equal to the algebraic sum of the voltage drop across the battery and the voltage drop across the resistor.

$$V_C = V_B + I_R R, \tag{1.18}$$

whereas V_C represents the voltage across the capacitor, V_B signifies the voltage across the battery, and I_R denotes the current flowing through the resistor, as defined by Ohm's law ($V_R = I_R R$). Thus, we have

$$I_R(t) = \frac{V_C(t) - V_B(t)}{R}. \tag{1.19}$$

The neuron receives an external current through the synapses, denoted as I_{ext}. According to Kirchhoff's current law, this can be expressed as follows

$$I_{ext.} = I_C + I_R. \tag{1.20}$$

Equations 1.19 and 1.20 lead to the following expression

$$I_{ext} = C\frac{dV_C}{dt} + \frac{V_C(t) - V_B(t)}{R}. \tag{1.21}$$

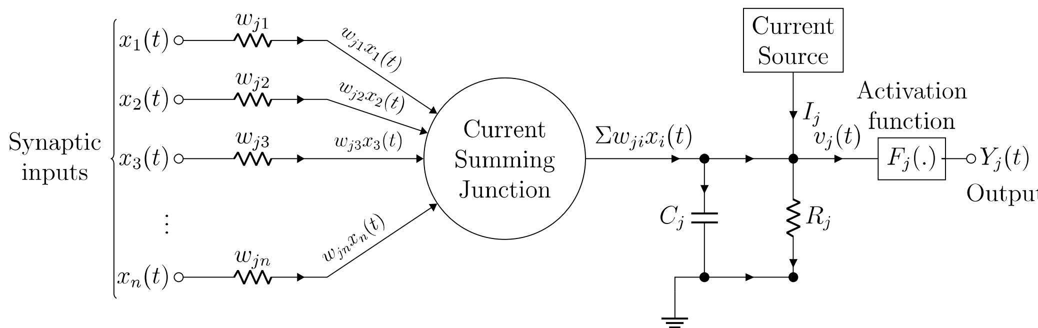


FIGURE 1.10: Additive model of a neuron.

1.4.4 Additive model

Inspired by the behaviour of a biological neuron modeled as an RC circuit, the model (1.13) can be extended to accommodate the temporal nature of input data. In this context, where inputs to the neuron vary over time, one approach to consider the time-varying inputs is depicted in Figure 1.10. Here, synaptic weights $w_{j1}, w_{j2}, ..., w_{jn}$ are represented by conductance (reciprocal of resistance), and the corresponding inputs $x_1(t), x_2(t), ..., x_n(t)$ are represented by potentials (voltages). The products of inputs and their respective synaptic weights are aggregated at the current summing junction, characterized by low input resistance, unity current gain, and high output resistance.

The sum of the current flowing to the input node of the activation function $F_j(.)$ is

$$\sum_{i=1}^{n} w_{ji}x_i(t) + I_j, \tag{1.22}$$

the initial term arises from the input signals of the j-th neuron, while the subsequent term results from an external current source applied as a bias to the neuron. If we denote the induced local field at the input node of the activation function as $v_j(t)$, the overall current departing from the input node of the activation function is expressed as

$$\frac{v_j(t)}{R_j} + C_j\frac{dv_j(t)}{dt}, \tag{1.23}$$

Whereas the initial term represents the current passing through the resistor R_j, while the subsequent term represents the current arising from the potential drop across the capacitor C_j. In accordance with Kirchhoff's current law, the collective current traversing the input node of the activation function in Figure 1.10 is zero. Consequently, deduce the following from equations (1.22) and (1.23).

$$\frac{v_j(t)}{R_j} + C_j\frac{dv_j(t)}{dt} = \sum_{i=1}^{n} w_{ji}x_i(t) + I_j. \tag{1.24}$$

The output of neuron j is determined by the activation function, given the induced input field $v_j(t)$, and is expressed as

$$Y_j(t) = F_j(v_j(t)). \tag{1.25}$$

Typically, in the additive model, the activation function is selected to be bounded and differentiable, exhibiting asymptotic behaviour such as the logistic function illustrated in Figure 1.4(a):

$$F_j(v_j) = \frac{1}{1 + \exp(-v_j)}. \tag{1.26}$$

The differential equation (1.24) characterizes a neuron model referred to as an additive model. The term 'additive' is employed to distinguish it from multiplicative (shunting) models, which incorporate state-dependent synaptic weights [22].

1.4.5 Hopfield neural network

Let's examine a fully connected recurrent network with n neurons, each having a structure like the one depicted in Figure 1.10. The fundamental interconnection diagram is illustrated in Figure 1.11, where each neuron feeds back its output, via a unit delay element, to the inputs of the other neurons through feedback loops. Specifically, there are no self-feedback loops in the network. Therefore, it is possible to analyze the network dynamics by taking into account the instantaneous propagation of signals between neurons and Equation (1.23). We have

$$C_j \frac{dv_j(t)}{dt} = -\frac{v_j(t)}{R_j} + \sum_{i=1}^{n} w_{ji} F_i(v_i(t)) + I_j, \text{for } j = 1, 2, ..., n. \tag{1.27}$$

Here, $x_i(t) = F_i(v_i(t))$, implying that each neuron possesses its own activation function. Equation (1.27) represents the Hopfield neural network without time delay. The Hopfield model is commonly formulated in two forms: discrete and continuous flow. The solution of the differential equation (1.27) corresponds to the continuous flow of the Hopfield neural network. The discrete flow of the Hopfield network follows the McCulloch-Pitts model, where the state of each neuron takes on values of -1 or 1 depending on the signs of the induced local field. Specifically, $x_i(t) = -1$ if $v_i(t) > 0$, and $x_i(t) = 1$ if $v_i(t) < 0$, for all i. This thesis primarily focuses on the continuous flow of neural networks rather than the discrete one.

The Hopfield neural network has gained significant attention in applications, particularly as an associative memory. Associative memory is a form of content-addressable memory where the fixed points of the network store patterns in memory. The primary role of the network is to retrieve a pattern stored in memory based on incomplete or noisy information about that pattern. Therefore, a crucial feature of content addressable memory is the ability to retrieve the stored pattern with the assistance

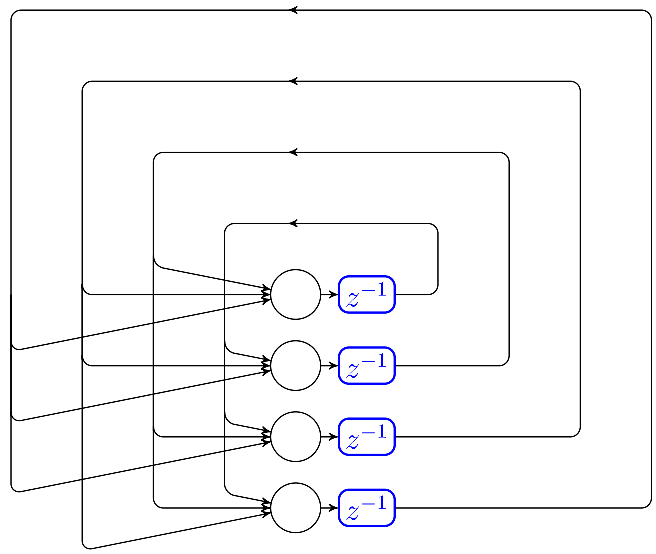


FIGURE 1.11: An architecture of Hopfield neural network consisting of $n = 4$ neurons.

