Received: May 14, 2017

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Machine Learning with Resilient Propagation in Quaternionic Domain

Sushil Kumar[1]* **Bipin Kumar Tripathi**[1]

[1]*Department of Computer Science and Engineering*
Harcourt Butler Technical University, Kanpur, India
* Corresponding author's Email: sushil0402k5@gmail.com

Abstract: In the last few years there have been a growing number of studies concerning the introduction of quaternions into neural networks, which demand a faster learning technique with superior performance. In this paper, we propose a fast, but novel quaternionic resilient propagation (ℍ-RPROP) algorithm for high dimensional problems. It achieves significantly faster learning over quaternionic domain back propagation (ℍ-BP) algorithm. The slow convergence and stability of weight update around the local minima are the main drawbacks of ℍ-BP. The gradient descent based ℍ-BP algorithm takes the value of partial derivative (error gradient) and scales the weight updates through a learning rate while ℍ-RPROP does not takes the value of partial derivatives, but it considers only the sign of partial derivatives that indicates the direction for each component of quaternionic weight update. The main aim of ℍ-RPROP is to eliminate the value which is a little increased by constant increasing factor in order to accelerate convergence in shallow regions. ℍ-RPROP computes an individual delta for each connection of the network, which determines the size of weight update. Therefore, the faster convergence and higher accuracy are the main key features of proposed algorithm. The intelligent behavior of the proposed learning approach is demonstrated through a wide spectrum of prediction problems with different statistical performance evaluation metrics. In order to illustrate the learning and generalization of 3D motion as its inherent behavior, a solid set of experiments is presented where the training is performed through input-output mapping over a line and the generalization ability is verified over various non-linear geometrical objects. The slow convergence problem of back-propagation algorithm has been well combated by ℍ-RPROP. It has always demonstrated drastic reduction in the training cycles.

Keywords: Quaternion, Quaternionic resilient propagation, Time series prediction, Linear transformation, 3D motion.

1. Introduction

The processing of high dimensional information through conventional real-valued neural network (RVNN) is challenging, but a vital task because it requires a large number of neurons in network [1]. The structure of RVNN is not only complicated and unnatural, but it also leads to very slow processing and incapable to learn phase information among the intended components in high dimensions. This problem has been overcome by the extension of RVNN to complex-valued neural network (CVNN) for achieving the computational power of two dimensional information [2-4]. Further, the computational power of CVNN is achieved by the various methodologies such as the combination of gradient-descent and random-search algorithm [5], along with development of higher order neurons [6, 7]. But, the field of neural network with quaternion for three or four dimension problems is still not investigated thoroughly. The quaternionic-valued neural network (QVNN) will not only efficiently solve the difficult learning and generalization problems in high dimension with lesser number of neurons, but will yield a natural way of processing of high dimension information. Although, the conventional learning can be achieved by error backpropagation (BP) methodology in quaternionic domain, but the basic drawbacks of this method is still prevailing in high dimension [3, 5]. Therefore, we propose a quaternionic domain resilient

DOI: 10.22266/ijies2017.0831.22

propagation algorithm (ℍ-RPROP) for the faster convergence with better accuracy. The successive weight updates during learning in BP depends on the value of derivatives of error function. The basic principle in ℍ-RPROP algorithm is to eliminate the harmful influence of the size of partial derivatives of error function with respect to quaternionic weight, and adaptation is made dependent on the sign of its derivative. ℍ-RPROP is based on the signs of successive partial derivatives of error function with respect to each component of quaternionic weight update. The basic idea of resilient propagation algorithm [9, 10] is further boosted by error-dependent weight backtracking step in the quaternionic weights updation. This correction will not only accelerates the training speed appreciably and provide better approximation accuracy but also establish the logical way of adaptation for the proposed ℍ-RPROP algorithm.

The processing in quaternionic-valued neural network (QVNN) is as simple as conventional neural networks and error back-propagation algorithm in quaternionic domain (ℍ-BP) [11] has been obtained using the concept of gradient-decent optimization. The superiority and capability of QVNN are verified by the recent publications in the area of inverse kinematics of robot manipulator [12], nonlinear adaptive filtering [13], spoken language understanding [14], PolSAR land classification [15]. In QVNN, the split type activation functions is used instead of analytic because of its merit investigated in past for complex-valued neural networks [3, 7, 8]. The split type activation function, though, has a lack of analyticity concerned, appeared to be suitable tool for high dimension learning algorithm because of its boundedness. In this paper, the proposed ℍ-RPROP algorithm is compared with ℍ-BP through a wide spectrum of simulation experiments. The learning algorithm has an inherent ability to learn and generalize 3D motion because the signal flowing through the network is quaternion which possesses the values of different components and phase information is embedded among them. This strong point of the proposed method is further boosted by fast learning which results drastic reduction in training epochs. The main advantages of the results over others are clearly demonstrated through different statistical parameters in predication problems. The new features of the proposed ℍ-RPROP algorithm is to learn 3D motion or transformation through a set of points lying on a line which further emulates the generalization over complicated geometrical objects in space. In real life applications, the different transformations facilitate the viewing of 3D objects from different orientations as well as their motion interpretation in space. The proposed ℍ-RPROP algorithm has demonstrated its superiority over ℍ-BP in all perspectives.

The rest of the paper is organized as follows: Section 2 formally presents the proposed resilient propagation algorithm in quaternionic domain and also explains the theoretical reason why the proposed technique is better than existing techniques. The Section 3 mainly focuses on a comparative analysis through variety of statistical parameters of the proposed algorithm over ℍ-BP with a wide spectrum of prediction problems. Section 4 presents a solid set of experiments to demonstrate the capability of proposed ℍ-RPROP algorithm in learning and generalization of motion or transformations. The final inferences and discussion is given in the section 5.

2. Fast learning in quaternionic domain

The resilient propagation learning algorithm in real domain [10] and in complex domain [7, 9] are very popular to exhibit the fast and robust learning scheme because the only sign of the partial derivatives in successive steps is used to perform adaptation. It accomplishes the local adaptation of weight updates (Δw) according to the nature of error at each iteration, instead of value, to overcome the disadvantages of pure gradient descent approach [10]. A quaternion is a four-dimensional hyper-complex number system discovered by Hamilton [16]. It is capable to treat and operate three or four dimensional vector as one entity where phase information is imbedded within it. Therefore, it is highly desirable to propose a resilient propagation learning algorithm so that fast and effective information processing can be achieved through the operations on quaternionic variables. In this paper a three layer feed-forward network $(L - M - N)$ is considered in quaternionic domain that employs L inputs, M and N neurons in hidden and output layers respectively. All inputs, outputs, weights and biases are quaternion. Let quaternion $\boldsymbol{q}_l$ be the l^{th} $(l = 1 \ldots L)$ input at the input layer of the network, presented as

$$\boldsymbol{q}_l = \mathfrak{R}(\boldsymbol{q}_l) + \mathfrak{I}_1(\boldsymbol{q}_l)\boldsymbol{i} + \mathfrak{I}_2(\boldsymbol{q}_l)\boldsymbol{j} + \mathfrak{I}_3(\boldsymbol{q}_l)\boldsymbol{k} \quad (1)$$

where, $\mathfrak{R}$ and $\mathfrak{I}_i$ $(i = 1, 2, 3)$ denote the real and imaginary parts of $\boldsymbol{q}_l$. The bases $(\boldsymbol{i}, \boldsymbol{j}$ and $\boldsymbol{k})$ follow the properties [16] $\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1$ and $\boldsymbol{ij} = -\boldsymbol{ji} = \boldsymbol{k},\ \boldsymbol{jk} = -\boldsymbol{kj} = \boldsymbol{i},\ \boldsymbol{ki} = -\boldsymbol{ik} = \boldsymbol{j}$.

