Chapter 2
Neurocomputing with High Dimensional Parameters

Abstract Neurocomputing has established its identity for robustness toward ill-defined and noisy problems in science and engineering. This is due to the fact that artificial neural networks have good ability of learning, generalization, and association. In recent past, different kinds of neural networks are proposed and successfully applied for various applications concerning single dimension parameters. Some of the important variants are radial basis neural network, multilayer perceptron, support vector machines, functional link networks, and higher order neural network. These variants with single dimension parameters have been employed for various machine learning problems in single and high dimensions. A single neuron can take only real value as its input, therefore a network should be configured so that conventionally use as many neurons as the dimensions (parameters) in high dimensional data for accepting each input. This type of configuration is sometimes unnatural and also may not achieve satisfactory performance for high dimensional problems. It has been revealed by extensive research work done in recent past that neural networks with high dimension parameters have several advantages and better learning capability for high dimensional problems over conventional one. Moreover, they have surprising ability to learn and generalize phase information among the different components simultaneously with magnitude, which is not possible with the conventional neural network. There are two approaches to naturally extend the dimensionality of data elements as single entity in high dimensional neural networks. In first line of attack the number field is extended from real number (single dimension) to complex number (two dimension), to quaternion (four dimension), to octanion (eight dimension). The second tactic is to extend the dimensionality of data element using high dimensional vector with scalar components, i.e., three dimension and N-dimension real-valued vectors. Applications of these numbers and vectors to neural networks have been extensively investigated in this chapter.

2.1 Neuro-Computing with Single Dimensional Parameters

In recent years, neurocomputing have emerged as a powerful technique for various tasks such as function approximation, classification, clustering, and prediction in wide spectrum of applications. Multilayer neural network and back-propagation

B.K. Tripathi, *High Dimensional Neurocomputing*,
Studies in Computational Intelligence 571, DOI 10.1007/978-81-322-2074-9_2

learning algorithm for its training are most popular in neural networks community. The primary aim of the neural network is to learn input to output mapping, and the learning algorithm achieves it by adjusting the parameters of the network, which are weights and threshold values. As these weights and thresholds are real values in the conventional neural network, it is also called real valued neural network (RVNN) or conventional ANN or refers to neurocomputing with single dimension parameters.

A conventional ANN is a model that apes the real neuron described by many researchers in the model description suggested time to time. The artificial neurons are shown connected with links going from one layer to the one immediately succeeding it, and some applications of neural networks, however, have had synapses that link the neurons of the present layer with the ones not only of the immediately succeeding layer, but also to the neurons that lie further up in the line (Lang and Witbrock 1988). The strength of the synapses (connections) is the synaptic strength, is the weight associated with the connection. In the human brain, the weight is actually the potential that controls the flow of electric impulses through the link. Each neuron has a well defined aggregation function to process the integration of impinging signals and an activation function that limits the output in predefined range. A typical activation function that closely resembles the activation of real biological neuron is sigmoid function as shown in Fig. 2.1. A steepness factor was introduced to adjust the shape of the activation function and tailor it to a form that closely resembles the actual characteristic. A number of neurobiological studies and biophysics of computation have inspired researchers to precisely state different aggregation function in literature. Chapter 4 of this book presents three new higher order neuron models based on these studies.

ANN in real domain have limitations such as slow convergence and degree of accuracy achieved is normally lower, specially for many applications, which deal with high dimensional signals. The easiest solution would be to consider a conventional real domain neural network, where high dimensional signals are replaced by inde-

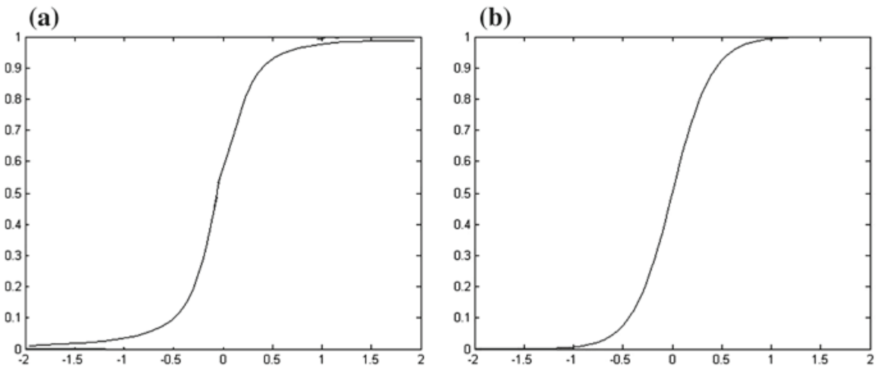

Fig. 2.1 Comparing the actual response *curve* of biological neuron with the mathematical function. **a** The actual characteristic at the output of the neuron in the brain. **b** The characteristic described by sigmoid function

pendent real-valued signals. Such a real-valued neural network is unable to perform mapping in higher dimensions because corresponding learning algorithms cannot preserve each input point's angle in magnitude as well as in sense. Besides that the huge network topology is another limiting factor, which enhances storage memory requirement, If conventional ANN is applied for high dimensional problems, they also require a large number of training iterations for the acceptable solutions and provide poor class distinctiveness in classification. There may be cases in which learning speed is a limiting factor in practical applications of neural networks to problems that require high accuracy. The most acceptable solution resulted from researches is to consider neural network designed with high dimensional parameters. High dimension neural networks have been found worth while in recent researches to overcome from these problems considerably. This book is an attempt to further improve many issues through consideration of theoretical and practical aspects of high dimensional neurocomputing.

2.2 Neurocomputing in High Dimension

In our day to day life, we come across many quantities that involve only one value (magnitude), which is a real number. However, there are also many quantities that involve magnitude and direction. Such quantities are generally called vectors, which may be represented by hypercomplex number system and/or real-valued vectors. It is a matter of universal incidence that a vector represents a cluster of particles in the factual world. The recent researches in neurocomputing are dedicated to formulate a model neuron that can deal with N signals as one cluster, called N-dimensional vector neuron. In this book, high dimensional neurons are defined through these vectors, and high dimensional neural networks such as complex neural networks, quaternary neural networks, and three-dimensional exterior neural networks are unified in terms of a vector representation. These vectors (signals), which are supposed to flow through a high dimensional neural network, are the unit of learning. Therefore, they are capable of learning high-dimensional motions as its inherent property, which is not possible in real domain neural networks.

In science and engineering, we frequently come across with both types of quantities. In mathematics, a hypercomplex number is a traditional term for an element of algebra over a Field.[1] The hypercomplex numbers are the generalization of the concept of real numbers to n dimensions which come up with an unexpected outcome. Our idea of "number like" behavior in $\boldsymbol{R}^n$ is motivated by the cases $n = 1$ (real numbers $\boldsymbol{R}$), 2 (complex numbers $\boldsymbol{C}$) that we already know. In trying to generalize the real number to higher dimensions, we find only four dimensions where the

[1] In abstract algebra, a field is a set F, together with two associative binary operations, typically referred to as addition and multiplication.

idea works: n = 1, 2, 4, 8. These number systems have many common algebraic and geometric properties.