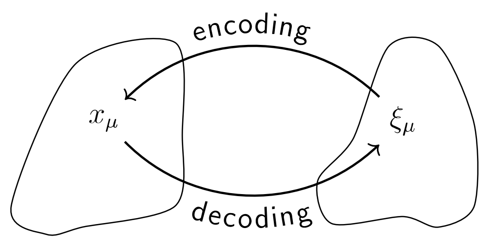


FIGURE 1.12: Encoding-Decoding illustration between the space of fundamental memories ξ_μ and the space of stored vectors x_μ.

of a reasonable subpart or subset of the information related to that pattern [23, 24]. Mathematically, the core concept of content addressable memory involves mapping a fundamental memory ξ_μ to a fixed (stable) point x_μ within the network. From right to left, the arrow in Figure 1.12 illustrates a fundamental memory's encoding process onto a fixed (stable) point. Conversely, the left-to-right arrow describes the decoding process used to retrieve the stored memory. Suppose now that the network contains a pattern with partial but sufficient information about one of the fundamental memories. This partial information of the pattern serves as an initial state of the flow, residing in the basin of attraction of the stable fixed point. In other words, from a mathematical perspective, the question is to determine the conditions under which the network's flow approaches the fixed point in response to arbitrary initial data. In the context of content addressable memory, the number of fixed points in the neural network is also a crucial consideration in applications. A greater number of fixed points indicates a higher storage capacity in the network.

The dynamics of the Hopfield network were studied by John Hopfield in his paper [25], published in 1984. He defined an energy function for the network by assuming the following conditions.

(i) The synaptic weights matrix is symmetric, denoted as $w_{ji} = w_{ij}$ for all j, i.

(ii) An inverse of the activation function (1.26) exists, allowing us to express v as $F_j^{-1}(x)$.

Based on the equation (1.26), we have

$$F_j^{-1}(x) = -\ln\left(\frac{1-x}{1+x}\right). \tag{1.28}$$

The Lyapunov function for the Hopefield model (1.27) is given as

$$V = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_i(t)x_j(t) + \sum_{j=1}^{n}\frac{1}{R_j}\int_0^{x_j} F_j^{-1}(j)dx - \sum_{j=1}^{n} I_j x_j. \tag{1.29}$$

The term Lyapunov function will be stated later. The Lyapunov function specified in equation (1.29) can exhibit multiple minima points, each corresponding to stable fixed points of the Hopfield model (1.27). The network's dynamics aim to identify and converge to these minima. Therefore, by differentiating V with respect to time and employing equations (1.27) and (1.28), we get

$$\frac{dV}{dt} = -\sum_{j=1}^{n} C_j\left(\frac{dx_j}{dt}\right)^2\left[\frac{dF_j^{-1}(x_j)}{dx_j}\right]. \tag{1.30}$$

Observing from the equation (1.28), it is evident that the inverse activation function is a monotonically non-decreasing function of the output. Consequently, it follows that

$$\frac{dF_j^{-1}(x_j)}{dx_j} \geq 0 \; \forall \; x_j. \tag{1.31}$$

Thus, deduce from equation (1.30) that

$$\frac{dV}{dt} \leq 0. \tag{1.32}$$

In accordance with Lyapunov's method of stability, the inequality (1.32) suggests that the continuous evolution of the Hopfield neural network, as described by equation (1.27), follows a trajectory. This trajectory aims to converge towards the minima of the function V, ultimately coming to a halt at these fixed points. It is observed that the inequality (1.32) equals zero only if

$$\frac{dx_j}{dt} = 0 \text{ for all } j. \tag{1.33}$$

Therefore, express the following

$$\frac{dV}{dt} < 0 \text{ except at fixed points.} \tag{1.34}$$

Hence, the trajectory of the continuous Hopfield neural network asymptotically converges to the fixed points.

1.4.6 Cohen-Grossberg neural network

One widely recognized neural network is the Cohen-Grossberg neural network, often regarded as a generalized version of the Hopfield network. Cohen and Grossberg, in a 1983 paper [26], outlined a fundamental principle for crafting content-addressable memory networks. They demonstrated that these models can be formulated in the following manner

$$\frac{dx_i(t)}{dt} = a_i(x_i(t))\big[b_i(x_i) - \sum_{j=1}^{n} w_{ij}F_j(x_j)\big], \text{ for all } i = 1, 2, ..., n. \tag{1.35}$$

In the equation (1.35), where $x_i(t)$ represents the activation state of the $i-$th neuron, $a_i(x_i(t))$ denotes the amplification function, $b_i(x_i)$ stands for the self-signal function, and $F_j(x_j)$ signifies the standard signal function. Notably, from this equation, it can be observed that the rate of change in the neuron's activity diminishes only if the net input to the neuron surpasses a certain intrinsic function b_i of its activity and $a_i(x_i(t))$ is positive. The nonlinear system (1.35) possesses the Lyapunov function

$$V(x) = -\sum_{i=1}^{n} \int_0^{x_i} b_i(\xi_i)F_i^{'}(\xi_i)d\xi_i + \frac{1}{2}\sum_{j,l=1}^{n} w_{jl}F_j(x_j)F_l(x_l). \tag{1.36}$$

If the synaptic coefficients w_{ik}, as well as the other components a_i, b_i, and F_j of the system, meet the following requirements.

(i) *Symmetric*: $w_{ij} = w_{ji}$;

(ii) *Continuity*: $a_i(x)$ and $b_i(x)$ are continuous for $x > 0$;

(iii) *Positivity*: $a_i(x) > 0$ if $x > 0$;

(iv) *Monotonicity*: $F_i(x)$ is continuously differentiable, and $F_i^{'}(x) \geq 0$ for $x \geq 0$.

