World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:2, No:1, 2008

Complex-Valued Neural Network in Signal Processing: A Study on the Effectiveness of Complex Valued Generalized Mean Neuron Model

Anupama Pande, Ashok Kumar Thakur, and Swapnoneel Roy

Abstract**—**A complex valued neural network is a neural network which consists of complex valued input and/or weights and/or thresholds and/or activation functions. Complex-valued neural networks have been widening the scope of applications not only in electronics and informatics, but also in social systems. One of the most important applications of the complex valued neural network is in signal processing. In Neural networks, generalized mean neuron model (GMN) is often discussed and studied. The GMN includes a new aggregation function based on the concept of generalized mean of all the inputs to the neuron. This paper aims to present exhaustive results of using Generalized Mean Neuron model in a complex-valued neural network model that uses the back-propagation algorithm (called 'Complex-BP') for learning. Our experiments results demonstrate the effectiveness of a Generalized Mean Neuron Model in a complex plane for signal processing over a real valued neural network. We have studied and stated various observations like effect of learning rates, ranges of the initial weights randomly selected, error functions used and number of iterations for the convergence of error required on a Generalized Mean neural network model. Some inherent properties of this complex back propagation algorithm are also studied and discussed.

Keywords**—**Complex valued neural network, Generalized Mean neuron model, Signal processing.

I. Introduction

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. [2]

Signal Processing is one of the many fields of work that has been significantly influenced by neural network. We have sought to take a step further the works of T.Nitta in the field of complex back propagation and of P. K. Kalra, J. John and R. N. Yadav in Generalized Mean Neural Model. The aim of our paper is to demonstrate the effectiveness of a complex valued Generalized Mean Neural Model in signal processing.
Our experiments demonstrated an increased efficiency in training phase. A complex valued neural network improves the models efficiency 1.5 times and a complex valued generalized mean neuron model improves the efficiency 1.67 times over a real valued neuron network.

IBM India Private Limited

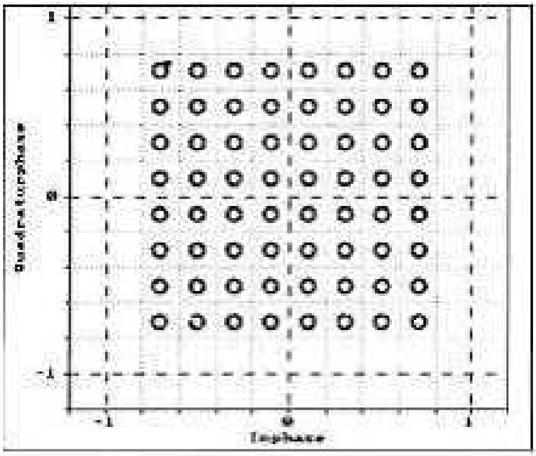

Fig. 1. Constellation diagram [9]

II. Preliminaries

A. Constellation Diagram in Signal processing

Quadrature Amplitude Modulation (QAM) uses many different phases known as states: 16, 32, 64, and 256. Each state is defined by a specific amplitude and phase. This means the generation and detection of symbols is more complex than a simple phase or amplitude device. Each time the number of states per symbol is increased the total data and bandwidth increases. Constellation diagrams are used to graphically represent the quality and distortion of a digital signal. In practice, there is always a combination of modulation errors that may be difficult to separate and identify, as such, it is recommended to evaluate the measured constellation diagrams using mathematical and statistically methods.

In the constellation diagram, an interferer shows in the form of a rotating pointer superimposed on each signal status. The example applies the condition that there is no other error present at the same time. The constellation diagram shows the path of the pointer as a circle around each ideal signal status.