A literature survey into the vector algebra brings out the fact that many searches had gone to enable the analysis of quantities, which involve magnitude and direction in three-dimensional space, in the same way as complex numbers had enabled analysis of two-dimensional space, but no one could arrive at a complex numbers like system. Therefore, another way to represent a vector in three dimensional space (or $\boldsymbol{R}^3$) are identified with triples of scalar components. These quantities are often arranged into a real-valued vectors, particularly, when dealing with matrices. These standard basis vectors can also be generalized into n-dimensional space (or $\boldsymbol{R}^n$).

This chapter is devoted to present a theoretical foundation of hypercomplex number system and high dimensional real-valued vector. This chapter also establishes their basic concepts for vector representation along with their algebraic and geometric properties in view of designing high dimensional neural networks. The successive chapters of the book lead to their vital applicability in various areas.

2.2.1 Hypercomplex Number System

The nineteenth century is observed as the very thrilling time for philosophy of complex numbers. Though, complex numbers had been discussed in works published in the sixteenth century, the study of complex numbers was totally dismissed as worthless at that time. After three centuries the sensibleness of complex numbers was truly understood when most of the fundamental results, which now form the core of complex analysis, were exposed by Cauchy, Riemann along with many others. Even Irish mathematician and physicist William Rowen Hamilton was fascinated by the role of complex number system in two-dimensional geometry in the nineteenth century before discovering quaternions. Indeed, set of real numbers '$\boldsymbol{R}$' is a subset of set of complex numbers '$\boldsymbol{C}$'; $\boldsymbol{C}$ is a Field extension over $\boldsymbol{R}$. It is a number system where we can add, subtract, multiply, and divide. The algebraic structure "doublets" or "couplets" ($a + ib \in R^2$) was regarded as a algebraic representation of the Complex Numbers, which can easily use complex arithmetic to do various geometric operations. The Field of complex numbers is defined by

$$\boldsymbol{C} = \{a + i\ b \ \mid \ a, b \ \in \ R; \ \ i^2 = -1\}$$

The field of complex numbers is a degree two field extension over the field of real numbers; the vector space of C forms the basis $\{\mathbf{1}, \mathbf{i}\}$ over $\boldsymbol{R}$. This means that every complex number can be written in the form $a + \mathbf{i}\ b$, where a and b are real numbers and i is an imaginary unit ($i^2 = -1$). Mathematicians want to construct a new fields for hypercomplex numbers such that $\boldsymbol{C}$ becomes a subset of hypercomplex number system, and the new operations in them are compatible with the old operations in $\boldsymbol{C}$. Therefore, mathematicians were looking for a field extension of $\boldsymbol{C}$ to higher dimensions, hence hypercomplex numbers.

As next natural step, Hamilton was eager to extend the complex numbers to a new algebraic structure with each element consisting of one real part and two distinct imaginary parts, which would be known as "Triplets" and forms the basis $\{\mathbf{1}, \mathbf{i}, \mathbf{j}\}$. There are necessary and sufficient mathematical reasons, why one should attempt for such a construction. He desired to use these triplets to operate in three-dimensional space, as complex numbers were used to define operations in the two-dimensional plane. He attempted for years to invent an algebra of triplets ($a + \mathbf{i}\, b + \mathbf{jc} \; \in R^3$) to play same role in three dimensions.

Can triplets be multiplied? Hamilton worked unsuccessfully in creating this structure for over 10 years. After a long work he observed that they can only be added and subtracted; Hamilton could not solve the problem of multiplication and division of triplet. We now know that this pursuit was in vain. Historically, it was noted that on October 16th, 1843, while walking with his wife along the Royal Canal in Dublin, the concepts of using quadruple with the rules of multiplications dawned on him. Hamilton discovered a four dimensional algebraic structure called the quaternions. The discovery of the quaternions is one of the most well documented discoveries in mathematics. In general, it is very rare that the date and location of a major mathematical discovery are known. Hamilton explicitly stated, "I then and there felt the galvanic circuit of thought and the sparks which fell from it were the fundamental equations between $\mathbf{i}, \mathbf{j}, \mathbf{k}$; exactly such as I have used them ever since in complex number system".

The set of quaternions, often denoted by $\boldsymbol{H}$ in honor of its discoverer, constitute a noncommutative field (a skew field) that extends the field $\boldsymbol{C}$ of complex numbers. The quaternions is constructed by adding two new elements $\mathbf{j}$ and $\mathbf{k}$ in complex number, thus new algebraic structure would require three imaginary parts along with one real part. For this new structure to work, Hamilton realized that these new imaginary elements would have to satisfy the following conditions:

$$i^2 = j^2 = k^2 = ijk = -1$$

One could now talk about the additive and multiplicative operations that can be defined on elements of H and turn it into a field. The field of quaternions can then be written as

$$\boldsymbol{H} = \{q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \;\mid\; q_n \in \boldsymbol{R}; \;\; \text{and} \;\; i^2 = j^2 = k^2 = ijk = -1\}$$

If Hamilton had been able to develop his Theory of Triplets, he would have effectively built a degree three field extension of $\boldsymbol{R}$ whose vector space forms the basis $\{\mathbf{1};\ \mathbf{i};\ \mathbf{j}\}$ over $\boldsymbol{R}$ such that $i^2 = j^2 = -1$. This field must be closed under multiplication. After struggling with all possibilities, researchers including himself came in conclusion that there is no third degree Field extension over $\boldsymbol{R}$ with basis $\{\mathbf{1};\ \mathbf{i};\ \mathbf{j}\}$ holding the properties as in $\boldsymbol{C}$. Thus, it is not possible to create the Theory of Triplets while satisfying the requirements of a Field. Hamilton had to abandon the Theory of Triplets.[2] The extensive investigations show why Hamilton had to consider a four dimensional Field extension by adding a new element **k** that is linearly independent of the generators **1**, **i**, and **j**; and whose vector space forms the basis {**1**; **i**; **j**; **k**} such that $i^2 = j^2 = k^2 = -1$.

A quaternion is a hypercomplex number, which is an extension to the complex numbers. The hypercomplex number domain possesses a "Numberlike" behavior in $\boldsymbol{R}^n$ and consists of symbolic expression of n terms with real coefficients, where n may be 1 (real numbers), 2 (complex numbers), 4 (quaternion), 8 (cayley numbers or octonion). These numbers can also be considered as Field extension of classical algebra in hypercomplex number domain. They share many properties with complex numbers with interesting exceptions. Quaternion algebra has all the required properties except commutative multiplication, whereas the octonion algebra has all the required properties except commutative and associative multiplication. They are now used in computer graphics, computer vision, robotics, control theory, signal processing, attitude control, physics, bioinformatics, molecular dynamics, computer simulations, and orbital mechanics.