Following from integration of V, one have

$$\frac{dV}{dt} = -\sum_{i=1}^{n} a_i F_i^{'} \big[b_i - \sum_{j=1}^{n} w_{ij} F_j \big]^2. \tag{1.37}$$

Considering (iii) and (iv), we have $\frac{dV}{dt} \leq 0$ along the trajectories. In other words, the trajectories of the system (1.35) globally converge to the equilibrium points. Cohen and Grossberg, in their publication [27], pointed out that the differential equation (1.35) can represent the additive model (1.26) by employing coefficients with the standard electrical circuit interpretation as

$$a_i(x_i(t)) = \frac{1}{C_i}, \tag{1.38}$$

$$b_i(x_i(t)) = -\frac{1}{R_i} + I_i, \tag{1.39}$$

$$w_{ij} = -w_{ij}. \tag{1.40}$$

In the additive case, the amplification function is a positive constant, thus satisfying positivity, and the self-signal function is linear. Substitute equations (1.38), (1.39), and (1.40) into the Lyapunov function (1.36), we obtain

$$V = -\frac{1}{2} \sum_{l,j=1}^{n} w_{lj} F_l(x_l) F_j(x_j) + \sum_{i=1}^{n} \frac{1}{R_i} \int_0^{x_i} \xi_i F_i^{'}(\xi_i) d\xi_i - \sum_{i=1}^{n} I_i F_i(x_i). \tag{1.41}$$

In the Hopfield energy function, it was assumed that the activation function should be invertible. However, Cohen and Grossberg considered the activation function to be non-decreasing. Hopfield's work was published in 1984, one year after Cohen and Grossberg's publication. Despite this chronological order, physicists and engineers commonly referred to the additive model as the Hopfield additive model in their literature. As a response, Stephen Grossberg published a review paper [28] on his works, illustrating how the additive model is a special case of the work published in collaboration with Michael A. Cohen [27].

1.5 Glimpse of Some Useful Mathematical Concepts

This section will provide a glimpse of the mathematical concepts employed in the chapters of the present thesis. The primary focus will be stability theory, Lyapunov stability theory, delayed differential equations, Inertial systems, and matrix measure theory.

1.5.1 Stability Theory

Let us consider the autonomous system

$$\dot{x} = f(x) \tag{1.42}$$

where $f : D \to \mathbf{R^n}$ is a locally Lipschitz map from a domain $D \subset \mathbf{R^n}$ into $\mathbf{R^n}$. Suppose $\bar{x}_e \in D$ is an equilibrium point of (1.42); that is,

$$f(\bar{x}_e) = 0$$

Without loss of generality and for convenience, all definitions and theorems specifically for the scenario when the equilibrium point is located at the origin of $\mathbf{R^n}$, denoted as $\bar{x} = 0$. If we have another equilibrium point, then by use of variable transformation, shift that equilibrium point to the origin. Suppose $\bar{x} \neq 0$, and consider the change of variables $y = x - \bar{x}$. The derivative of y is given by

$$\dot{y} = \dot{x} = f(x) = f(y + \bar{x}) \overset{\text{def}}{=} g(y), \quad \text{where } g(0) = 0$$

The system exhibits equilibrium at the origin in the newly defined variable, denoted as y. Henceforth, to ensure the applicability of analysis, consistently assume that $f(x)$ follows the condition $f(0) = 0$ and proceed to examine the stability of the origin at $x = 0$.

Definition 1.5.1. *The equilibrium point $x = 0$ of system (1.42) is*

(i) ***Stable****: If, for every $\epsilon > 0$, there exists a $\delta > 0$ such that, if $\|x(0) - x_e\| < \delta$, then for every $t \geq 0$ we have $\|x(t) - x_e\| < \epsilon$.*

(ii) ***Asymptotically Stable****: If it is stable and there exists $\delta > 0$ such that if $\|x(0) - x_e\| < \delta$, then $\lim_{t\to\infty} \|x(t) - x_e\| = 0$.*

(iii) ***Exponential Stable****: The equilibrium of the above system is said to be exponentially stable if it is asymptotically stable and there exists $\alpha > 0, \beta > 0, \delta > 0$ such that if $\|x(0) - x_e\| = M < \delta$, then $\|x(t) - x_e\| \leq Me^{-\beta t}$, for all $t \geq 0$.*

(iv) ***Unstable*** *If it is not stable.*

The stability requirement, expressed in the $\epsilon - \delta$ framework, presents itself as a challenge and response. To establish the stability of the origin, it is necessary to show that for any chosen ϵ, there exists a corresponding δ, which may be dependent on ϵ. This δ ensures that a trajectory originating within a δ-neighborhood of the origin will remain within the ϵ-neighborhood throughout its course.

1.5.2 Lyapunov Stability

In 1892, Lyapunov functions, named in honour of Aleksandr Lyapunov, serve as scalar functions to verify an equilibrium's stability in differential equations. These functions, alternatively known as Lyapunov's second method for stability, play a crucial role in the stability theory of dynamical systems and control theory. Consider a continuously differentiable function $V : D \to \mathbb{R}$ defined in a domain $D \subset \mathbb{R}^n$ that encompasses the origin. The derivative of V along the trajectories of equation (1.42), denoted by $\dot{V}$, is expressed as follows

$$\begin{aligned}\dot{V}(x,t) &= \sum_{i=1}^{n} \frac{\partial V}{\partial x_i}\dot{x}_i = \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} f_i(x) \\ &= \begin{bmatrix} \frac{\partial V}{\partial x_1}, & \frac{\partial V}{\partial x_2}, & \cdots, & \frac{\partial V}{\partial x_n} \end{bmatrix} \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix} = \frac{\partial V}{\partial x} f(x)\end{aligned}$$

The derivative of V along the trajectories of a system is dependent on the original equation governing the system. Consequently, $\dot{V}(x,t)$ will exhibit variability across different systems. If $\phi(t;x)$ denotes the solution of equation (1.42) with an initial state of x at time $t = 0$, then the expression for $\dot{V}(x,t)$ can be articulated as follows

$$\dot{V}(x,t) = \frac{d}{dt} V(\phi(t;x)) \Big|_{t=0}$$

Consequently, when $\dot{V}$ is negative, the function V will experience a decrease along the solution trajectory of equation (1.42).