Carrier suppression or leakage is a special type of interference in which its frequency equals the carrier frequency in the RF channel. Carrier leakage can be superimposed on the QAM signal in the I/Q modulator. In the constellation diagram, carrier leakage shows up as a shifting of the signal states corresponding to the DC components of the In-phase and Quadrature components. Additive Gaussian noise can disturb

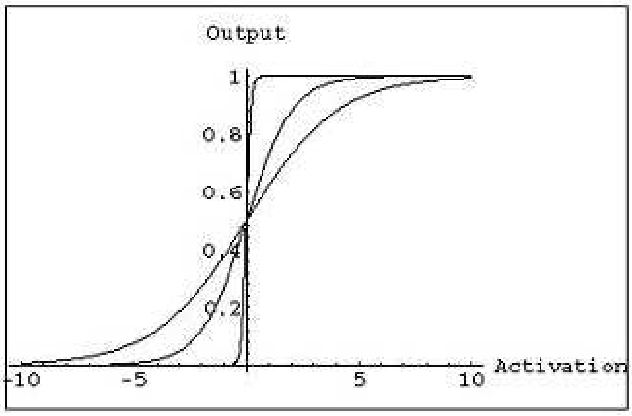

Fig. 2. Input/Output graph of a unit

the digitally modulated signal during analog transmission, for instance in the analog channel. Additive superimposed noise normally has a constant power density and a Gaussian amplitude distribution throughout the bandwidth of a channel. If no other error is present at the same time, the points representing the ideal signal status are expanded to form circular ”clouds” [11].

B. Back Propagation Algorithm In A Real Plane

Back propagation algorithm has been used extensively in neuron models. This algorithm is a development from the simple Delta rule in which extra hidden layers (layers additional to the input and output layers, not connected externally) are added. The network topology is constrained to be feed forward or loop-free - generally connections are allowed from the input layer to the first (and possibly only) hidden layer; from the first hidden layer to the second and from the last hidden layer to the output layer. In a typical back propagation network, the hidden layer learns to recode (or to provide a representation for) the inputs. More than one hidden layer can be used. The architecture is more powerful than single-layer networks: it can be shown that any mapping can be learned, given two hidden layers (of units). The units are a little more complex than those in the original perceptron.
Their input/output graph is shown in Figure 2. As a function:

$$Y = 1/(1 + e^{(-k(\sum Win*Xin))}) \ldots\ldots (1)$$

The graph shows the output for k = 0.5, 1, and 10, as the activation varies from -10 to 10.The weight change rule is a development of the perceptron learning rule. Weights are changed by an amount proportional to the error at that unit times the output of the unit feeding into the weight. Running the network consists of the forward pass and backward pass. In the forward pass the outputs are calculated and the error at the output units calculated. In the backward pass the output unit error is used to alter weights on the output units. The error at the hidden nodes is calculated (by back-propagating the error at the output units through the weights), and the weights on the hidden nodes are altered using these values. For each data pair to be learned a forward pass and backwards pass is performed. This is repeated over and over again until the error is at a low enough level. [7]

C. Generalized Mean Neuron Model

This new neuron model uses a new aggregation function. The origin of this new aggregation function is the generalized mean-operator of fuzzy sets, given by Piegat. [1] In GMN model the aggregation function is generalized mean of all the inputs of neuron. The mathematical representation of this model is given as

$$net = (\sum(w_{in}(x_n)^r/N)^{1/r} \ldots\ldots (2)$$

Where net represent the total input activity. This operator is known as Generalized Mean Operator and r is known as generalization parameter. For different values of r it represents some useful mathematical operators

$r \to \infty$ $net = \max(x_n)$

$r = 1$ $net = (\sum((x_n)/N)$ or the arithmetic mean

$r \to 0$ $net = (\Pi((x_n)/N)$ or geometric mean

$r = -1$ $net = (\sum((x_n)/N)^{-1}$ or harmonic mean

$r \to \infty$ $net = \min(x_n) \ldots\ldots (3)$

Generalized mean neuron model adapts various orders, i.e., for r=1 it represents first order neuron and for r=0 it represents Nth order neuron. For our model we used the first order for our generalized mean neuron model with complex inputs and outputs [8].