2.2.2 Neurocomputing with Hypercomplex Numbers

The brief survey into the development of family of hypercomplex numbers point out the fact that the idea of developing these numbers may generate normed division algebras only in dimensions 1 (real numbers), 2 (complex numbers), 4 (quaternions), and 8 (octonions). Hypercomplex numbers are direct extension of the complex number into the high dimension space. They can be seen as high dimensional vectors comprising of components with one scalar and a vector in space. High-dimensional neural networks developed using these numbers has natural ability of learning motion in corresponding dimension because the unit of learning are these numbers (signals) flowing through respective neural network. The neural network in hypercomplex domain is an extension of the classical neural network in real domain, whose weights, threshold values, input, and output signals are all hypercomplex numbers. This chapter will clarify the fundamental properties of a neuron and neurocomputing with hypercomplex number system.

2.2.3 Neurocomputing with Vectors

The word vector was originated from the Latin word vectus, which stands for "to carry". The modern vector theory was evolved from early nineteenth century when an

---

[2] Interested readers may consult modern abstract algebra to understand difficulty of building a three-dimensional field extension over $\boldsymbol{R}$ and Hamiltons breakthrough concerning the necessity of three distinct imaginary parts along with one real.

interpretation of vector in different dimensions had been given through the tuples of scalar components. Vectors in a three dimensional space (or $\boldsymbol{R}^3$) can be represented as coordinate vectors in a Cartesian coordinate system and can be identified with an ordered list of three real numbers (tuples [x, y, z]). These numbers are typically called the scalar projections (or scalar components) of the vector on the axes of the coordinate system. In wide spectrum, a vector in n-dimensional space (or $\boldsymbol{R}^n$ or spatial vector) is a geometric quantity having magnitude (or length) and direction expressed numerically as n-tuples, splitting the entire quantity into its orthogonal-axis components. The vector-valued neurons were introduced as natural extension of conventional real-valued neuron, which influence the behavioral characteristics of the neuron in high dimensional space. We live in a three dimensional world, certainly all of our movements are in 3D. There are many natural aspects of learning 3-D motion in space particularly through neurocomputing. The purpose of 3D vector-valued neural network to 3D geometry is that it makes the neurocomputing study simple and elegant.

2.3 Neurocomputing with Two Dimensional Parameters

The neurocomputing has found application in almost all industry and every branch of science and technology. Our understanding of the ANN improved over the years with the much light research in the direction threw, with the lasting contributions of McCulloch and Pitts (1943), Donald Hebb (1949), Minsky (1954), Rosenblatt (1958), Minsky and Papert (1969), Werbos (1974), Fukushima and Miyaka (1980), John Hopfield (1982), Nitta (1997), Adeli (2002), Aizenberg (2007), Tripathi (2010) to name a few. Among the most recent developments in the area are the complex variable based neural networks (CVNN) that represent a second generation of architectures and also scored over the standard real variable based networks (ANN) in certain aspects. All the parameters including synaptic weights, bias, input-output, and signals flowing through network are complex numbers, aggregation, and activation functions are also in complex domain. Since it operates in the complex variables setting, the conventional Back-Propagation Algorithm (BP) that trains the ANN is not suitable to train the CVNN. The operations on functions in complex domain are not as straightforward as in real domain; therefore, variations in extension of the BP to the complex variables was reported by Leung and Haykin (2010) [1], Piazza (1992) [2], Nitta (2000) [3], Aizenberg (2007) [4], Adeli (2002) [5] called the Back-Propagation Algorithm in Complex Domain ($\boldsymbol{C}$BP). It is imperative that we study the ANNs with two-dimensional parameters with a view to investigate how the new tools of approximation perform in comparison with the existing ones.

In order to preserve the relationship between phase and magnitude in signals, one certainly requires a mathematical representation; and this representation is only possible in the domain of complex numbers. Hence, the model representation of the systems involving these signals should deal with complex values rather than real values. This indicates that the complex variable-based neural network may be useful for such applications. CVNN is the extension of RVNN, in which all the parameters and signals flowing through it are complex numbers (Two Dimension Parameters) in contrast to real numbers in the RVNN. The different neurobiological studies revealed that the action potential in human brain may have different pulse patterns and the distance between pulses may be different. This justifies the introduction of complex numbers representing phase and magnitude by a single entity into neural networks. CVNN has been applied to various fields like adaptive signal processing, speech processing, and communicating, but are not limited. These applications often use signals, which have two types of information embedded in it, the magnitude and the phase. In real domain it is not possible to represent both these quantities by a single quantity; so the magnitude and phase is represented by two numbers and then the neural network can be trained using these two quantities as separate inputs. While the neural network trained on this topology may give satisfactory results, but the relationship between phase and magnitude of a signal cannot be represented by the model because of their separation. Chapter 5 demonstrate that the CVNN can learn and generalize linear and bilinear transformation over typical geometric structures on plane. These transformations cannot be learned using RVNN. The CVNN shows excellent generalization capabilities for these transformations because representation of magnitude and phase by single quantity i.e. complex number. It also worth to mention here that CVNN has yielded far better result even in case of real-valued problem (Chap. 4), hence outperformed over equivalent real-valued neural network.

2.3.1 Properties of Complex Plane

The complex plane is the geometric analog of complex numbers, which has a long and mathematically rich history. It is the set of dual numbers over the real that possesses one to one correspondence with the points of cartesian plane. The complex plane is unlike real line, for it is two-dimensional with respect to real numbers and one-dimensional with respect to the set of complex numbers (Halmos 1974). A point on the plane can be viewed as a complex number with the x and y coordinates regarded as the real and imaginary parts of the number. The set of complex numbers is a Field equipped with both addition and multiplication operations, and hence makes a perfect platform of operation. But the order that existed on the set of real numbers is absent in the set of complex numbers, and as a result, no two complex numbers could be compared as being big or small with respect to each other, but their magnitudes (which are real numbers) could well be compared. The properties of the complex plane are different from those of the real line. The set of real numbers was one-dimensional, while as was pointed out, the set of complex numbers is one-dimensional if the field in question is the set of complex numbers itself, while it is two-dimensional if the filed is the set of real numbers. The complex numbers have a magnitude associated with them and a phase that locates the complex number uniquely on the plane. It is hence clear that the learning algorithm on complex plane (***C***BP) that trains the CVNN must not only obtain a convergence with respect to the magnitude, but also with respect to the phase. This is equivalent to stating that the real as well as imaginary parts of the complex numbers must be separately captured by the ***C***BP.