Let $\boldsymbol{V}$ is internal potential and f is a suitable activation function. The output of $m^{th}(m = 1 \ldots M)$ hidden neuron can be given as

$$\boldsymbol{Y}_m = f_{\mathbb{H}}(\boldsymbol{V}_m) = f_{\mathbb{H}}\left(\sum_{l=1}^{L} \boldsymbol{w}_{lm} \otimes \boldsymbol{q}_l + \boldsymbol{\alpha}_m\right) \quad (2)$$

where, symbols $\otimes$, $\boldsymbol{w}_{lm}$ and $\boldsymbol{\alpha}_m$ denote quaternion multiplication which follow the Hamilton properties as mentioned above, the synaptic weight that connects l^{th} input to m^{th} hidden neurons and bias of the m^{th} hidden neuron respectively. A split type function in quaternionic domain can be expressed as

$$f_{\mathbb{H}}(\boldsymbol{V}_m) = f(\Re(\boldsymbol{V}_m)) + f(\Im_1(\boldsymbol{V}_m))\boldsymbol{i} + f(\Im_2(\boldsymbol{V}_m))\boldsymbol{j} + f(\Im_3(\boldsymbol{V}_m))\boldsymbol{k} \quad (3)$$

Similarly, the output of $n^{th}(n = 1 \ldots N)$ output neuronwith internal potential (V_n)

$$\boldsymbol{Y}_n = f_{\mathbb{H}}(\boldsymbol{V}_n) = f_{\mathbb{H}}\left(\sum_{m=1}^{M} \boldsymbol{w}_{mn} \otimes \boldsymbol{Y}_m + \boldsymbol{\beta}_n\right) \quad (4)$$

where, $\boldsymbol{w}_{mn}$ and $\boldsymbol{\beta}_n$ denote the synaptic weight and bias in output layer respectively.

2.1 Backpropagation learning in quaternionic domain

The gradient based error backpropagation learning for feed-forward network in quaternionic domain is recapitulated here. Let the error e_n be the difference between desired Y_n^D and actual Y_n output of n^{th}output neuron, then

$$e_n = Y_n^D - Y_n = \Re(e_n) + \Im_1(e_n)i + \Im_2(e_n)j + \Im_3(e_n)k \quad (5)$$

The weight update equation can be obtained by minimizing the mean square error (MSE) function

$$E = \frac{1}{2N}\sum_{n=1}^{N}(e_n \otimes \bar{e}_n) \quad (6)$$

where, $\overline{(.)}$ denotes the quaternion conjugate. The update rules for weight and bias at output layer can be computed by a gradient of error function as

$$\Delta w_{mn} = -\eta\nabla_{(w_{mn})}E = -\eta\left(\frac{\partial E}{\partial\Re(w_{mn})} + \frac{\partial E}{\partial\Im_1(w_{mn})}\boldsymbol{i} + \frac{\partial E}{\partial\Im_2(w_{mn})}\boldsymbol{j} + \frac{\partial E}{\partial\Im_3(w_{mn})}\boldsymbol{k}\right) = \frac{\eta}{N}\left(e_n \odot f'_{\mathbb{H}}(V_n)\right)\otimes\bar{Y}_m \quad (7)$$

$$\Delta\beta_n = -\eta\nabla_{(\beta_n)}E = \frac{\eta}{N}e_n \odot f'_{\mathbb{H}}(V_n) \quad (8)$$

where, $\eta \in \mathbb{R}^+$denotes learning rate and $\odot$ denotes the component wise multiplication. The derivative of quaternion-split type function $f'_{\mathbb{H}}(V_n)$ is given as

$$f'_{\mathbb{H}}(V_n) = f'\left(\Re(V_n)\right) + f'\left(\Im_1(V_n)\right)\boldsymbol{i} + f'\left(\Im_2(V_n)\right)\boldsymbol{j} + f'\left(\Im_3(V_n)\right)\boldsymbol{k} \quad (9)$$

The weight and bias update rules at hidden layer:

$$\Delta w_{lm} = -\eta\nabla_{(w_{lm})}E = \frac{\eta}{N}\left(\left(\sum_{n=1}^{N}\bar{w}_{mn}\otimes\left(e_n \odot f'_{\mathbb{H}}(V_n)\right)\right) \odot f'_{\mathbb{H}}(V_m)\right)\otimes\bar{q}_l \quad (10)$$

$$\Delta\alpha_m = -\eta\nabla_{(\alpha_m)}E = \frac{\eta}{N}\left(\sum_{n=1}^{N}\bar{w}_{mn}\otimes\left(e_n \odot f'_{\mathbb{H}}(V_n)\right)\right) \odot f'_{\mathbb{H}}(V_m) \quad (11)$$

2.2 Resilient propagation algorithm in quaternionic domain

Resilient propagation in quaternionic domain (ℍ-RPROP) depends only on the sign of gradient of error function that indicates the direction of weight update. The main aim of ℍ-RPROP is to find weight update by a constant factor in order to accelerate the convergence in shallow regions. It computes an individual update value (Δ) for each connection of the network, which determines the size of weight update. The aim of this paper is to present a resilient propagation algorithm in quaternionic domain (ℍ-RPROP) with error-dependent weight backtracking step. The theoretical reason for the need of error-dependent weight backtracking step and why the proposed technique is better than the existing one can be logically explained by the examination of successive partial derivatives. If the signs of successive partial derivatives are both positive that means error is increasing then the weight is decreased by the update value. Similarly, if the signs of successive partial derivatives are both negative that means error is decreasing then the weight is increased by the update value. If the signs of successive partial derivatives are opposite that means a minima is missing in between, then the weight is reverted to previous state and the previous partial derivatives is set to be zero otherwise updated weight will reflect the same changes repeatedly. Thus ℍ-RPROP algorithm is boosted by error-dependent weight backtracking step in which accelerates the training speed appreciably and also provides better approximation accuracy. Thus, the faster convergence and higher accuracy are the main key features of the proposed algorithm.

The sign of partial derivatives of the error function with respect to each part of quaternionic weight determines the direction of weight update $\mathfrak{R}\,(\Delta w(t))$, $\mathfrak{I}_1(\Delta w(t))$, $\mathfrak{I}_2(\Delta w(t))$ and $\mathfrak{I}_3(\Delta w(t)$. Each component of quaternionic weight is modified by an amount Δ (update value) with a view to decrease the overall error in learning cycles. Each update value determines the size of weight update. The $\mathbb{H}$-RPROP algorithm uses various parameters, initial step size (Δ_0), increase factor (μ^+), decrease factor (μ^-), minimum step size (Δ_{min}) and maximum step size (Δ_{max}) to initiate the process. The step size (update value) is modified according to gradient direction. For the sake of simplicity a generalized symbol $\mathfrak{G}$ is used to denote the all parts of quaternionic weight. The weight updates are determined individually as follows

$$\mathfrak{G}(\Delta\boldsymbol{w}(t)) = -\text{sign}\left(\frac{\partial E(t)}{\partial \mathfrak{G}(\boldsymbol{w})}\right)\mathfrak{G}(\Delta(t)) \tag{12}$$

The proposed learning algorithm depends on the signs of successive partial derivatives of the error function with respect to each component of quaternionic weight update. If the signs of successive partial derivatives are both positive that means error is increasing then the weight is decreased by the update value. Similarly, if the signs of successive partial derivatives are both negative that means error is decreasing then the weight is increased by the update value, presented as

$$\begin{gathered}\text{if }\left(\frac{\partial E(t-1)}{\partial \mathfrak{G}(w)} \times \frac{\partial E(t)}{\partial \mathfrak{G}(w)} > 0\right)\text{ then}\\ \{\mathfrak{G}(\Delta(t)) = \min\left(\mathfrak{G}(\Delta(t-1)) \times \mu^+, \mathfrak{G}(\Delta_{max})\right);\\ \mathfrak{G}(w(t+1)) = \mathfrak{G}(w(t)) - \text{sign}\left(\frac{\partial E(t)}{\partial \mathfrak{G}(w)}\right)\mathfrak{G}(\Delta(t));\}\end{gathered} \tag{13}$$