III. Complex Valued Neural Network

A complex valued neural network is a neural network (of arbitrary topology) which consists of complex valued input and/or weights and/or thresholds and/or activation functions. The need for such neural networks is widespread For instance, in electrical engineering, signals are complex valued. The processing of such signals requires the design and implementation of new complex valued neural network architectures. This subject has been gaining increasing interest and significance in recent years. One of the most important characteristics of the Complex-valued neural networks is the proper treatment of complex-amplitude information, e.g., the treatment of wave-related / rotation-related phenomena such as electromagnetism, light waves, quantum waves, oscillatory phenomena even including traffic signal control, and color images processing based on adaptive signal rotation.

A. The Complex BP Algorithm

This algorithm is a complex valued version of a probabilistic-descent method. It has been proved that the learning algorithm for the complex APCM converges. This algorithm states the following. If n is a parameter representing discrete time then we can modify the complex valued parameter w as

$w_{n+1} = w_n + \Delta w_n,$

where wn denotes a complex valued parameter at time n. We can also re-write the equation as follows:

$$\Re[w_{n+1}] = \Re[w_n] + \Re[\Delta w_n],$$

$$\Im[w_{n+1}] = \Im[w_n] + \Im[\Delta w_n] \ldots\ldots (4),$$

where $\Re[z]$, $\Im[z]$ denotes the real and imaginary part of a complex number z, respectively. By definition we say that aparameter w is optimal if and only if the average error R(w) is local or global minimum. [3], [4] Then the following theorem holds.

Let A be a positive definite matrix. Then by the using the update rules :

$$\Re[\Delta w_n] = -\varepsilon A \Delta^{\Re} r(z(w_n, x_n), y_n),$$

$$\Im[\Delta w_n] = -\varepsilon A \Delta^{\Im} r(z(w_n, x_n), y_n), n = 0, 1 \ldots$$

The (complex-valued) parameter w approaches the optimum as near as desired by choosing a sufficiently small learning constant $\varepsilon > 0$ ($\Delta^{\Re}$ is a gradient operator with respect to the real part of w, and $\Delta^{\Im}$ with respect to the imaginary part).[5]

B. Generalization of Real Back Propagation Algorithm

The theory of Complex APCM has been applied to a multi-layer (complex valued) neural network. For complex valued back propagation model we used the following derived results.

In a complex Back Propagation model all the input signals, weights, thresholds and output signals are complex numbers. The output activity (analogous to the activity of a real BP) for a neuron n is defined as:

$$Y_n = \sum W_{nm} X_m + V_n,$$

where W_{nm} is the (complex-valued) weight connecting neuron n and m, X_m is the (complex-valued) input signal from neuron m, and V_n is the (complex-valued) threshold value of neuron n. To obtain the (complex-valued) output signal, the activity Y_n is converted into its real and imaginary part

$$Y_n = x + iy = z$$

where i denotes $\sqrt{-1}$. Although various output functions can be considered, we used the output definition defined by

$$f_C(z) = f_R(x) + i f_R(y)$$

where $f_R(x)$ is a sigmoid function. Also $f_C(z)$ is not holomorphic.

Consider the following variables, w_{ml} is the weight between the input layer m and the hidden node l, v_{nm} is the weight between the hidden layer n and the output neuron m, Θ is the threshold of the m hidden neuron, γ is the threshold of the m output neuron. Let I_l , H_m , O_n denote the output values of the input neuron l, the hidden neuron m and the output neuron n, respectively. Let U_m and S_n denote the internal potentials of the hidden neuron m and the output neuron n, respectively. That is $U_m = \sum w_{ml}\ I_l + \Theta m$, $S_n = \sum v_{nm}\ H_m + \gamma n$, $H_m = f_C(Um)$ and $On = f_C(Sn)$.