2.3.1.1 Beauty of Complex Numbers

Complex domain (***C***) itself is gaining more attention because many real applications involve signals that are inherently complex-valued. A complex number is directly related to the two-dimension data. It comprises of two real numbers and comes with phase information embedded into it. Addition and multiplication are much easier in ***C***. Any complex number has a length and angle, hence forms a plane. Their operations are very related to two-dimensional geometry where one can use complex arithmetic to do various geometric operations. Thus, it is more significant in problems where we wish to learn and analyze signal amplitude and phase precisely. Let ***C*** be the set of complex numbers and triplet $(F, \bullet, \otimes)$ be a Field equipped with operations $\bullet$, $\otimes$, satisfying the Closure, Commutative, Associative, Identity, Distribute ($\otimes$ distributes over $\bullet$), and Inverse properties for arbitrary elements belonging to F. A two-dimensional Field with basis $\{\mathbf{1}, \mathbf{i}\}$ forms a two-dimensional vector space of ***C*** over ***R***.

Definition 2.1 The triplet $(F, \bullet, \otimes)$ is said to be a Field equipped with operations $\bullet, \otimes$; $\forall c_1, c_2, c_3 \in F$, if it satisfies the following axioms with respect to $\bullet$:

1. *Closure* If $c_1, c_2 \in F \Longrightarrow c_1 \bullet c_2 \in F$.
2. *Commutative* $c_1 \bullet c_2 = c_2 \bullet c_1$.
3. *Associative* $c_1 \bullet (c_2 \bullet c_3) = (c_1 \bullet c_2) \bullet c_3$.
4. *Identity* $\exists$ an element called '0' such that $c_1 \bullet 0 = 0 \bullet c_1 = c_1$.
5. *Inverse* $\exists$ a unique c_1 for every c_2 such that $c_1 \bullet c_2 = 0$.

and following axioms with respect to $\otimes$:

1. *Closure* If $c_1, c_2 \in F \Longrightarrow c_1 \otimes c_2 \in F$.
2. *Commutative* $c_1 \otimes c_2 = c_2 \otimes c_1$.
3. *Associative* $c_1 \otimes (c_2 \otimes c_3) = (c_1 \otimes c_2) \otimes c_3$.
4. *Identity* $\exists$ an element called '1' such that $c_1 \otimes 1 = 1 \otimes c_1 = c_1$.
5. *Inverse* $\exists$ a unique c_1 for every $c_2 (\neq 0)$ such that $c_1 \otimes c_2 = 1$.

and also *Distributive Property*, $\otimes$ distributes over $\bullet$ such that $c_1 \otimes (c_2 \bullet c_3) = c_1 \otimes c_2 \bullet c_1 \otimes c_3$.

Definition 2.2 The Field of complex numbers is a degree two field extension over the field of real numbers. The set of complex numbers is a Field equipped with operations $+, \times$ such that for every two elements $a + jb$, $c + jd$:

- $(a + jb) + (c + jd) = (a + c) + j(b + d)$.
- $(a + jb) \times (c + jd) = (ac - bd) + j(bc + ad)$.

where j is the imaginary unit defined by the equation $j^2 = -1$. It can be verified that the definitions for $+$, $\times$ satisfy all the postulates of the Field.

2.3.1.2 Cauchy-Riemann Equations and Liouville's Theorem

In real domain, the property of differentiability is not a very strong property for functions of real variables. It is surprisingly true that study of complex function for differentiability (analyticity) is a different topic from real analysis. The power and importance of complex numbers cannot be exploited until a full theory of analytic (holomorphic) function is developed. Interested readers may consult the theory of complex numbers for details, this section only presents the brief discussion on differentiability necessary for the development of learning algorithm in neurocomputing.

Definition 2.3 A complex valued function $f\colon \boldsymbol{C} \longrightarrow \boldsymbol{C}$ is said to be ***analytic*** (complex differentiable) at $z \in \boldsymbol{C}$ if the following limit exists at every point z in the complex plane. If the function is analytic over the whole finite complex plane, it is said to be an entire function.

$$\mathrm{Limit}_{\delta z \to 0} \frac{f(z + \delta z) - f(z)}{\delta z} \tag{2.1}$$

The definition demands that the function be differentiable at every point in some neighborhood of the point z. The function f is said to be differentiable at z when its derivative at z exists. The limit will be called the first derivative of f at z and denoted by $f'(z)$.

Further, there are a pair of equations that the first-order partial derivatives of the component functions of a function (f) with complex variable, must satisfy at a point when the derivative of f exists there.

Let $f(z) = U(x, y) + jV(x, y)$ be a complex valued function. Cauchy-Riemann equations (***CR***) are pair of equations that first-order partial derivatives of the component functions, U and V, of function f must satisfy at a point when the derivative of f exist there.

Definition 2.4 The first order partial derivatives of component function of f must exist. A complex valued function $f(z) = U(x, y) + jV(x, y)$ is said to satisfy ***CR*** equations if the following equalities hold:

$$\frac{\delta U}{\delta x} = \frac{\delta V}{\delta y} \quad \text{and} \quad \frac{\delta U}{\delta y} = -\frac{\delta V}{\delta x} \tag{2.2}$$

It can be shown that every analytic function satisfies CR equations. The converse is true if an additional condition of continuity of the partial derivatives of the CR is assumed (Ahlfors 1979).

Above equations, not only give the derivative of f in terms of partial derivatives of component function, but also they (***CR*** equations) are necessary conditions for the existence of the derivative of a function at a point z. They can be used to locate points at which function does not have a derivative. These equations are not sufficient to ensure the existence of the derivative of a function at that point.

The complex plane unlike the real line is a two-dimensional space. The second dimension adds flexibility and at the same time restricts the choice of activation functions for neural network applications by imposing certain constraints. More precisely, the important constraint imposed by the complex plane is epitomized in the Liouville Theorem.

Theorem 2.1 *The theory of functions in complex domain imposes its own constraint in the form of Liouville Theorem, which states that if a function in complex domain is both entire and bounded, it must be a constant function. As a ramification of the theorem, the constraints emerge:*

- *No analytic function except a constant is bounded in the complex plane.*
- *An analytic complex function cannot be bounded on all points of the complex plane unless it is constant.*

Analyticity, boundedness are the parameters of concern in the search for complex activation functions. The term regular and holomorphic are also interchangeably used in the litrature to denote analyticity. In view of theorem, if a nontrivial complex-valued function is analytic it must go unbounded at least one point on the complex plane, and if the function is bounded it must be nonanalytic in some region for it to qualify as activation function. Hence, a search for activation function should make sure these conditions are satisfied. The second dimension of the complex plane necessitates a study of three-dimensional surfaces (Chap. 3), as the real and imaginary parts of the complex activation functions are both functions of real and imaginary parts of the variable.