Theorem 1.1. *Consider $x = 0$ as an equilibrium point for equation (1.42), and let $D \subset \mathbb{R}^n$ be a domain containing $x = 0$. Suppose there exists a continuously* *differentiable function* $V : D \to \mathbb{R}$ *such that*

$$V(0) = 0 \text{ and } V(x,t) > 0 \text{ in } D - \{0\}$$

$$\dot{V}(x,t) \leq 0 \text{ in } D$$

In such a case, the equilibrium point $x = 0$ *is stable.*

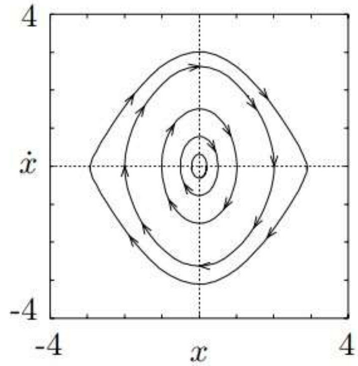
Furthermore, if the condition $\dot{V}(x,t) < 0$ *holds within the domain* $D - \{0\}$*, then* $x = 0$ *is characterized as asymptotically stable.*

Proof. The proof of this theorem is easily found in [29]. □

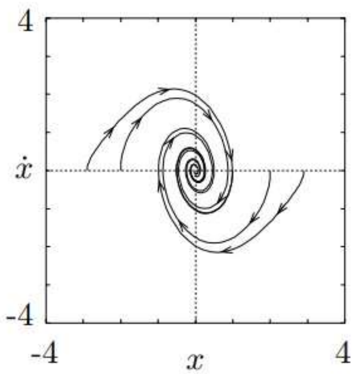
Theorem 1.2. *[29] If within every neighborhood of the equilibrium point* $x = 0$*, there exists at least one point where* $V(x,t)$ *is positive (negative), and additionally, along the trajectories of the system, the derivative* $\dot{V} > 0$ $(\dot{V} < 0)$*, then the origin of the system is deemed an unstable critical point of* (1.42)*.*

1.5.3 Finite and Fixed Time Stability

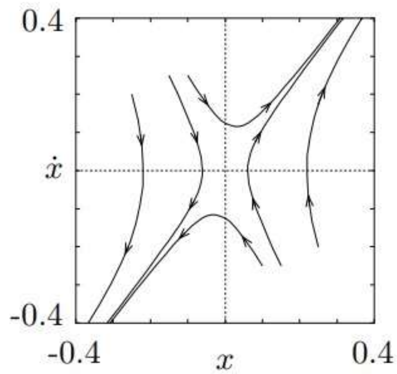
The concept of finite-time stability was initially introduced in the 1950s [30]. In control theory, finite and fixed-time stability (FITS) are concepts that describe the behaviour of dynamical systems in response to perturbations. Unlike asymptotic stability, which only guarantees that the system's state will approach an equilibrium point as time goes to infinity, FITS guarantees that the system will reach equilibrium within a finite time. Fixed-time stability (FTS) is a stronger form of FITS in which the settling time, the time it takes for the system to reach the equilibrium point, is bounded by a constant independent of the system's initial state. The FTS required for this convergence depends on the initial conditions. On the other hand, fixed-time stability is unique in that it remains independent of the initial conditions.



(a) Stable in the sense of Lyapunov



(b) Asymptotically stable



(c) Unstable (saddle)

FIGURE 1.13: Phase portrait for stable and unstable equilibrium point

The concept of fixed-time stability presented by Polyakov [31] in 2012 aimed to eliminate the dependence on initial conditions. Moreover, A dynamical system is said to be finite-time stable (FTS) if there exists a finite settling time T such that for any initial state within a certain region of attraction, the system's state will reach the equilibrium point within T seconds. The settling time may depend on the initial state, but it is always finite. A dynamical system is said to be fixed-time stable (FXTS) if a finite settling time T exists independent of the system's initial state. This means that for any initial state within a certain region of attraction, the system's state will reach the equilibrium point within the same fixed amount of time T. i.e.

1. **Finite Time Stability** The equilibrium point of a dynamical system is considered finite-time stable [31] if, for any solution $x(t)$ starting at x_0, it is asymptotically stable, and $\lim_{t \to T(x_0)} |x(t)| = 0$ holds, with $x(t) = 0$ for all $t \geq T_0$, where T is referred to as the settling time function.

2. **Fixed Time Stability** An equilibrium point is termed fixed-time stable [31] if it satisfies the conditions of finite-time stability and, additionally, the settling time is bounded, i.e., $T(x_0) < T_{\max}$ for all $x_0 \in \mathbb{R}^n$, where $T_{\max}$ is a constant.

A concrete example of finite-time convergence can be observed in the context of bang-bang time-optimal feedback control [32]. The practical applications of both finite and fixed-time stabilities extend across a wide range of fields, encompassing robotics, hypersonic missiles, power systems, engineering systems, and space technology [33, 34, 35].

1.5.4 Delay Differential Equations

A delay differential equation (DDE) is a type of differential equation in which the highest order derivative occurs at only one specific value of the independent variable, and this value is not less than the corresponding values of the independent variable for the unknown function and its lower order derivatives appearing in the equation. Certainly, the first example,

$$\dot{x}(t) = x^3(t-1) + x(t) \tag{1.43}$$

qualifies as a delay differential equation (DDE) because the highest order derivative $\dot{x}(t)$ is dependent on the value of $x(t-3)$. On the other hand, the second example,

$$\dot{x}(t-3) = x^3(t) + x(t) \tag{1.44}$$

does not meet the criteria for a DDE. This is because the highest order derivative $\dot{x}(t-3)$ involves a delay of 3 units, but the terms $x^3(t)$ and $x(t)$ do not share the same delayed argument, violating the characteristic structure of a delay differential equation.

Let us define

$$C = C\left([-\tau,0];\mathbb{R}^n\right) = \left\{\phi; \phi : [-\tau,0] \to \mathbb{R}^n \text{ is continuous } \right\}.$$

This space becomes a Banach space when equipped with the norm.

$$\|\phi\| = \sup_{s\in[-\tau,0]} |\phi(s)|, \quad \phi \in C,$$

where $|\cdot|$ denotes the usual Euclidean norm. Lets define $x_t : [-\tau,0] \to \mathbb{R}^n$ by

$$x_t(s) = x(t+s), \quad \text{for } s \in [-\tau,0]. \tag{1.45}$$

From a geometric perspective, the variable x_t represents the portion of the graph of x that is confined to the interval $[t-\tau,t]$ and then transformed back to the original interval $[-\tau,0]$.

For $E \subset \mathbb{R}^n$, the space $\mathcal{C}_E = C([-\tau,0],E)$ is defined as a Banach space. This space consists of continuous functions that map the interval $[-\tau,0]$ into the set E.

Definition 1.5.2. *Let $O \subset \mathbb{R}$, $f : O \times \mathcal{C}_E \to \mathbb{R}^n$ is a given function, and $D^+(x(t))$ denotes the right-hand time derivative. Then, the following relation*

$$\dot{x}(t) = f(t, x_t), \tag{1.46}$$

is a delay differential equation on $O \times \mathcal{C}_E$.