Let $\delta_n = Tn - On$ denote the error between the actual pattern O_n and the target T_n of output neuron n. The square error for the pattern p is $E_p = (1/2) \sum (Tn - On)^2$ where N is the number of output neurons. The learning rule for the complex-BP model described above is as follows. For a sufficiently small learning constant (learning rate) $\varepsilon > 0$ and a unit matrix A, using Theorem stated above, it has been shown that the weights and the threshold should be modified according to the following equations:

$$\delta v_{nm} = \hat{H}_{m\delta\gamma_n}$$

$$\gamma_n = \varepsilon(\Re[\delta_n](1 - \Re[O_n])\Re[O_n] + i\Im[\delta_n](1 - \Im[O_n])\Im[O_n])$$

$$\omega_{mt} = \hat{I}_t \delta\theta_m \ldots\ldots (5)$$

$$\begin{aligned}\theta_m = \varepsilon[(1 - \Re[H_m])\Re[H_m] \textstyle\sum(\Re[d_n](1 - \Re[O_n])\Re[O_n])\\ \Re[O_n]\Re[v_{nm}] + \Im[d_n](1 - \Im[O_n])\Im[O_n])\Im[O_n]\Im[v_{nm}])\\ -i(1 - \Im[H_m])\Im[H_m] \textstyle\sum(\Re[d_n](1 - \Re[O_n])\Re[O_n])\Re[O_n]\\ \Im[v_{nm}] - \Im[d_n](1 - \Im[O_n])\Im[O_n])\Im[O_n]\Re[v_{nm}])]\end{aligned}$$

where $\hat{A}$ denotes the complex conjugate of A.[6]

IV. Experiments

A. C-XOR Benchmark Problem

Multi-layer networks use a variety of learning techniques. We used one of the popular techniques of back-propagation. We compared the output values with the correct answer to compute the value of predefined error-function. The error was then fed back through the network. Using this information, the algorithm adjusted the weights of each connection and thus reduced the value of the error by some small amount in successive iterations. After repeating this process for a sufficiently large number of training cycles the network converged to a stable state where the error of the calculations is insignificant. In this way the network has learns a certain target function. To adjust weights properly we applied a general method for non-linear optimization that is called gradient descent. For this, the derivative of the error function with respect to the network weights is calculated and the weights are then changed such that the error decreases (thus going downhill on the surface of the error function). This is also the reason why back-propagation can only be applied on networks with differentiable activation functions.

To check the efficiency of Generalized mean neuron model in a complex plane, we first observed the effects of the model on the C-XOR benchmark mode. To enhance our experiments results we varied the learning rates, the initial weights taken, and the iterations of the neural network. We also studied the effect of various error functions on the complex valued generalized mean neuron model.

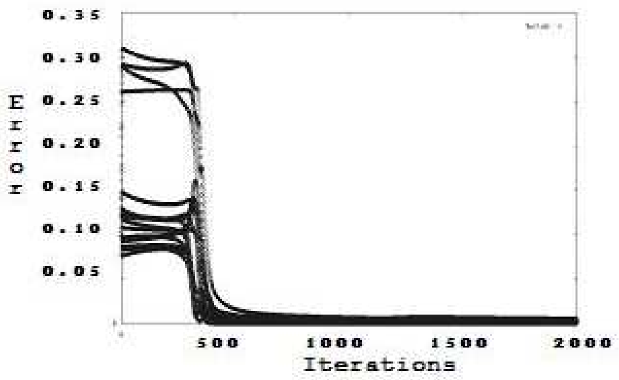

Fig. 3. Error convergence for Complex Quadratic error function

1) Model Description: Our first basic model had two complex valued inputs which corresponded to the inputs of the CXOR benchmark problem. The model had one hidden layers. The layer had four nodes in the first hidden layer. All the nodes had generalized mean function. All the outputs were finally sent to the output layer which also had another generalized mean function.

2) Input: We used f(x) = (1/ (1+exp (-x))) as our activation function. The value of the learning rate was varied from 0.001 to 0.999. The initial weights were chosen randomly by rand( ) function. This was also varied to get the optimal results. We experimented with various error functions on our model so see which error function converged the error in minimum number of iterations. All the observations were extensively researched and documented.