The important element is to investigate whether the error is increasing or decreasing in case of change in sign of partial derivatives in successive steps, caused by weight update. When the partial derivatives for each part of error in successive step are opposite in sign and the overall error increases, then only the previous weight update is reverted as

$$\begin{gathered}\text{if }\left(\frac{\partial E(t-1)}{\partial \mathfrak{G}(w)} \times \frac{\partial E(t)}{\partial \mathfrak{G}(w)} < 0\right)\text{ then}\\ \{\mathfrak{G}(\Delta(t)) = \max\left(\mathfrak{G}(\Delta(t-1)) \times \mu^-, \mathfrak{G}(\Delta_{min})\right);\\ \text{if }(E(t) > E(t-1))\text{ then}\\ \mathfrak{G}(w(t+1)) = \mathfrak{G}(w(t)) - \mathfrak{G}(\Delta w(t-1));\\ \frac{\partial E(t)}{\partial \mathfrak{G}(w)} = 0\,;\}\end{gathered} \tag{14}$$

The abstract pseudo code of proposed $\mathbb{H}$-RPROP algorithm is as follows:

```
$t = 1;\ 0 < \mu^- < \mu^+ < 1.2$
$\forall$ parts of quaternion $\mathfrak{G}(\Delta(t)) = \Delta_0$; $\frac{\partial E(t-1)}{\partial \mathfrak{G}(w)} = 0$;
$\mathfrak{G}(\Delta_{max}) = \Delta_{max}$; $\mathfrak{G}(\Delta_{min}) = \Delta_{min}$
**Repeat** {$\forall$ parts of quaternionic weights, compute $\frac{\partial E(t)}{\partial \mathfrak{G}(w)}$
**Do** {If $\left(\frac{\partial E(t-1)}{\partial \mathfrak{G}(w)} \times \frac{\partial E(t)}{\partial \mathfrak{G}(w)} > 0\right)$ then
  {$\mathfrak{G}(\Delta(t)) = \min\left(\mathfrak{G}(\Delta(t-1)) \times \mu^+, \mathfrak{G}(\Delta_{max})\right)$;
  $\mathfrak{G}(w(t+1)) = \mathfrak{G}(w(t)) - \text{sign}\left(\frac{\partial E(t)}{\partial \mathfrak{G}(w)}\right)\mathfrak{G}(\Delta(t))$;}
If $\left(\frac{\partial E(t-1)}{\partial \mathfrak{G}(w)} \times \frac{\partial E(t)}{\partial \mathfrak{G}(w)} < 0\right)$ then
  {$\mathfrak{G}(\Delta(t)) = \max\left(\mathfrak{G}(\Delta(t-1)) \times \mu^-, \mathfrak{G}(\Delta_{min})\right)$;
  If $(E(t) > E(t-1))$ then
    $\mathfrak{G}(w(t+1)) = \mathfrak{G}(w(t)) - \mathfrak{G}(\Delta w(t-1))$;
    $\frac{\partial E(t)}{\partial \mathfrak{G}(w)} = 0$ ; }
If $\left(\frac{\partial E(t-1)}{\partial \mathfrak{G}(w)} \times \frac{\partial E(t)}{\partial \mathfrak{G}(w)} = 0\right)$ then
  $\mathfrak{G}(w(t+1)) = \mathfrak{G}(w(t)) - \text{sign}\left(\frac{\partial E(t)}{\partial \mathfrak{G}(w)}\right)\mathfrak{G}(\Delta(t))$;}
**While** (for all parts ($\mathfrak{G}(.)$) of quaternionic weight)
$t = t + 1$; } **until (converged)**.
```

3. Performance analysis of $\mathbb{H}$-RPROP through prediction problems

In order to estimate the learning and generalization effectiveness of proposed $\mathbb{H}$-RPROP algorithm; the performance evaluation has been carried out on different prediction and function approximation problems. All synaptic weights and biases of a three layer network are randomly initialized between -1 to 1 with bias input $(1 + i + j + k)$ in quaternionic domain. The hypertangent activation function is used for analysis purpose. An intelligent choice of RPROP parameters and weight initialization gives good results. The computational power and approximation capability have been compared in terms of number of epochs, learning parameters, network topology along with other statistical performance evaluation metrics like correlation, error variance, and Akaike's information criteria (AIC) [17]. The proposed $\mathbb{H}$-RPROP algorithm with error dependent weight backtracking step accelerates the training speed significantly with better approximation accuracy.

3.1 Chua's circuit as time series prediction

Chua's circuit is the simplest autonomous electronic circuit containing registers, capacitors and inductors that exhibit the chaotic behaviour [18].

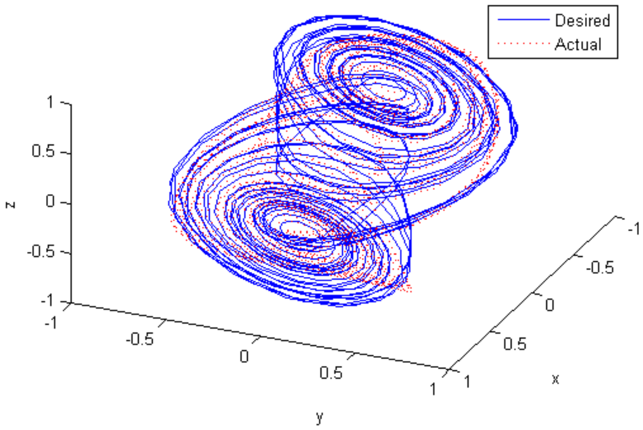


Figure.1 3D plot of Chua's circuit tested by the network trained by ℍ-BP

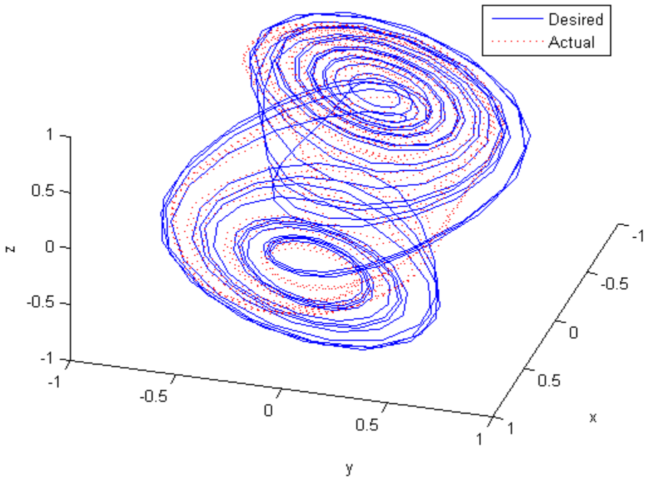


Figure.2 3D plot of Chua's circuit tested by the network trained by ℍ-RPROP

This circuit satisfies the chaotic criteria that contain one or more non-linear elements, one or more active registers and three or more energy storage devices.

The presented model uses the one Chua's diode as non-linear element, one locally active register and two capacitors and one inductor as energy storage devices, whose dynamics is given as state equations:

$$\frac{dx}{dt} = \alpha[y - x - h(x)]$$
$$\frac{dy}{dt} = x - y + z$$
$$\frac{dz}{dt} = -\beta y - \gamma z \quad (15)$$

where, $h(x)$ presents the electrical response of non-linear register and defined as

$$h(x) = m_1 x + \tfrac{1}{2}(m_0 - m_1)(|x+1| - |x-1|) \quad (16)$$

Table 1. Comparison of training and testing performance of Chua's circuit

	ℍ-BP	**ℍ-RPROP**
Network	1-4-1	1-4-1
Parameters	13	13
MSE Training	0.0074	0.0010
Average Epoch	12000	2100
MSE Testing	0.0075	0.0044
Error Variance	0.0038	0.0009
Correlation	0.9893	0.9910
AIC	-4.8843	-6.8858

where α, β, γ, m_0 and m_1 denote the constant parameters. The symbols x, y and z are the voltages across two capacitors and an inductor respectively, and combination of them shows the chaotic attractor in three dimensions. The double scrolled chaotic attractor is obtained with the parameters $\alpha = 15.6$, $\beta = 28$, $\gamma = 0$, $m_0 = -1.143$ and $m_1 = -0.714$. The chaotic time series has been obtained from the simulation of the system (Eq. 15 and Eq. 16) with time step 0.1 Sec and initial voltages x=0.1, $y = 0.1$ and $z = 0.1$. The normalized input-output values in the range -0.8 to 0.8 are considered in imaginary quaternion (real part is set to zero).