The right-hand time derivative of a function $x(t) : \mathbb{R} \to \mathbb{R}$ is defined as

$$D^+(x(t)) = \lim_{h \to 0^+} \sup \frac{x(t+h) - x(t)}{h}. \tag{1.47}$$

For a given $t_0 \in I$ and $\phi_0 \in \mathcal{C}_E$, the initial value problem (IVP) associated with the DDE (1.46) is given by

$$\begin{cases} \dot{x}(t) = f(t, x_t), & t > t_0, \\ x(s) = \phi_0(s - t_0), & \forall\ s \in [t_0 - \tau, t_0]. \end{cases} \tag{1.48}$$

It implies that the initial value at t_0 needs to specify the solution $x(t)$ for the entire past interval $[t_0 - \tau, t_0]$. Notably, the initial function or history ϕ_0 is a continuous function but may not necessarily be compatible with the delay differential equation (1.48). Consequently, the solution might not exhibit differentiability at the initial instant t_0. For instance, consider the simple model of a delay differential equation given by

$$\begin{cases} \dot{x}(t) = x(t-1), & t > 0, \\ x(s) = 1, & \forall\ s \in [-1, 0]. \end{cases} \tag{1.49}$$

The solution to the DDE (1.49) for the interval $t \in (0, 1]$ is given by $x(t) = t + 1$ with the initial condition $x(0) = 1$. It is evident from the solution that $\dot{x}(0^+) = 1 \neq$

$\dot{x}(0^-) = 0$, indicating that the first derivative of $x(t)$ at $t_0 = 0$ is not continuous. The equation (1.54) serves as a general framework encompassing various types of differential equations.

(i) If $\tau = 0$, it reduces to the ordinary differential equation

$$\dot{x}(t) = f(t, x). \tag{1.50}$$

(ii) For discrete delays, the DDE takes the form

$$\dot{x}(t) = f(t, x(t), x(t - \tau_1), ..., x(t - \tau_n)), \quad \text{where } \tau = \max_{1 \le i \le n} \tau_i. \tag{1.51}$$

(iii) In the case of distributed delay, the DDE is expressed as

$$\dot{x}(t) = \int_{-\tau}^{0} f(t, s, x(t + s))\, ds. \tag{1.52}$$

In the third case, the delay can extend to infinity, leading to the expression

$$\dot{x}(t) = \int_{-\infty}^{0} f(t, s, x(t + s))\, ds, \tag{1.53}$$

which is associated with the history function $x(s) = \phi_0(s - t_0)$ for all $s \in (-\infty, t_0]$.

The consideration of time delays in neural networks is crucial for understanding their dynamical behaviour. In the previous section, it was assumed that signal transmission in neurons is instantaneous, but this is not always the case in practice. The varied shapes and sizes of axons can lead to non-instantaneous signal transmission between neurons [36]. To incorporate this aspect, the additive model of Hopfield, represented by equation (1.27), can be transformed into the following DDE.

$$C_j \frac{dv_j(t)}{dt} = -\frac{v_j(t)}{R_j} + \sum_{i=1}^{n} w_{ji} F_i(v_i(t - \tau_{ji})) + I_j, \text{for } j = 1, 2, ..., n. \tag{1.54}$$

In this equation, τ_{ji} represents the time signals are transmitted from the i-th to the j-th neurons in the network. This type of delay is referred to as discrete delay. However, with their spatial structure and numerous parallel pathways among neurons, neural networks cannot be accurately modeled using discrete delays. Instead, there exists a distribution of propagation time delays. Therefore, to more faithfully represent neural networks, it is preferred to introduce continuously distributed delays [37, 38, 39]. The equation (1.26) can be reformulated in the form of a Delay Differential Equation (DDE) with distributed delays.

$$C_j \frac{dv_j(t)}{dt} = -\frac{v_j(t)}{R_j} + \sum_{i=1}^{n} w_{ji} F_i\big(\int_0^{\infty} v_i(t-u) g_{ji}(u) du\big) + I_j, \ j = 1, 2, .., n. \tag{1.55}$$

In the provided equation, u represents a signal delay from the i-th to the j-th neuron, occurring with a probability distribution function $g_{ji}(u)$ and a mean delay $\tau_{ji} = \int_0^{\infty} u g_{ji}(u) du$. A Delay Differential Equation (DDE) that incorporates both discrete and distributed delays is termed a DDE with mixed-time delays. The time derivative of the state variable v_j depends on v_i for the entire past interval $(-\infty, t]$. Several publications in recent years [39, 40, 41, 42] have presented results and findings related to delayed neural networks.

1.5.5 Matrix measure theory

Numerous algebraic methods are employed for the stability analysis of neural networks, including approaches based on Linear Matrix Inequality (LMI) concepts [43, 44], M-Matrices [45], H-Matrices [46], and Matrix Measure Theory [47, 48], among others. The concept of measuring matrices is an extension of the normed linear space, evolving from the norms of vectors and matrices.

Consider a finite-dimensional Euclidean linear space $\mathbb{R}^n$ over the real field $\mathbb{R}$. The norm $\|(.)\|_p$ in $\mathbb{R}^n$ for $1 \leq p < \infty$ is defined as

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}. \tag{1.56}$$

The linear space $\mathbb{R}^n$ associated with the norm defined in (1.56) is termed a normed linear space. Thus, for any $x \in \mathbb{R}^n$, the following expressions hold for $p = 1, 2, \infty$.

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|, \quad \|x\|_2 = \big(\sum_{i=1}^{n} |x_i|^2 \big)^2, \quad \|x\|_\infty = \max_{1 \leq i \leq n} |x_i|. \tag{1.57}$$

Similarly, let's define the norm in the linear space $\mathbb{R}^{n \times n}$ of matrices over the real field $\mathbb{R}$. For any matrix $A = [a_{ij}]_{n \times n} \in \mathbb{R}^{n \times n}$, the following expressions are obtained for $p = 1, 2, \infty$

$$\begin{gathered} \|A\|_1 = \max_j \sum_{i=1}^{n} |a_{ij}| \text{ (column sum)}, \quad \|A\|_2 = (\max_i \lambda_i(A^T A))^{\frac{1}{2}}, \\ \|A\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}| \text{ (row sum)}. \end{gathered} \tag{1.58}$$

The function $\|(.)\| : \mathbb{R}^{n \times n} \to \mathbb{R}^+$ is uniformly continuous and exhibits convex behavior. At any point $X \in \mathbb{R}^{n \times n}$, the one-sided directional derivative of the function $\|(.)\|$ along the direction of matrix A is defined as

$$\lim_{h \to 0^+} \frac{\|X + hA\| - \|X\|}{h}. \tag{1.59}$$

Definition 1.5.3. *The one-sided directional derivative of the norm function $\|(.)\|$ at point $I \in \mathbb{R}^{n \times n}$ in the direction of A is referred to as the matrix measure of A,* *denoted by* $\mu(A)$*. Mathematically, it is defined as*

$$\mu(A) = \lim_{h \to 0^+} \frac{\|I + hA\| - 1}{h}. \tag{1.60}$$

The existence of the limit (1.60) can be demonstrated by considering the function $f(h) = \frac{\|I+hA\|-1}{h}$. If $f(h)$ is decreasing with respect to h and bounded below, then the limit (1.59) must exist for all $A \in \mathbb{R}^{n\times n}$. Let $k \in (0,1)$, then we obtain

$$\begin{aligned} khf(kh) = \|I + khA\| - 1 &= \|k(I + hA) + (1-k)I\| - 1 \\ &\leq k\|I + kA\| + 1 - k - 1 \\ &\leq k(\|I + hA\| - 1). \end{aligned}$$