Definition 2.5 If a function fails to be analytic at a point z_0, but is analytic at some point in every neighborhood of z_0, then z_0 is called a singular point or singularity of function.

2.3.2 Complex Variable Based Neural Networks

Complex numbers form a superset of real numbers, an algebraic structure that defines real world phenomenon like signal magnitude and phase. These are useful in analyzing various mathematical and geometrical relationships in two dimension space. For nearly a decade, the extension of real-valued neurons for operation on complex signals [2, 5–7] has received much attention in the field of neural networks. The main motivation in designing complex variable based neural networks is to utilize the promising capabilities of complex numbers. Complex numbers are a subfield of quaternions. The decision boundary of the complex-valued neuron consists of two hypersurfaces, which intersect orthogonally each other and divides a decision region into 2^2 (=4) equal sections. Now, neural network dealing with complex numbers are not the new entrants to the field of neural networks; they have established the basic theories, yet they require more exploration in new coming applications. A brief survey into the CVNN brings out the fact that it provides faster convergence with better results, reduction in learning parameters (network topology), and the ability to learn two-dimensional motion of signal [3, 8]. The weight update rule for the CVNN is exactly same as the one used to training networks using the most popular error correction learning.

$$w(t+1) = w(t) + \eta \frac{\delta E)}{\delta w(t)} \tag{2.3}$$

where w is the weight that get updated as the algorithm runs iteratively, E is the Error Function that gets minimized in the process of weight update and η is the learning rate. The difference of course lies in the fact that the weights are complex numbers, while the error function and learning constant are positive real numbers. All the signals that the neurons fire in response to aggregation and activation functions are all complex in nature. It must be emphasized here that continuity and differentiability of a function in complex domain play a central role in development of complex variable based neural networks. It is hence obvious that a thorough discussion of complex variables and complex mappings is essential to comprehend the mechanism by which a CVNN operates. The definitions discussed here are required to prepare the ground for a systematic study to develop a theory for analysis.

2.4 Neurocomputing with Three Dimensional Parameters

Artificial neural networks have been studied for many years in the hope of achieving human like flexibility in processing typical information. Some of the recent researches in neurocomputing concerns the development of neurons dealing with three-dimensional parameters [9] and their applications to the problems, which deals with three-dimensional information. There has been rapid development in the field of 3D imaging, computer vision, and robotics in last few years. These are multidisciplinary fields, which encompasses various research areas and deal with information processing through modern neurocomputing paradigm. They are at their infancy [10–12] and requires to explore methods based on neural networks. The 3D motion interpretation and 3D feature recognition are essential part of high level analysis, and found wide practical uses in the system development of these fields. Although, there are many methodologies [10, 11, 13, 14] to solve them, they instead use extensive mathematics and are time consuming. They are also weak to noise. Therefore, it is desirable for realistic system to consider iterative methods, which can adapt system for three-dimensional applications. This book is aimed at presenting relevant theoretical and experimental framework based on multilayer neural networks of 3D vector-valued neurons. In Chap. 6, we present a straightforward technique that uses 3D geometric point set (point cloud) representation of objects. The method described here is fully automatic, does not require much preprocessing steps, and converges rapidly to a global minima.

2.4.1 Properties of Vector Space

A vector space is a set whose elements are called vectors. The vector space is intuitively spatial since all available directions of motion can be plotted directly onto a spatial map. The idea of a vector is far more general than the picture of a line with an arrowhead attached to its end. In general, a vector is thought a directed arrow pointing from the origin to the end point given by the list of numbers. A vector is a list of numbers, and the dimensionality of a vector is length of the list, where each number represents the vector's component in the dimensions. A 3-dimensional vector would be a list of three numbers, and they live in a 3-D volume. Vector space offers a convenient way to describe different geometric properties. There are two operations (addition and scalar multiplication) defined on them, which must obey certain simple rules, the axioms for a vector space.

Definition 2.6 Let set $\boldsymbol{V}$ be the vector space and the tuple $(F, +)$ is a Field equipped with operations $+$. The $\boldsymbol{V}$ equipped with the operation '+' is said to be a vector space over the field F if the following axioms are satisfied $(\forall v_1, v_2, v_3 \in V)$

- *Closure* vector space is closed under addition and multiplication by scalars $v_1 + v_2 \in \boldsymbol{V}$ and $\alpha\ v_1 \in \boldsymbol{V}$ where α is a scalar.
- *Commutative* The commutative law of addition holds $v_1 + v_2 = v_2 + v_1$.
- *Associative* $v_1 + (v_2 + v_3) = (v_1 + v_2) + v_3$.
- *Identity* There is a zero vector, so that for each vector, $\exists$ an element called '0' such that $v_1 + 0 = v_1$.
- *Inverse* There is a unique additive inverse for each vector, for each vector v_1, $\exists$ another vector v_2 such that $v_1 + v_2 = 0$; then $v_2 = -v_1$.

2.4.2 3D Vector Based Neural Networks

In machine learning the interpretation of 3D motion, 3D transformations, and 3D object matching are few expected applications. Though, there have been many methodologies to solve them, however, these methods are time consuming and weak to noise. This book presents an efficient solution using multilayered network of 3D vector-valued neurons. In 3D vector valued neural network the parameters like threshold values, input-output signals are all 3D real valued vectors, and weights associated with connections are 3D orthogonal matrices. All the operations in such neural models are scaler matrix operations. The corresponding 3D vector valued back-propagation algorithm (3DV-BP) is a natural extension of complex valued back-propagation learning algorithm [2, 8], and has the ability to learn *3D motion* as complex-BP learn 2D motion as its inherent property. In this book, the author investigates the characteristics of 3D vector-valued neural networks by various computational experiments. The experiments suggest that 3DV-BP networks can approximate 3D mapping just by training them only over a part of the domain of the mapping. Chapter 6 explains the learning rule for 3D vector-valued neural networks. The generalization ability of 3D neural network in 3D motion interpretation and in 3D face recognition applications is confirmed through diverse test patterns in Chap. 6.

2.5 Neurocomputing with Four-Dimensional Parameters

The four-dimensional hypercomplex numbers, the Quaternions, have been extensively employed in several fields, such as modern mathematics, physics, control of satellites, computer graphics, etc. One of the benefits in graphics provided by quaternions is that affine transformations (especially spatial rotations) of geometric constructs in three-dimensional spaces, can be represented compactly and efficiently. How we should treat data with four-dimension in artificial neural networks? Although this problem can of course be solved by applying several real-valued or complex-valued neurons. But, a better choice may be to introduce a four-dimensional hypercomplex number system based neural network, that could be confronted to the Quaternion in the same way as the complex numbers are confronted to the CVNN. This hypercomplex number system was introduced by Hamilton [15], which treat four-dimensional data elements as a single entity. There has been a growing number of interests concerning the use of neural networks in the quaternionic domain [16]. All variables in the multilayered quaternionic-valued neural network (QVNN), such as input, output, action potential, and connection weights are encoded by quaternions. A quaternionic equivalent of error back-propagation algorithm has also been investigated and theoretically explored by many researchers. The derivation of this learning scheme adopted a famous Wirtinger calculus [17] because this calculus enables a more straightforward derivation of learning rules.