A time series containing 500 terms obtained from simulated system has been used to train a three layer neural network (1-4-1) in quaternion domain using ℍ-BP and ℍ-RPROP algorithms. Table 1 demonstrates that the proposed ℍ-RPROP algorithm requires a significantly lesser number of training epochs than ℍ-BP algorithm. The next 500 terms of the series have been tested through trained network by both the algorithms. The 3D plot in Fig. 1 and Fig. 2 present the comparisons of desired and actual pattern for Chua's circuit. The superiority of ℍ-RPROP over ℍ-BP is demonstrated through different statistical parameters like error variance, correlation, and AIC in Table 2. This experiment reveals the better approximation capability of proposed technique over ℍ-BP.

3.2 Lorenz system as time series prediction

The chaotic behaviour of Lorenz's system [19] is presented by the system of differential equations:

$$\frac{dx}{dt} = \sigma(y - x)$$
$$\frac{dy}{dt} = x(\rho - z) - y$$
$$\frac{dz}{dt} = xy - \beta z \quad (17)$$

where, the symbols σ, ρ and β denote the Lorenz's parameters. Let $\sigma = 15$, $\rho = 28$ and $\beta = 8/3$, the system (Eq. (17)) generates 6537 terms of the time series in 100 seconds with initial condition ($x = 0.7$, $y = 0.1$, $z = 0.1$); which are further normalized in the range -0.8 to 0.8.

The first 500 terms of the time series have been used to train the networks using ℍ-BP and ℍ-RPROP learning algorithms and the rest of the terms used for testing. The Table 2 presents better performance by ℍ-RPROP algorithm in comparison to ℍ-BP with drastic reduction in training epochs.

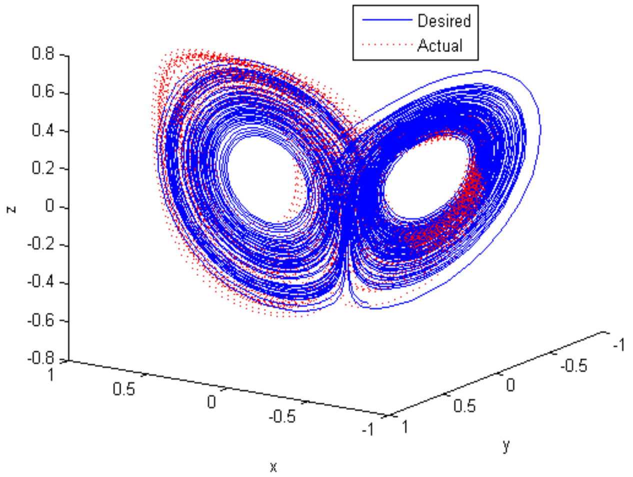


Figure.3 3D plot of the Lorenz system tested by the network trained through ℍ-BP

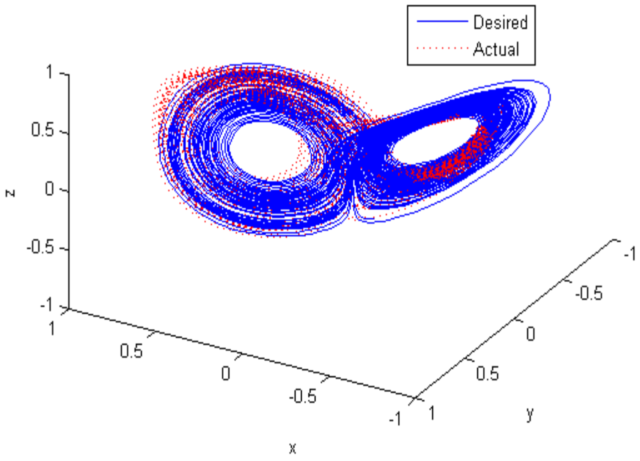


Figure.4 3D plot of the Lorenz system tested by the network trained through ℍ-RPROP learning

Table 2. Comparison of training and testing performance for Lorenz system

	ℍ-BP	**ℍ-RPROP**
Network	1-4-1	1-4-1
Parameters	13	13
MSE Training	0.0015	0.0006
Average Epoch	11000	3100
MSE Testing	0.0024	0.0008
Error Variance	0.0016	0.0004
Correlation	0.9871	0.9982
AIC	-6.4503	-7.3666

The testing results for the prediction of Lorenz time series by ℍ-BP and ℍ-RPROP are presented in Fig. 3 and Fig. 4 respectively. The results again reveal the excellent superiority of ℍ-RPROP over ℍ-BP.

3.3 Channel equalization problem

The equalization problem establishes the assessment of input signals using received signals $q(n)$ and desired delayed signals $q(n-\tau)$. An experiment is conducted here to evaluate the comparative performance of ℍ-BP and ℍ-RPROP learning algorithms for highly complex non-linear quaternionic-valued channel model, defined as:

$$y(n) = O(n) + 0.1[O(n)]^2 + 0.05[O(n)]^3 + \vartheta_n, \quad \vartheta_n \sim N(1, 0.1569) \qquad (18)$$

where,

$$O(n) = (0.34 - i * 0.27 + j * 0.50 + k * 0.62) \otimes q(n) + (0.87 + i * 0.43 + j * 0.72 + k * 0.22) \otimes q(n-1) + (0.34 - i * 0.21 + j * 0.64 + k * 0.44) \otimes q(n-2) \qquad (19)$$

In this experiment, the transmitted signals are 4-QAM with input constellations $0.7 * (0 \pm i \pm j + k * 0)$ and in equalizer the received signal are in one of the four possible classes. The equalization model is of order three with decision delay $\tau = 1$. The 200 samples with 16 dB SNR from Eq. 18 and 19 were considered for the training of equalizer through ℍ-BP and ℍ-RPROP algorithms. The 1000 samples for testing set by the noisy channel distribution are presented in Fig. 5.

In order to demonstrate the effectiveness of channel equalization using different algorithms, the equalizer output distribution of the test set is presented by eye diagram in Fig. 6 and Fig. 7. The relative analysis through different parameters presented in Table 3 demonstrates the outperformance of ℍ-RPROP algorithm along with significant faster learning. The superiority of ℍ-RPROP is again observed in this experiment.