This implies that $f(kh) \leq f(h)$, indicating that $f(h)$ is decreasing with respect to h. It is worth noting that $f(h) \geq -\|A\|$. Therefore, the existence of the limit (1.59) is established. The matrix measure of A induced by the norm $\|(.)\|_p$ for $p = 1, 2, \infty$ is defined as

$$\mu_p(A) = \lim_{h \to 0^+} \frac{\|I + hA\|_p - 1}{h}. \tag{1.61}$$

Thus, we have

$$\begin{aligned} \mu_1(A) &= \max_j \big[a_{jj} + \sum_{\substack{i=1 \\ i\neq j}}^{n} |a_{ij}|\big], \quad \mu_2(A) = \max_i [\lambda_i(\frac{A + A^T}{2})], \\ \mu_\infty(A) &= \max_i \big[a_{ii} + \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|\big] \end{aligned}$$

The matrix measure has some useful properties, which are listed below

(i) $-\|A\| \leq -\mu(-A) \leq \mu(A) \leq \|A\|$.

(ii) $\mu(cA) = c\mu(A), \ \forall c \geq 0$.

(iii) $\mu(A + cI) = \mu(A) + c, \ \forall c \in \mathbb{R}$.

(iv) $\max[\mu(A) - \mu(-B), -\mu(-A) + \mu(B)] \leq \mu(A + B) \leq \mu(A) + \mu(B)$.

(v) $\mu : \mathbb{R}^{n\times n} \to \mathbb{R}$ is convex on $\mathbb{R}^{n\times n}$,i.e.,

$$\mu[\lambda A + (1-\lambda)B] \leq \lambda\mu(A) + (1-\lambda)\mu(B), \ \forall \lambda \in (0, 1).$$

(vi) $|\mu(A) - \mu(B)| \leq |\mu(A - B)| \leq \|A - B\|$.

(vii) $-\mu(-A) \leq \mathrm{Re}\lambda_i(A) \leq \mu(A)$ for all $i \in \{1, 2, ..., n\}$.

(viii) If A is non-singular $-\mu(-A) \leq (\|A\|)^{-1} \leq \|A\|$.

Proof. The proof for the properties of the matrix measure listed above can be referenced in [49]. □

The advantages of employing the matrix measure method for conducting stability analysis of neural networks are outlined as follows: (i) matrix measures can yield results that are zero, negative, or positive, whereas norms are always positive; (ii) constructing a Lyapunov function for the system is known to be challenging due to the absence of a general method. However, in the matrix measure approach, the construction of a Lyapunov function becomes more accessible. Due to these advantages of the matrix measure method, several researchers have directed their focus toward addressing the stability analysis of neural networks using this approach [48, 50, 51].

1.6 Synchronization

The term ”synchronization” originates from the Greek words: ”Syn”, meaning common, and ”Chronos” meaning time. In its literal sense, synchronization refers to the sharing of standard time. However, within the context of the theory of dynamical systems, synchronization entails the sharing of the current time by oscillations. Chaos represents a crucial phenomenon in nonlinear dynamical systems characterized by high sensitivity to initial conditions. The chaotic nature of neural networks has been extensively demonstrated by researchers in various articles [52, 53]. Achieving synchronization in chaotic systems is challenging due to their extreme sensitivity to initial conditions. Any initial correlation between identical systems, even with very close starting conditions, rapidly diminishes to zero over time. Therefore, practical synchronization between such systems tends to disappear quickly. Despite these challenges, recent efforts have proposed methods for achieving synchronized behaviour in chaotic systems. Pecora and Carroll conducted pioneering work in this area [54] and introduced the concept of a response system locking onto a driver system. While previous studies focused on driving a response system with a single driver system, the understanding gained from these simple systems may not fully capture the dynamics of systems with multiple independent driver systems competing to synchronize the same response system. The Pecora-Carroll driving mechanism can be viewed as the ”strong-coupling” limit within a broader framework of directionally oriented couplings in a network of chaotic elements.

Various types of synchronization have been reported in the literature [55, 56]. Here are descriptions of some of them

(1) **Exact Synchronization**: This type of synchronization occurs when systems are identical and coupled either unidirectionally or bidirectionally. Consider

two coupled identical systems described as follows

$$\dot{x}(t) = f(x(t)) \tag{1.62}$$

$$\dot{y}(t) = f(y(t)) + U(x(t), y(t)), \tag{1.63}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a continuous vector field and $U(x, y)$ is a coupling term. The systems (1.62) and (1.63) are said to be completely synchronized if $\|y(t) - x(t)\| \to 0$ as $t \to \infty$.

(2) **Approximate or Quasi Synchronization**: Consider the coupled systems as follows

$$\dot{x}(t) = f(x(t)) \tag{1.64}$$

$$\dot{y}(t) = g(y(t)) + U(x(t), y(t)), \tag{1.65}$$

where $g : \mathbb{R}^n \to \mathbb{R}^n$ is a continuous vector field. The systems (1.64) and (1.65) are said to be synchronized in a quasi way if there exists $T > 0$ such that $e(t) \in D = \{e(t) : \|e(t)\| < \epsilon\}$ for all $t > T$, where $e(t) = y(t) - x(t)$ and ϵ is a synchronization error bound.

(3) **Projective Synchronization**: In projective synchronization, the response system is synchronized with the drive system up to a scaling factor. The error system is defined as

$$e(t) = x(t) - \alpha y(t), \tag{1.66}$$

where $\alpha \neq 0$ is a scaling factor, and $\|e(t)\| \to 0$ as $t \to \infty$.

- Generally, if α is replaced by a diagonal matrix $\Omega = \{\alpha_1, \alpha_2, ..., \alpha_n\}$ with constant elements $\alpha_i \neq 0 \forall i$, it is called modified projective synchronization.
- If α is replaced by a diagonal matrix $\Omega(t) = \text{diag}\{\alpha_1(t), \alpha_2(t), ..., \alpha_n(t)\}$, where $\alpha_i(t) \neq 0$ is a bounded and continuously differentiable function for all i, it is called modified function projective synchronization (MFPS).
- If $\alpha_1(t) = \alpha_2(t) = ... = \alpha_n(t)$, then it is called function projective synchronization.
- When the sum of two signals converges to zero, anti-synchronization is observed; let $\alpha = -1$, then it is called anti-synchronization, i.e., their phases exhibit a difference of 180 degrees. In chaotic systems, it is an observable phenomenon with significant practical implications. Using anti-synchronization to lasers, one may generate not only drop-outs of the intensity as with ordinary low-frequency fluctuations but also short pulses of high intensity, which offer new ways of generating pulses of special shapes

Synchronization, also known as phase-locking, manifests in a wide range of systems, including neural networks, lasers, charge density waves, Josephson junction arrays, heart/breathing systems, and populations of flashing fireflies. This phenomenon holds potential applications in the treatment of Parkinson's disease and in signal processing and optomechanical systems. Other types of synchronization schemes are reported in the literature, such as anticipated, phase, and generalized synchronization; see [57, 58, 59] and the references cited therein.