2.5.1 Properties of Quaternionic Space

The quaternionic space is a four-dimensional vector space over the real numbers. Since the quaternionic algebra is at infancy there are many representations of it, which leads to variations in operations and properties. Hence, demands for wide consensus among researchers in quaternionic space. Most of the researchers have followed Wirtinger calculus in their basic constructions. Representing quaternion as a vector is more compact as well as intuitively straightforward. The quaternions may also be used for three-dimensional operations assuming 3D space as being pure imaginary quaternion.

2.5.1.1 Beauty of Quaternionic Numbers

A quaternion, the generalization of complex number, is a hypercomplex number where complex analysis would be self evident within the structure of quaternion analysis. Unlike the complex number, the quaternion has four components: one is real and the other three are all imaginary.

Definition 2.7 A class of hypercomplex numbers, the quaternions, are defined as a vector **q** in a four-dimensional vector space over the real numbers (***R***) with an ordered basis. Each number is a quadruple consisting a real number and three imaginary numbers i, j, and k. A quaternion $q \in \boldsymbol{H}$ is expressed by fundamental formula

$$\mathbf{q} = q_r + q_i\,\mathbf{i} + q_j\,\mathbf{j} + q_k\,\mathbf{k} \tag{2.4}$$

where $q_r,\ q_i,\ q_j$ and q_k are real numbers. The set of quaternions $\boldsymbol{H}$, which is equal to R^4, constitutes a four dimensional vector space over the real numbers with basis $\{\mathbf{1}; \mathbf{i}; \mathbf{j}; \mathbf{k}\}$.

Definition 2.8 The quaternion $\mathbf{q} \in \boldsymbol{H}$ can also be interpreted as having a real part q_r and vector part $\bar{q}$, where the elements $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ are given an added geometric interpretation as unit vectors along the $X,\ Y,\ Z$ axis respectively. Equation 2.4 can also be written using 4-tuple or 2-tuple (one scalar and one vector in three space) notation as

$$\mathbf{q} = (q_r,\ q_i,\ q_j,\ q_k) = (q_r, \bar{q}) \tag{2.5}$$

where $\bar{q} = q_i,\ q_j,\ q_k$. Accordingly, the subspace $\mathbf{q} = q_r + 0\,\mathbf{i} + 0\,\mathbf{j} + 0\,\mathbf{k}$ of quaternions may be regarded as being equivalent to the real numbers. The subspace $\mathbf{q} = 0 + q_i\,\mathbf{i} + q_j\,\mathbf{j} + q_k\,\mathbf{k}$ may be regarded as being equivalent to the ordinary 3D vector in R^3.

Definition 2.9 In Wirtinger calculus, a quaternion and its conjugate are treated as independent of each other, which makes the derivation of the learning scheme in neural network clear and compact. The quaternion conjugate is defined as

$$\mathbf{q}^{\aleph} = (q_r,\ -\bar{q}) = q_r - q_i\,\mathbf{i} - q_j\,\mathbf{j} - q_k\,\mathbf{k} \tag{2.6}$$

Definition 2.10 According to the Hamilton rule the quaternion basis satisfy the following identities, which immediately follows that multiplication of quaternions is not commutative.

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \tag{2.7}$$

$$\mathbf{ij} = -\mathbf{ji} = \mathbf{k};\ \ \mathbf{jk} = -\mathbf{kj} = \mathbf{i};\ \ \mathbf{ki} = -\mathbf{ik} = \mathbf{j}; \tag{2.8}$$

Definition 2.11 Let $\mathbf{p} = (p_r, \bar{p})$ and $\mathbf{q} = (q_r, \bar{q})$. Customarily, the extension of an algebra attempts to preserve the basic operations defined in the original algebra.

Let $\bar{p} \cdot \bar{q}$ and $\bar{p} \times \bar{q}$ denote the dot and cross products between the three dimensional vectors $\bar{p}\ \bar{q}$. Then basic operations between quaternions can be defined as follows:

- The addition and subtraction of quaternions are defined in a similar manner as for complex-valued numbers:

$$\mathbf{p} \pm \mathbf{q} = (p_r \pm q_r,\ \bar{p} \pm \bar{q}) = (p_r \pm q_r,\ p_i \pm q_i,\ p_j \pm q_j,\ p_k \pm q_k) \quad (2.9)$$

- The product of p and q is determined using Eq. 2.8 as

$$\mathbf{p}\ \ \mathbf{q} = (p_r\, q_r - \bar{p} \cdot \bar{q},\ \ p_r\, \bar{q} + q_r\, \bar{p} + \bar{p} \times \bar{q}) \quad (2.10)$$

- The conjugate of the product is defined as

$$(\mathbf{pq})^{\aleph} = \mathbf{q}^{\aleph}\mathbf{p}^{\aleph} \quad (2.11)$$

- The quaternion norm of $\mathbf{q}$, denoted by $|\mathbf{q}|$, is defined as

$$|\mathbf{q}| = \sqrt{\mathbf{q}\mathbf{q}^{\aleph}} = \sqrt{q_r^2 + q_i^2 + q_j^2 + q_k^2} \quad (2.12)$$

2.5.1.2 Cauchy-Riemann-Fueter Equation

The Swiss mathematician Fueter developed the appropriate generalization of the Cauchy-Riemann equations to the quaternionic functions. The analytic condition for the quaternionic functions is defined by Cauchy-Riemann-Fueter (CRF) equation, which corresponds as an extension of the Cauchy-Riemann (CR) equations defined for the functions in complex domain. In order to construct learning rules for quaternionic neural networks, CRF equation describes the required analyticity (or differentiability) of the function in the quaternionic domain.

Definition 2.12 Let $f\colon \boldsymbol{H} \to \boldsymbol{H}$ be a quaternionic valued function defined over a quaternionic variable. The condition for differentiability of any quaternionic function f is defined as follows:

$$\frac{\partial f(\mathbf{q})}{\partial q_r} = -\mathbf{i}\frac{\partial f(\mathbf{q})}{\partial q_i} = -\mathbf{j}\frac{\partial f(\mathbf{q})}{\partial q_j} = -\mathbf{k}\frac{\partial f(\mathbf{q})}{\partial q_k} \quad (2.13)$$

An analytic function can serve as the activation function in the neural network. The analytic condition for the quaternionic function, called the Cauchy-Riemann-Fueter (CRF) equation, yields:

$$\frac{\partial f(\mathbf{q})}{\partial q_r} + \mathbf{i}\frac{\partial f(\mathbf{q})}{\partial q_i} + \mathbf{j}\frac{\partial f(\mathbf{q})}{\partial q_j} + \mathbf{k}\frac{\partial f(\mathbf{q})}{\partial q_k} = 0 \quad (2.14)$$

CRF equation is necessary condition to assure analyticity in the quaternionic domain.