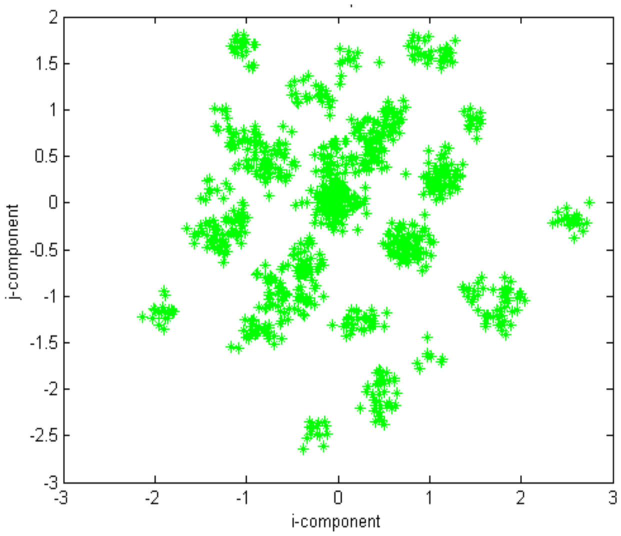


Figure.5 A test set of input signals

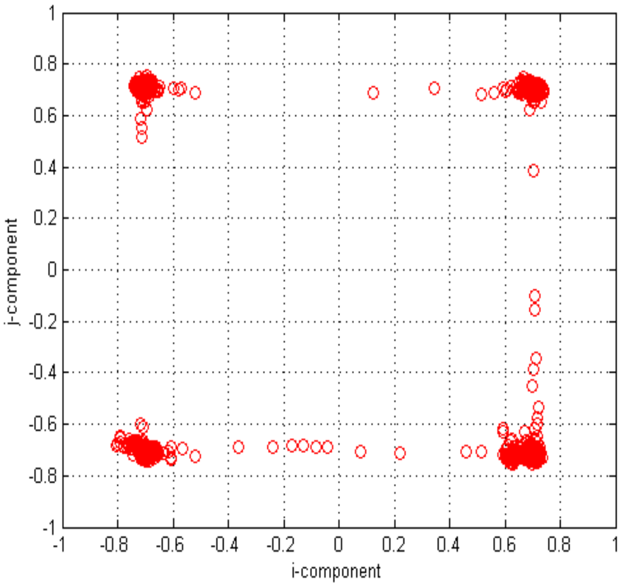


Figure.6 Eye diagram of the outputs of the ℍ-BP algorithm based equalizer

Table 3. Comparison of equalizers based on ℍ-BP and ℍ-RPROP algorithms

	ℍ-BP	**ℍ-RPROP**
Network	3-14-1	3-14-1
Parameters	71	71
MSE Training	0.0016	0.0010
Average Epoch	30000	9000
MSE Testing	0.0005	0.0003
Error Variance	0.0013	0.0006
Correlation	0.9945	0.9978
AIC	-6.4555	-6.9111

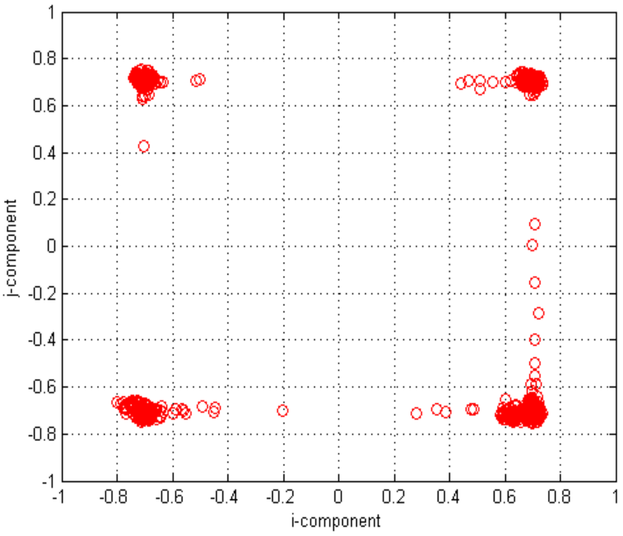
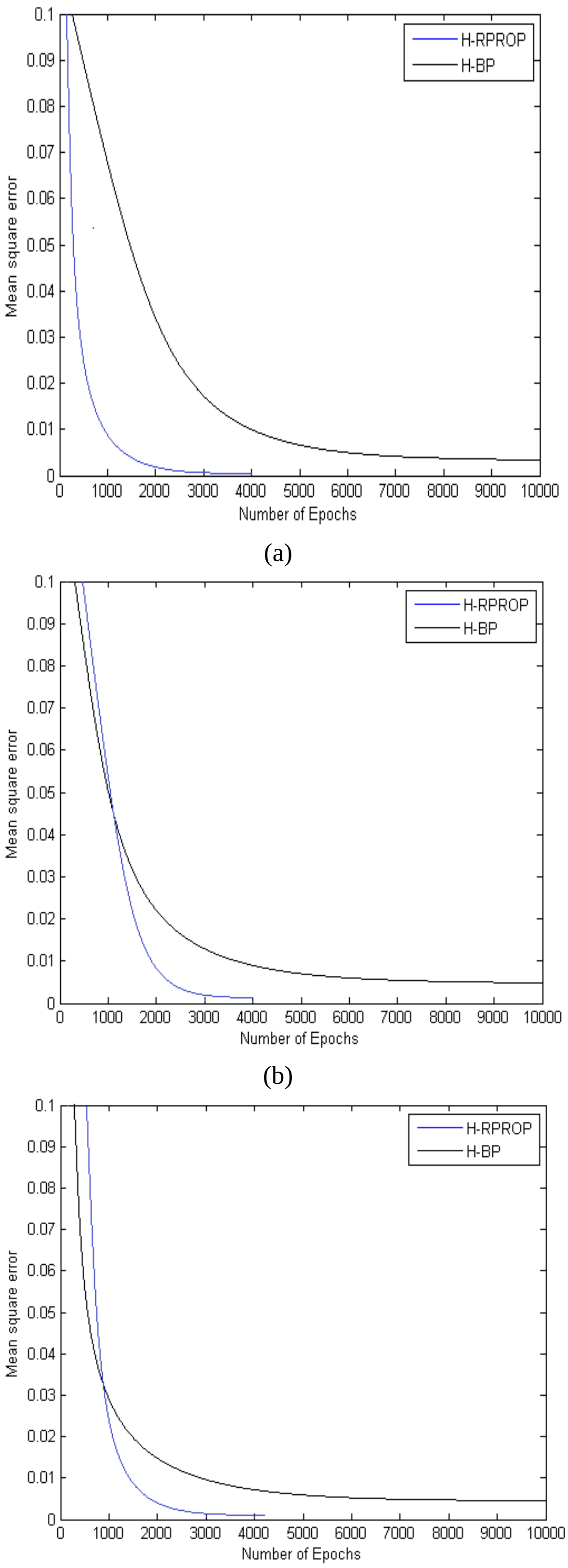


Figure.7 Eye diagram of the outputs of the ℍ-RPROP algorithm based equalizer

4. Performance analysis of ℍ-RPROP through transformations

In order to compare training and testing performance of proposed ℍ-RPROP algorithm with ℍ-BP for three dimensional motion or transformation, we consider a three layer (2-6-2) QVNN which contains two inputs, six hidden neurons and two output neurons. For training, an input-output mapping over a straight line and a reference point (middle point of the line) has been considered in all experiments. First input on the network takes a set of point lying on a straight line and second input passes the reference point. The learning of a three layer network is done by both algorithms for different class of transformations. The input-output training mapping is used for three cases of transformations; scaling factor ½ in first case; scaling with factor ½ followed by 0.3 unit translation along the positive z-direction in second case; and scaling factor ½ followed by 0.3 unit translation along the positive z-direction and $\pi/2$ radian rotation around the unit vector $(\boldsymbol{i})$ in third case. All the mappings are defined over straight line containing 21 points and referenced at (0, 0, 0). The error convergence curves in Fig. 8 compare the learning process of proposed ℍ-RPROP with ℍ-BP in above three cases of transformations. The training of network through ℍ-RPROP considers parameters, initial step size ($\Delta_0 = 0.01$), increase factor ($\mu^+ = 1.2$), decrease factor ($\mu^- = 0.5$), minimum step size ($\Delta_{min} = 10^{-6}$) and maximum step size ($\Delta_{max} = 0.005$). All weights and biases are initialized in the range -0.05 to +0.05.