1.7 Global dissipativity

In the year 1970, Willems [60] proposed the concept of dissipativity. Indeed, the stability problem holds central importance in the analysis of dynamic systems, where researchers focus on various types of stability concerning equilibrium points. However, it is worth noting that, from a practical standpoint, not every neural network necessarily exhibits orbits that converge to a single equilibrium point. In certain situations, there might not even be an equilibrium point. To address this, the concept of point dissipativity has been introduced. The concept of point dissipativity extends the notion of Lyapunov stability and has proven to be applicable in diverse areas, including stability theory, chaos and synchronization theory, system norm estimation, and robust control.

1.8 Inertial Neural Networks

Unlike traditional neural networks with first-order neuronal state variables, Babcock and Westervelt 1986 [61] explored electronic neural networks by introducing inductors into the neural circuit, giving them an inertial nature. The dynamics of such networks were characterized by second-order derivatives of neuronal state variables, leading to the development of INNs described by second-order differential equations. This consideration is particularly crucial due to its biological relevance observed in mammalian hair cells [62] and the axon of squid [63]. Understanding and incorporating this inertial term is essential for accurately modeling and simulating the dynamic behaviour of systems, drawing inspiration from biological phenomena. This sets them apart from the conventional first-order and fractional-order state variables in neural networks. The inclusion of inertial terms in neural networks introduces more complex behaviour, encompassing phenomena such as chaos and bifurcation in dynamical problems. Researchers have delved into the study of INNs and have obtained notable results in understanding and characterizing their dynamics [64, 65]. Later, the inertia is incorporated into a lag differential equation of the form,

$$\ddot{x}_1 = a\dot{x} - bx + cf(x - hx(t - \tau)),$$

Which exhibits chaotic behaviour over time. They demonstrated that the system becomes unstable as the delay increases, giving rise to an almost periodic and chaotic motion under the influence of periodic excitation. The successful application of INNs in various fields, including information processing and image encryption [4], has generated significant interest in the dynamical analysis of INNs. Researchers are increasingly focusing on stability, synchronization, and dissipativity to understand better and optimize these networks' performance in dynamic settings. This attention to the dynamical aspects of INNs reflects a growing recognition of their potential and the need for a comprehensive understanding of their behaviour in different applications. It is crucial to emphasize that the issues of stability and synchronization hold a significant position, not only within the realm of artificial neural networks but also in broader dynamical systems and the scientific community at large. Consequently, research on the stability and synchronization control of neural systems has gained extensive attention and has been thoroughly investigated. Addressing these challenges and the potential applications of findings across various scientific disciplines is important, for example, [66, 67, 68, 69]. Most stabilization and synchronization results for INNs, as described in the literature mentioned above, rely heavily on variable substitution methods. The original systems are typically transformed into two first-order systems in these approaches. This methodology facilitates the analysis and solution of stabilization and synchronization problems in INNs. While effective in addressing stabilization and synchronization for INNs, this approach has a potential drawback. The utilization of variable substitution methods often leads to an increase in the dimensions of the models. Consequently, this augmentation in model dimensions can introduce complexity to the theoretical analysis, making it more intricate and challenging to handle. Researchers must carefully navigate this trade-off between the analytical tractability of the substituted models and the increased complexity resulting from higher dimensions. Motivated by the discussions outlined above, most of the work for INNs in this thesis is founded on a non-reduction order approach. This strategic choice is driven by the aim to preserve the originality of the system. By opting for a non-reduction order methodology, the obtained results are more faithful to the system's inherent complexity and generally more straightforward to handle in terms of theoretical analysis. This approach allows for a more direct exploration of the system's dynamics without introducing unnecessary complexity through order reduction methods.

1.9 Complex Valued Neural networks

complex-valued signals are inherently present in electromagnetic and MRI signals [70]. Virtue et al. [70] reported that complex-valued neural networks (CVNNs) exhibit greater accuracy than RVNNs in the inverse mapping of MRI fingerprinting. This highlights the relevance of using complex numbers in real-world applications. CVNNs offer a distinct advantage in data representation compared to RVNNs when dealing with complex-valued problems. RVNNs typically handle the real and imaginary parts of complex-valued data independently. This approach effectively doubles the number of learnable parameters, leading to potential challenges in learning and capturing complex patterns within the data [71]. RVNNs undeniably demonstrate exceptional performance in the era of deep learning. However, the capabilities of a single real-valued neuron are still limited. A notable example is the XOR problem, which a single real-valued neuron struggles to solve. In contrast, a single complex-valued neuron can effectively address such problems and possess high generalization capability. Nitta [72] has reported the existence of the concept of orthogonal decision boundaries in complex-valued neurons. The neural network configuration mirrors the communication mechanism of neurons in the human brain. Despite the recent advancement of artificial neural networks, the well-known neural network models, whether real or complex domains, do not fully capture the intricate process of neuronal information communication. In 2014, a pertinent study [73] introduced a biologically plausible neuronal formulation, facilitating the development of a more intricate and adaptable deep network representation in CVNNs. In this context, neurons transmit information data through rhythmic electrical impulses, commonly known as spikes. Rhythmic spikes in neurons are defined by their average firing rate and the relative timing of their activity, where this data is encapsulated by the amplitude and phase in complex numbers. In contrast, RVNNs depict the neuron's output as a real number, offering an interpretation akin to the average firing rate of the neuron. In their research, experiments were carried out on deep Boltzmann machines to evaluate the synchronization of rhythmic neuronal spikes, with a particular emphasis on the significance of phase. Inputs with similar phases were termed synchronous, as they summed constructively, while those with different phases were considered asynchronous. The findings revealed that the phase of the resulting complex-valued inputs influenced the phase of the output neuron. This demonstrated the impact of synchronous neuronal firing on a postsynaptic neuron, emphasizing the significance of phase in neuronal communication. With the demonstrated computational capabilities of CVNNs and supporting evidence from a biological perspective, the application of complex numbers to neural networks appears reasonable. It was shown that the complex-valued counterpart had produced a simpler solution with better generalization. Generally, real-valued neural networks (RVNNs) are widely used, with all their parameters, inputs, and outputs represented as real numbers. In contrast, CVNNs receive less attention than RVNNs. CVNNs utilize complex numbers for both their inputs and network parameters. The outputs of CVNNs can be either real or complex numbers, depending on the focus of the study. Real numbers are often favored for their lower computational complexity and easier implementation when compared to complex numbers. The concept of CVNNs originated when a researcher introduced phase information in the computation of an invention known as "Parametron." Further research and comprehensive design of CVNNs have been explored and proposed by [74, 75]. In this work, the extension of network parameters and the widely used back-propagation (BP) algorithm into the complex domain is discussed [70]. Their proposal facilitates the processing of wave-typed signals, particularly electromagnetic and sonar signals. Recently, the applications of CVNNs have further extended into the field of MRI signal processing and various other applications, such as wind prediction, image classification, etc. [76, 77].