2.5.2 Quaternionic Activation Function

A major issue in designing neural networks in the quaternionic domain is about the introduction of suitable functions for the activation in updating the neurons' states. It is worthwhile to consider the capability of the quaternionic neural network with standard activation functions, as in CVNN, i.e., whether the considered network can approximate with given functions. A popular approach of activation function in complex domain, the so-called "split" type function,[3] has also been applied by researchers in many applications where a real-valued function is applied to update each component of a quaternionic value. Real-valued sigmoidal function and hyperbolic function, which are differentiable, are often used for this purpose. However, due to lack of analyticity in split-type quaternionic function many, others researchers suggested it an inappropriate approach.

The CRF equation has pointed out another issue on the standard analyticity of function in the quaternionic domain. Only linear functions and constants satisfy the CRF equation. For example, an analytic quaternionic tanh function can be used as an activation function, but this function may contain several kinds of singularities as in the case of corresponding function in complex domain, and hence unbounded. Thus, this quaternionic neural network may face the problem of the existence of singularities. Yet this problem can also be handled similarly as in the case of CVNN, through the removal or avoidance of such singularities; which can not considered a better choice.

Recently, another class of analyticity, called "local analyticity", has been developed for the quaternionic functions [18], and is distinguished from the standard or global analyticity. This analytic condition is derived at a quaternionic point with its local coordinate, rather than in a quaternionic space with a global coordinate. Interested reader my consult [18, 19] for details. The derivation of local analytic condition shows that a quaternion in the local coordinate system is isomorphic to the complex number system, and thus it can be treated as a complex value. A neural network with an activation function with local analyticity has been first proposed and analyzed by Mandic (2011), in [20]. Such networks demonstrated their outperformance over the network with a split-type activation function in several applications.

[3] Split complex function refers to functions $f\colon \boldsymbol{C} \longrightarrow \boldsymbol{C}$ for which the real and imaginary part of the complex argument are processed separately by a real function of real argument.

2.5.3 Quaternionic Variable Based Neural Networks

Quaternary neural networks were proposed by Arena and Nitta independently in the mid-1990s [21] The architecture of the multilayered QVNN is same as conventional neural networks except that the neurons, which process quaternionic signals and parameters in the network, are encoded by quaternionic values [16]. A quaternionic activation function introduces nonlinearity between the action potential and output in the neuron. One can treat quaternionic functions in the same manner as complex-valued functions, but under the condition of local analyticity. The quaternionic functions with local analytic conditions are isomorphic to the complex functions, thus several activation functions (such as complex-valued sigmoid functions) can be used extendedly in the quaternionic domain. Many researchers have independently considered the capability of the proposed quaternionic network with different activation functions. The connection weights in QVNN may be updated by error back-propagation learning algorithms, until the desired output signals could not be obtained with respect to the input signals. In 1927, Wirtinger calculus [17] was basically invented for analysis of complex numbers, but now it appears to be prominent in quaternionic algebra, and hence yielded a solid representation for QVNN constructions. Analytic conditions for quaternionic functions may be derived by defining a complex plane at a quaternionic point, which is a kind of reduction from quaternionic domain to complex domain.

A quaternion possesses four degrees of freedom, therefore the decision boundary of the quaternary neuron consists of four hypersurfaces, which intersect orthogonally with each other and divides a decision region into 2^4 (=16) equal sections. Nitta [16] has successfully solved the 4-bit parity problem by a single quaternary neuron with the orthogonal decision boundary, which cannot be solved with a single real-valued neuron, resulting in the highest generalization ability, and hence reveals a potent computational power of the quaternary neuron. The neural networks with quaternionic neurons has been recently explored in an effort to naturally process three or four dimensional vector data, such as color/multi-spectral image processing, predictions for three-dimensional protein structures and controls of motion in high-dimensional space. They can be highly effective in the fields such as robotics and computer vision in which quaternions have been found useful. The application of QVNN to engineering problems, such as color night vision, predictions for the output of chaos circuits, and winds in three-dimensional space is also challenging, but will be the candidates for application.

2.6 Neurocomputing with N-Dimensional Parameters

In many important applications of science and engineering, signals and system parameters are most conveniently represented as a vector in N-dimensional space using an ordered N-tuple $[x_1, x_2, \ldots, x_N]$. High dimensional neural networks designed taking account of task domains has shown their superior computational power in wide spectrum of tasks. Thus, neural network models dealing with N signals as one cluster are desired under the powerful framework of N-dimensional vector neuron. An N-dimensional vector neuron is a natural extension of the 3-dimensional vector neuron whose vital applicability has been presented in Chap. 6. The famous neuroscience researcher T Nitta (2007) proposed an efficient solution for the N -bit parity problem with a single N-dimensional vector-valued neuron [22] considering the orthogonal decision boundary. It reveals the potent computational power of N-dimensional vector neurons because this problem cannot be solved with a single usual real-valued neuron. It is reasonable to emphasize it as a new directionality for enhancing the capability of neural networks, and therefore worth researching the neural networks with N-dimensional vector neuron.

2.6.1 Properties of Vectors in R^N

The vectors in space can be directly extended to vectors in N-space. A vector in N-space is represented by an ordered N-tuples $[x_1, x_2, \ldots, x_N]$ of real numbers and same for a point in N-space, R^N. All the listed axioms in Definition 2.6 including two operations (vector addition and scalar multiplication) holds for any three vectors $v_1, v_2, v_3 \in \boldsymbol{V}$ in N-space (R^N), therefore $\boldsymbol{V}$ is called a vector space over the real numeric's $\boldsymbol{R}$.

2.6.2 N-Dimensional Vector Based Neural Networks

The structure of N-dimensional vector neuron, which can deal with N signals in one cluster, can be given by extending the structure of 3-dimensional vector-valued neuron. In an N-dimensional vector-valued neuron all the input-output signals, thresholds are N-D real-valued vectors and the weights are N-dimensional orthogonal matrices. Additional restrictions imposed on the N-dimensional orthogonal matrix (e.g., it can be regular, symmetric, or orthogonal etc.) will also influence the behavioral characteristics of the neuron. The net potential of a N-dimensional neuron can be given as:

$$Y = \sum_{l=1}^{L} W_l \quad X_l + \theta \tag{2.15}$$

where input signal $X_l = [x_1, x_2, \ldots, x_N]^T$ is lth input signal, W_l is the N-dimensional orthogonal weight matrix for the lth input signal and $\theta = [\theta_1, \theta_2, \ldots, \theta_N]^T$ is the threshold value. It is also important to mention here that the N-dimensional vector neuron presented here may be considered with the traditional activation functions. The output of the neuron will also be a N-dimensional real-valued vector. Similar to complex-valued neuron, the decision boundary of an N-dimensional vector neuron consists of N hyperplanes, which intersect orthogonally with each other and divides a decision region into N equal sections. Minsky and Papert (1969) considered the parity problem the most difficult because output required is 1 if the input pattern contains an odd number of 1 s and 0 otherwise. A solution for N-bit parity problem obtained using a single N-dimensional vector neuron demonstrates its highest generalization ability. It is significant to emphasize here that the rational improvement in the number of learning parameters and the number of layers could be achieved with the N-dimensional vector-valued neuron in solving problems possessing high dimensional parameters.