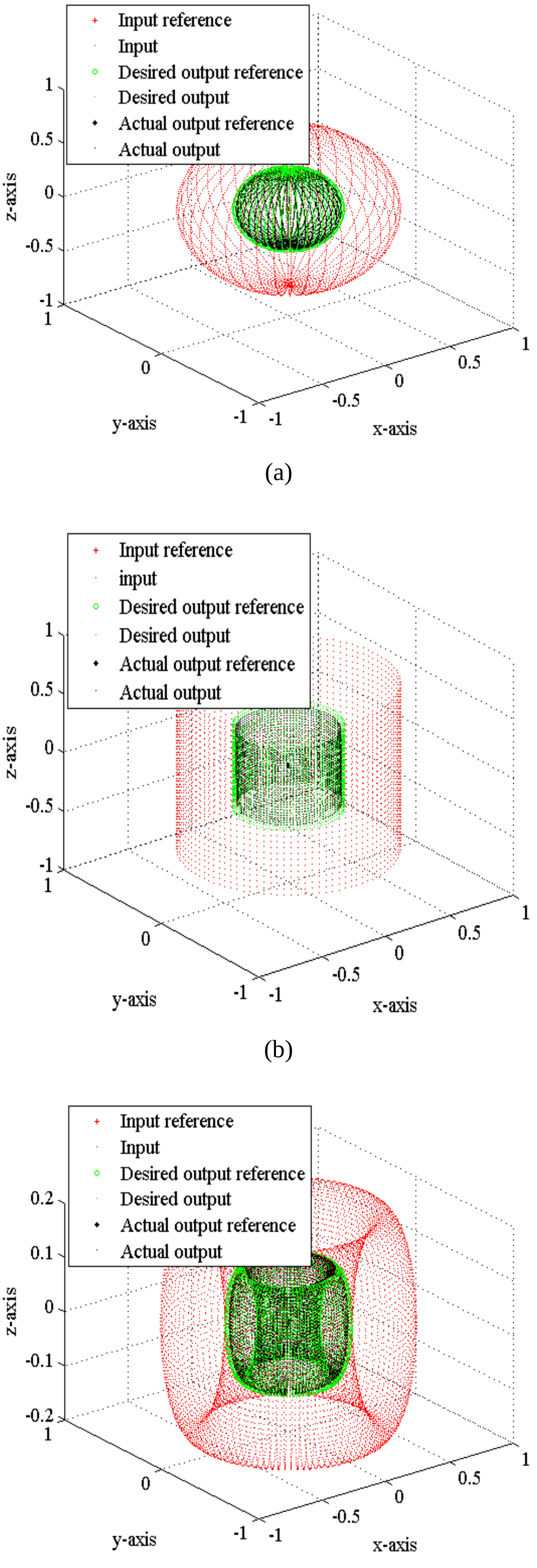


(a)

(b)

(c)

Figure.8 The MSE convergence through ℍ-BP and proposed ℍ-RPROP learning algorithm for: (a) scaling, (b) scaling and translation (c) scaling, translation and rotation

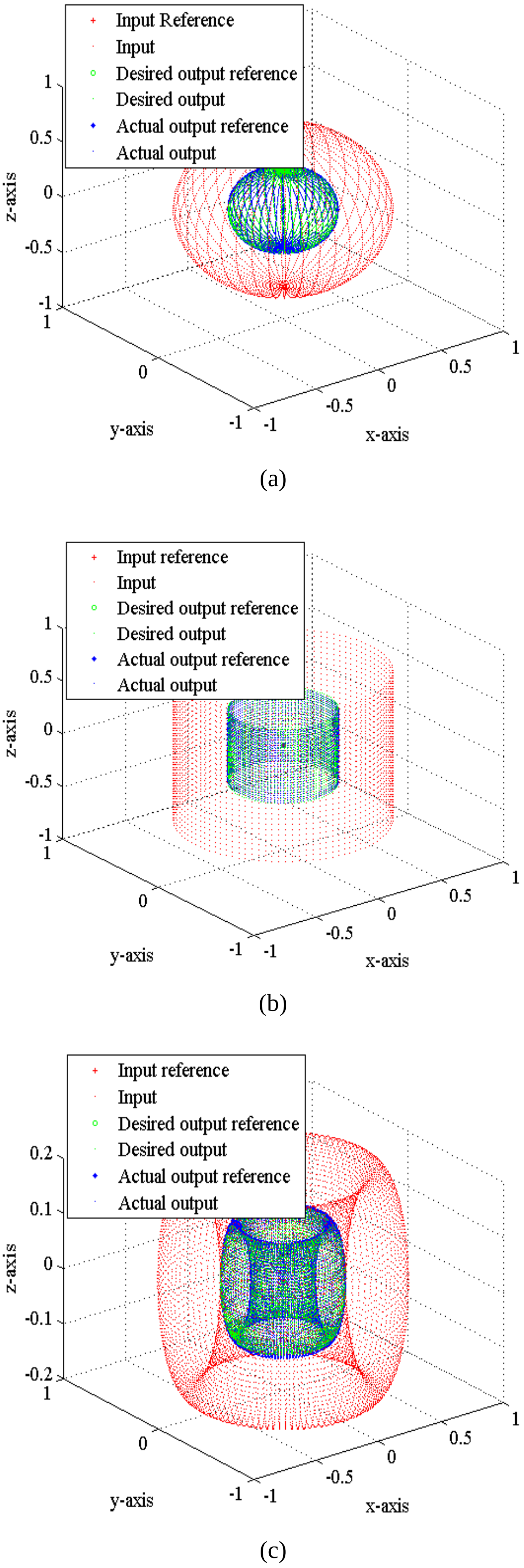


(a)

(b)

(c)

Figure.9 The generalization through ℍ-BP algorithm: Transformation with scaling factor ½ over (a) Sphere (b) Cylinder (c) Torus

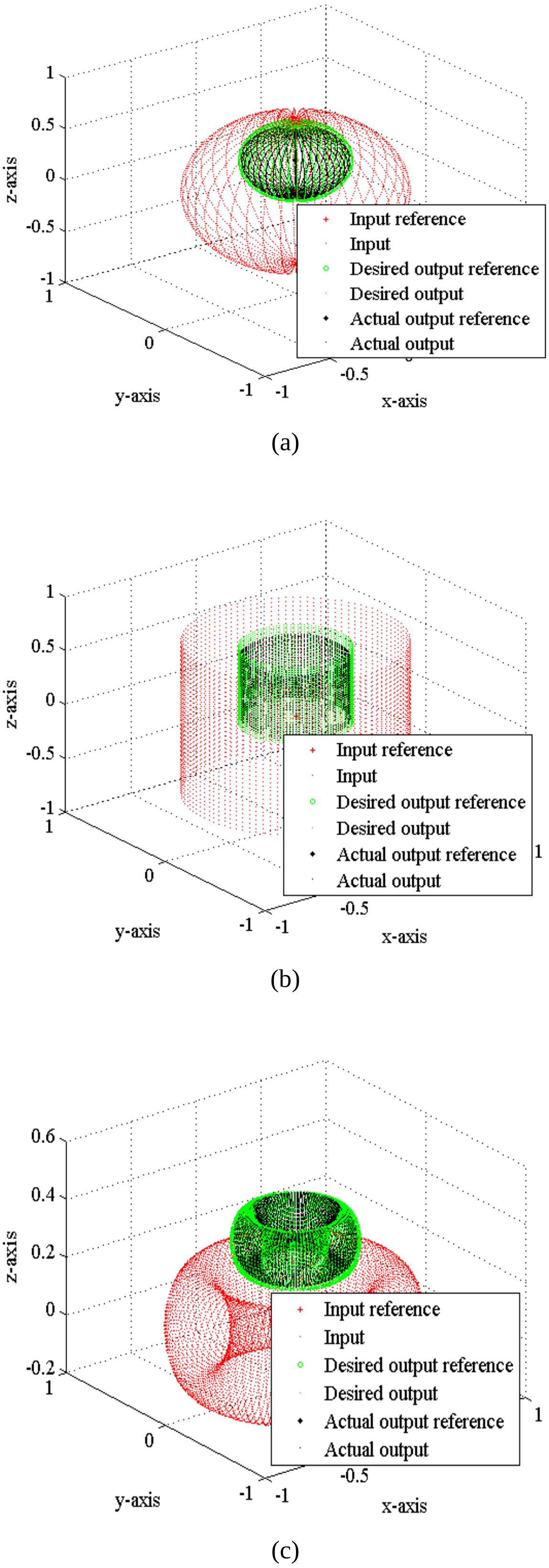


(a)

(b)

(c)

Figure.10 The generalization through ℍ-RPROP algorithm: Transformations with scaling factor ½; over (a) Sphere (b) Cylinder (c) Torus