1.10 Quaternion valued Neural networks

1.10.1 Background

Real-world data often come in multidimensional forms, necessitating specialized approaches to capture relationships within the information. For instance, in image processing, pixels are manipulated based on their three primary visual features: the red, green, and blue (R, G, B) channels. In robotic and human-pose estimation, points in space characterized by 3D coordinates are employed as system inputs. The automatic speech recognition (ASR) process commonly relies on time-frames defined with groups of Mel-filter-bank energies, including first and second-order time derivatives. It has been demonstrated that local relations exist within these components of multidimensional entities [78]. Specifically, a given pixel's (R, G, B) components represent a more complex color, such as pink or brown, with a three-coordinate position in the color space. These relations must be captured by neural networks to effectively generalize and represent the multidimensional aspects of the pixel [79]. However, traditional real-valued architectures process these composite entities as independent elements within a large real-valued input vector. In [78] demonstrated that a real-valued neural network cannot preserve the 3D shape of an input object when transformed into $3D$ space, thus failing to retain the relations between the coordinates. In response to this limitation, neural networks based on multidimensional numbers have been proposed. CVNNs have become recognized as an effective solution for tasks involving two-dimensional input vectors. A brief discussion is given in the previous section. Nevertheless, while complex representations are effective for two-dimensional tasks, the nature of human spatial features often requires a $3D$ representation.

Quaternion-valued neural networks (QVNNs) have recently emerged as an active field of research [80, 81, 82]. Quaternions are hyper-complex numbers that consist of a real part and three separate imaginary components, making them well-suited for representing three and four-dimensional feature vectors. This is particularly relevant in applications such as image processing, where the (R, G, B) channels form a three-dimensional feature space, or in robotics, where three-dimensional features are common. QVNNs were initially introduced by Arena [83] in (1994), along with a dedicated backpropagation algorithm designed to effectively train QVNNs analogous to RVNNs. Quaternions find applications in various real-world scenarios owing to their suitability for tasks involving 3D transformations, such as rotations in 3D space and image processing. They are commonly used in rendering 3D scenes in computer graphics [84]. Pletinckx (1989) [85] provides a comprehensive set of methods for utilizing quaternions to transform objects in space, including linear interpolation of quaternions, splining quaternions, and comparisons with Euler angles. Specifically, the authors highlight the application of quaternions in rendering, modeling, and animating simple objects using unit quaternions. Color image processing using quaternion algebra and a dedicated Fourier transform has been explored, as discussed by Sangwine [86] in 1996. The approach involves representing color images with quaternions and employing a specialized Fourier transform. According to Sangwine, this quaternion-based representation and the specific Fourier transform enables the system to construct more accurate filters and better preserve color information. Interestingly, quaternions have proven to be a suitable representation not only for 3D spaces but also for other tasks, as demonstrated by their application in molecular modeling. For instance, in realistic simulations of molecular structures, quaternions were used to represent the orientation of various molecules [87]. This showcases the versatility of quaternions and their effectiveness in providing enhanced representations in various applications. Motivated by the positive attributes discussed, the present thesis focuses on QVNNs, exploring their dynamic behaviour and illustrating various applications of quaternions.

1.10.2 Quaternion definition

Quaternion numbers, denoted as H, belong to the family of hyper-complex numbers and serve as a non-commutative extension of complex numbers. These numbers were first discovered by the Irish mathematician William Rowan Hamilton in 1844 [88]. The division ring of quaternions forms a four-dimensional vector space with bases represented by the elements $1, \mathbf{i}, \mathbf{j}$, and $\mathbf{k}$. Quaternions were introduced to extend the manipulations of complex numbers to three-dimensional space, providing a powerful mathematical framework for representing transformations in three dimensions. Indeed, a quaternion $\mathbf{Q}$ is expressed as

$$\mathbf{Q} = r + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

In quaternion algebra, the mathematical relations governing the imaginary components are given by

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$$

These relations ensure the algebraic properties of quaternions, similar to how $\mathbf{i}^2 = -1$ defines the properties of complex numbers. Indeed, these relations are fundamental to quaternions and render the quaternion algebra ($\mathbb{H}$) a powerful tool for representing spatial rotations in $\mathbb{R}^4$. The unit components of quaternion numbers obey the Hamilton rules, i.e., the units i, j and k satisfy the following

$ij = -ji = k, jk = -kj = i, ki = -ik = j, \quad i^2 = j^2 = k^2 = ijk = -1.$

This shows that multiplication in quaternions is non-commutative. Due to non-commutative, the quaternion is said to be a skew field.

Conjugate of the quaternion number is q^* or $\bar{q} = q^R - q^I i - q^J j - q^K k$, and the modulus value $|q|$ is defined as $|q| = \sqrt{q.q^*} = \sqrt{(q^R)^2 + (q^I)^2 + (q^J)^2 + (q^K)^2}$.

If $q_1, q_2 \in \mathbb{H}$, where $q_1 = q_1^R + q_1^I i + q_1^J j + q_1^K k \in \mathbb{H}$ and $q_2 = q_2^R + q_2^I i + q_2^J j + q_2^K k \in \mathbb{H}$. The addition $q_1 + q_2$ and multiplication $q_1.q_2$ are defined as

$$\begin{aligned} q_1 + q_2 =& (q_1^R + q_2^R) + (q_1^I + q_2^I)i + (q_1^J + q_2^J)j + (q_1^K + q_2^K)k, \\ q_1.q_2 =& (q_1^R q_2^R - q_1^I q_2^I - q_1^J q_2^J - q_1^K q_2^K) + (q_1^R q_2^I + q_1^I q_2^R + q_1^J q_2^K - q_1^K q_2^J)i \\ &+ (q_1^R q_2^J + q_1^J q_2^R - q_1^I q_2^K + q_1^K q_2^I)j + (q_1^R q_2^K + q_1^K q_2^R + q_1^I q_2^J - q_1^J q_2^I)k. \end{aligned}$$

****