2.7 Concluding Remarks

The theories in neurocomputing have been developed to build mathematical models that mimic the computing power of the human brain. Their powerful processing capability has been demonstrated in various applications of real domain. Traditional neural networks parameters are real numbers and usually used to deal with single dimension. Still, there are many applications, which deal with high dimensional signals. The easiest solution would be to consider a conventional real domain neural network, where high dimensional signals are replaced by independent real-valued signals. Such a real-valued neural network may be highly complex and unrealistic, and besides such network is unable to perform mapping on a high dimension because corresponding learning algorithms cannot preserve each point's angle in magnitude and sense. An alternative is to introduce a neural network with high-dimensional parameters, which comprises of different components as real numbers, and comes with phase information imbedded into it. This approach yields more efficient solution both in terms of computational complexity and performance. Besides, they overcome the users from huge network topology and large storage requirements, whereas enhances the learning speed.

Another competitive advantages of neuro-computing with high- dimensional parameters is the ease with which they may be applied to poorly understood problems in higher dimensions. Neuron is its basic working unit, which does not have predefined meaning, and it evolves during learning in a manner which can characterize the target function. A high dimensional neural network has natural tendency of acquiring high-dimensional information in training, which include magnitudes and phase in a single entity. They are specially useful in areas, where there is a need of capturing phase information in signals, and must be retained all through the problem. This book is an attempt to investigate the functional capabilities of neurons with high-dimensional parameters. The strength and effectiveness of the high-dimensional neural networks have been extensively justified in successive chapters through simulations on different types of problems viz. classification, function approximation, and conformal mapping.

The processing of high-dimensional data is an important task for artificial neural networks. Multilayered neural networks of different class of high-dimensional parameters are presented and analyzed in this book. All neuronal parameters such as input, output, action potential, and connection weights are encoded by respective high-dimensional number system. The computational capability of a single complex-valued, vector-valued, or quaternionic-valued neuron has been independently presented. In order to construct learning algorithm for respective networks analytic, local analytic, or non-analytic conditions may be imposed on the activation function in updating neuron's states. Instead of using conventional description, i.e., cartesian representation, the Cliff-Ford algebra, Vector Calculus, and Wirtinger calculus may be adopted to standardize the learning rules of high dimensional neural networks.

References

1. Leung, H., Haykin, S.: The complex backpropagation algorithm. IEEE Trans. Sig. Proc. **39**(9), 2101–2104 (1991)
2. Piazza, F., Benvenuto, N.: On the complex backpropagation algorithm. IEEE Trans. Sig. Proc. **40**(4), 967–969 (1992)
3. Nitta, T.: An analysis of the fundamental structure of complex-valued neurons. Neural Process. Lett. **12**, 239–246 (2000)
4. Aizenberg, I., Moraga, C.: Multilayer feedforward neural network based on multi-valued neurons (MLMVN) and a back-propagation learning algorithm. Soft Comput. **11**(2), 169–183 (2007)
5. Kim, T., Adali, T.: Approximation by fully complex multilayer perceptrons. Neural Comput. **15**, 1641–1666 (2003)
6. Hirose, A.: Complex-Valued Neural Networks. Springer, New York (2006)
7. Shin, Y., Keun-Sik, J., Byung-Moon, Y.: A complex pi-sigma network and its application to equalization of nonlinear satellite channels. In: IEEE International Conference on Neural Networks (1997)
8. Nitta, T.: An extension of the back-propagation algorithm to complex numbers. Neural Netw. **10**(8), 1391–1415 (1997)
9. Tripathi, B.K., Kalra, P.K.: On the learning machine for three dimensional mapping. Neural Comput. Appl. **20**(01), 105–111. Springer (2011)
10. Moreno, A.B., Sanchez, A., Velez, J.F., Daz, F.J.: Face recognition using 3D surface-extracted descriptors. In: Proceedings of IMVIPC (2003)
11. Xu, C., Wang, Y., Tan, T., Quan, L.: Automatic 3D face recognition combining global geometric features with local shape variation information. In: Proceedings of AFGR, pp. 308–313 (2004)
12. Chen, L., Zhang, L., Zhang, H., Abdel-Mottaleb, M.: 3D shape constraint for facial feature localization using probabilistic-like output. In: Proceedings of 6th IEEE International Conference on Automatic Face and Gesture Recognition (2004)
13. Achermann, B., Bunke, H.: Classifying range images of human faces with Hausdorff distance. In: Proceedings of ICPR, pp. 809–813 (2000)
14. Blanz, V., Vetter, T.: Face recognition based on fitting a 3D morphable model. IEEE Trans. PAMI **25**(9), 1063–1074 (2003)
15. Hamilton, W.R.: Lectures on Quaternions. Hodges and Smith, Dublin (1853)
16. Nitta, T.: A solution to the 4-bit parity problem with a single quaternary neuron. Neural Inf. Process. Lett. Rev. **5**, 33–39 (2004)
17. Wirtinger, W.: "Zur formalen theorie der funktionen von mehr komplexen ver", anderlichen. Math. Ann. **97**, 357–375 (1927)

18. Leo, S.D., Rotelli, P.P.: Quaternonic analyticity. Appl. Math. Lett. **16**, 1077–1081 (2003)
19. Schwartz, C.: Calculus with a quaternionic variable. J. Math. Phys. **50**, 013523:1013523:11 (2009)
20. Mandic, D., Goh, V.S.L.: Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models. Wiley, Hoboken (2009)
21. Nitta, T.: A quaternary version of the backpropagation algorithm. In: Proceedings of IEEE International Conference on Neural Networks, vol. 5, pp. 2753–2756 (1995)
22. Nitta, T.: N-dimensional vector neuron. IJCAI Workshop, Hyderabad, India (2007)

http://www.springer.com/978-81-322-2073-2

High Dimensional Neurocomputing
Growth, Appraisal and Applications
Tripathi, B.K.
2015, XIX, 165 p. 49 illus., Hardcover
ISBN: 978-81-322-2073-2