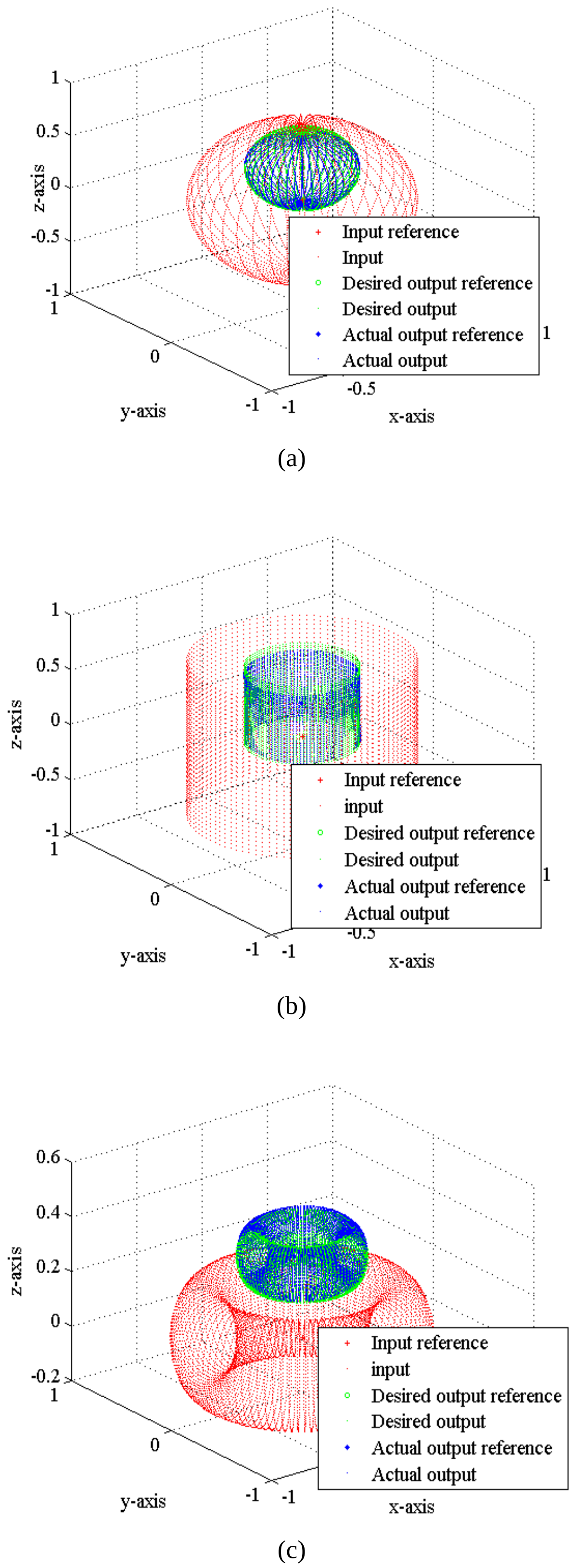


(a)

(b)

(c)

Figure.11 The generalization through ℍ-BP algorithm: Transformations with scaling factor ½ and 0.3 unit translation in positive z-direction; over (a) Sphere (b) Cylinder (c) Torus

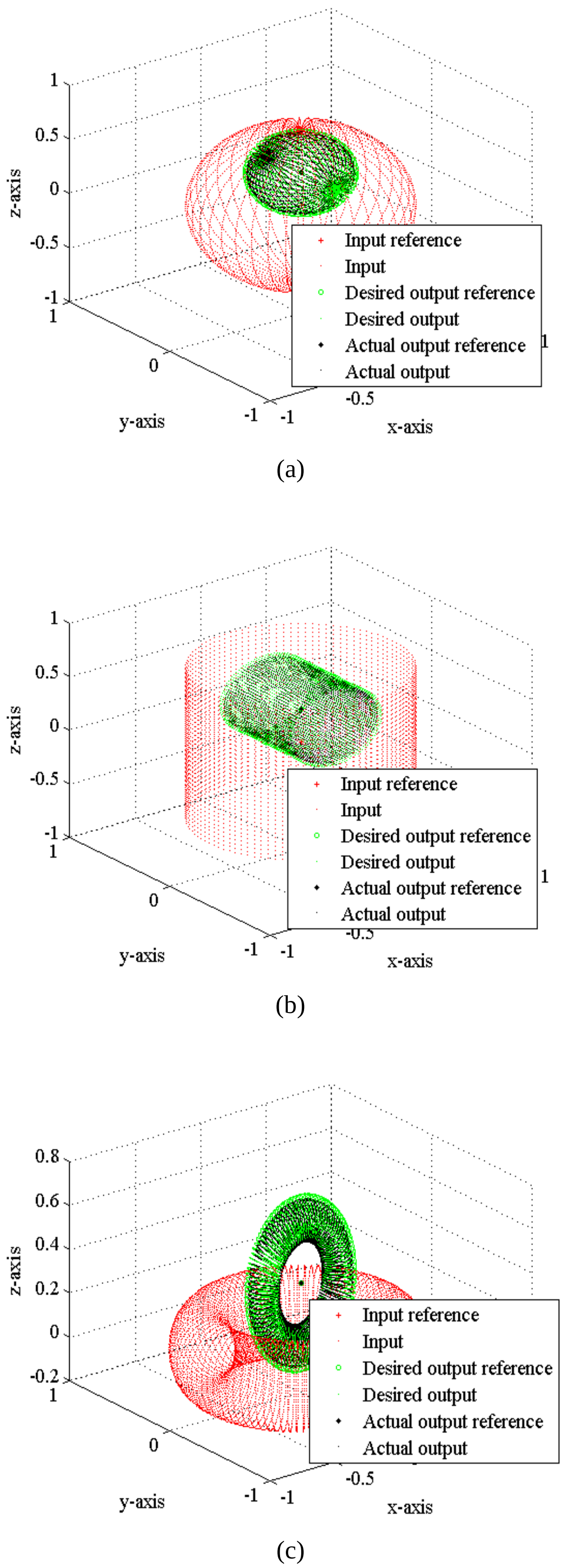


(a)

(b)

(c)

Figure.12 The generalization through ℍ-RPROP algorithm: Transformations with scaling factor ½ and 0.3 unit translation in positive z-direction; over (a) Sphere (b) Cylinder (c) Torus

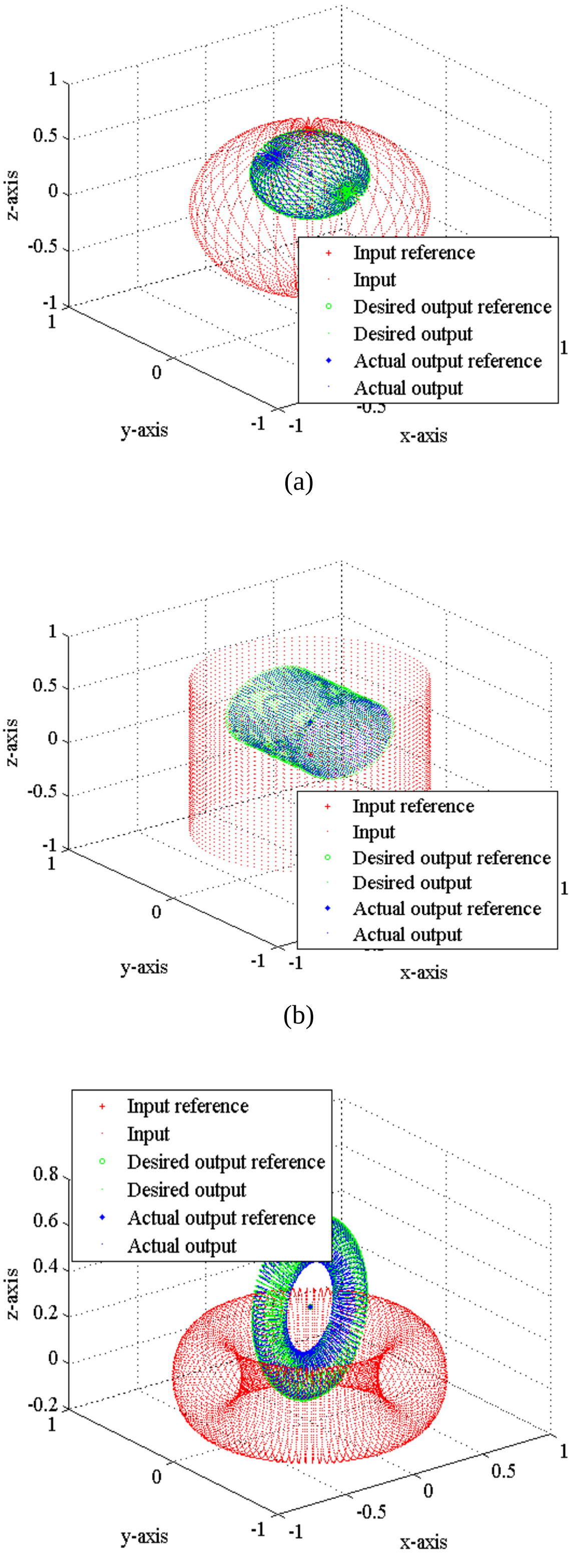


(a)

(b)

(c)

Figure.13 The generalization through ℍ-BP algorithm: Transformations with scaling factor ½, 0.3 unit translation in positive z-direction, and π/2 radian rotation around the unit vector (***i***); over (a) Sphere (b) Cylinder (c) Torus



(a)

(b)

(c)

Figure.14 The generalization through ℍ-RPROP algorithm: Transformations with scaling factor ½, 0.3 unit translation in positive z-direction, and π/2 radian rotation around the unit vector (***i***); over (a) Sphere (b) Cylinder (c) Torus

The hyperbolic tangent function is used as activation function. The different networks converge significantly faster in case of ℍ-RPROP than ℍ-BP ($\eta = 0.001$); as shown in Fig. 8.

The testing of trained networks through both algorithms has been done over complicated 3D objects like sphere (4141 data points), cylinder (2929 data points) and torus (10201 data points). The generalizations over complicated 3D objects for networks trained by ℍ-RPROP in a different class of transformations are presented in Fig. 10, Fig. 12 and Fig. 14 respectively; and for networks trained by ℍ-BP is presented in Fig. 9, Fig. 11 and Fig. 13. Testing processes demonstrate the accuracy and superiority of ℍ-RPROP over ℍ-BP and also reveal its effective generalization ability. The ℍ-RPROP drastically reduces the number of training epochs with comparable and better generalization accuracy in all classes of transformations. The theoretical reason for fast convergence lie behind the basic approach of ℍ-RPROP algorithm which considers only sign of partial derivatives to indicate the direction of weight update and eliminate the value which is a little increased by constant increasing factor in order to accelerate convergence in shallow regions. Since, quaternion is the unit of learning in quaternionic-valued neural networks therefore it has ability to learn different intended components in high dimension along with their phase information while any real-valued neural network does not have.

5. Inferences and discussions

In this paper, we propose a fast and efficient learning algorithm ℍ-RPROP (resilient propagation in quaternionic domain); and its superiority over back-propagation algorithm (ℍ-BP) has been verified through a wide spectrum of benchmark problems. The ℍ-RPROP algorithm not only reduces the number of training epochs drastically, but presents its outperformance through different statistical performance evaluation metrics like correlation, error variance, and Akaike's information criteria in all prediction problems.

The extremely faster convergence of the proposed algorithm is also seen in case of different classes of transformations. The ℍ-RPROP algorithm smartly learns the different compositions of basic transformations (translation, scaling, and rotation) through input-output mapping over straight line. The trained network is well capable to generalize the considered transformations over non-linear geometrical structures like sphere, cylinder, and torus possessing huge point cloud dataset. It presents the intelligent behaviour in motion interpretation in space as it also maintains phase information in learning and subsequent generalization. The proposed ℍ-RPROP algorithm will definitely provide a new direction to prospective researchers in various scientific and engineering applications for 3D and 4D datasets where faster training and higher accuracy are required.

References

[1] B. K. Tripathi, *High Dimensional Neurocomputing: Growth, Appraisal and Applications*, Springer, Landon, 2015.

[2] T. Nitta, "An Extension of the Back-Propagation Algorithm to Complex Numbers", *Neural Networks*, Vol. 10, No. 8, pp. 1391-1415, 1997.

[3] A. Hirose, *Complex-Valued Neural Networks*, Springer-Verlag, New York, 2006.

[4] M. F. Amin and K. Murase, "Single-Layered Complex-Valued Neural Network for Real-Valued Classification Problems", *Neurocomputing*, Vol. 72, No. 4, pp. 945-955, 2008.

[5] H. G. Zimmermann, A. Minin, and V. Kusherbaeva, "Comparison of the Complex Valued and Real Valued Neural Networks Trained with Gradient Descent and Random Search Algorithms", In: *Proc. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning"*, Bruges, Belgium, April, 2011.

[6] B. K. Tripathi and P. K. Kalra, "The Novel Aggregation Function-based Neuron Models in Complex Domain", *Soft Computing*, Vol. 14, No. 10, pp. 1069-1081, 2010.

[7] B. K. Tripathi and P. K. Kalra, "On Efficient Learning Machine with Root Power Mean Neuron in Complex Domain", *IEEE Trans. on Neural Netw.*, Vol. 22, No. 05, pp. 727-738, 2011.

[8] B. K. Tripathi, "On the Complex Domain Deep Machine Learning for Face Recognition", *Applied Intelligence*, Springer, DOI: 10.1007/s10489-017-0902-7, 2017.

[9] A. Kantsila, M. Lehtokangas, and J. Saarinen, "Complex RPROP-algorithm for neural network equalization of GSM data bursts", *Neurocomputing*, Vol. 61, pp. 339-360, Oct. 2004.

[10] C. Igel, and M. Husken, "Empirical Evaluation of the Improved Rprop Learning Algorithms", *Neurocomputing*, Vol. 50, No. pp. 105-123, 2003.

[11] T. Nitta, "A Quaternary Version of the Back-Propagation Algorithm", In: *Proc. International Conference on Neural Networks*, Perth, WA, Vol.5, pp. 2753-2756, 1995.

[12] Y. Cui, K. Takahashi, and M. Hashimoto, "Design of Control Systems using Quaternion Neural Network and Its Application to Inverse Kinematics of Robot Manipulator", In: *Proc. IEEE/SICE Int. Symposium on System Integration*, Kobe, Japan, pp. 527-532, Dec. 2013.

[13] B. Ujang, C.C. Took, and Mandic D. P., "Quaternion-Valued Nonlinear Adaptive Filtering", *IEEE Trans. on Neural Netw.*, Vol. 22, No. 8, pp. 1193-1206, 2011.

[14] T. Parcollet, M. Morchid, P. M. Bousquet, R. Dufour, G. Linarès and R. De Mori, "Quaternion Neural Networks for Spoken Language Understanding", In: *Proc. IEEE Spoken Language Technology Workshop (SLT)*, San Diego, CA, pp. 362-368, 2016.

[15] F. Shang and A. Hirose, "Quaternion Neural-Network-Based PolSAR Land Classification in Poincare-Sphere-Parameter Space", *IEEE Trans. on Geoscience and Remote Sensing*, Vol 52, No. 9, pp. 5693-5703, Sept. 2014.

[16] W. R. Hamilton, *Lectures on Quaternions*, Hodges and Smith: Dublin, Ireland, 1853.

[17] D. B. Foggel, "An Information Criterion for Optimal Neural Network Selection", *IEEE Trans. Neural Netw.*, Vol. 2, No. 5, pp. 490-497, Sep.1991.

[18] L. O. Chua, T. Matsumoto, and M. Komuro, "The Double Scroll", *IEEE Transactions on Circuits and Systems*, Vol. CAS-32, No. 8, pp. 798-818, August 1985.

[19] E. N. Lorenz, "Deterministic Nonperiodic Flow", *Journal of the Atmospheric Sciences*, Vol. 20, No. 2, pp. 130-141, 1963.