3) Results and Discussion: Various important results were deduced from the above experiments. They have been summarized as below:
To compare the effectiveness of our model we first experimented with a real valued neural model. The results obtained have been documented below:

TABLE I
REAL VALUED NEURAL MODEL

Iterations	Error
1	0.56
250	0.32
1000	0.032
2000	0.0021
10000	0.0001

The presences of a Generalized mean neuron function improves the efficiency of a neural model significantly in terms of number of iterations.

TABLE II
WITH A GENERALIZED MEAN NEURON FUNCTION

Iterations	Error
1	0.32
250	0.23
1000	0.001

TABLE III
COMPLEX QUADRATIC ERROR FUNCTION

Iterations	Error
1	0.117
250	0.06
1500	0.001

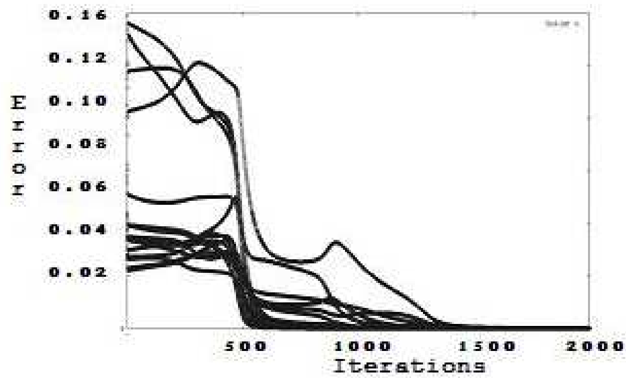

Fig. 4. Error convergence for Complex fourth error function

We varied the learning rates from 0.001 and 0.999 and observed the effect of the learning rates on the error convergence. For Complex Quadratic, Complex fourth, Complex Huber, complex Logcosh, Complex Welch, Complex Minkowski, Complex Fair error function the best error convergence rate was observed when the learning rate was 0.05 - 0.055 . For mean-median error function the best convergence rate was observed when the learning rate was 0.045 - 0.05. The best results were observed when the value of the learning rate was kept low.

B. Signal Processing

Phase Jitter or phase noise in the QAM signal is caused by transponders in the transmission path or by the I/Q modulator. It may also be produced in carrier recovery. In contrast to the phase error described above, phase jitter is a statistical quantity that affects the In-phase and Quadrature path equally. In the

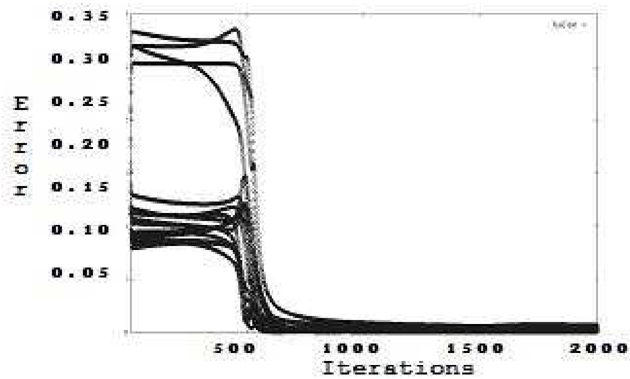

Fig. 5. Error convergence for Complex mean-median error function

TABLE IV
COMPLEX FOURTH ERROR FUNCTION

Iterations	Error
1	0.35
250	0.15
1200	0.001

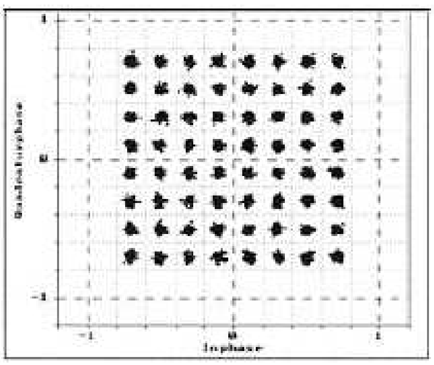

Fig. 6. Constellation Diagram with noise

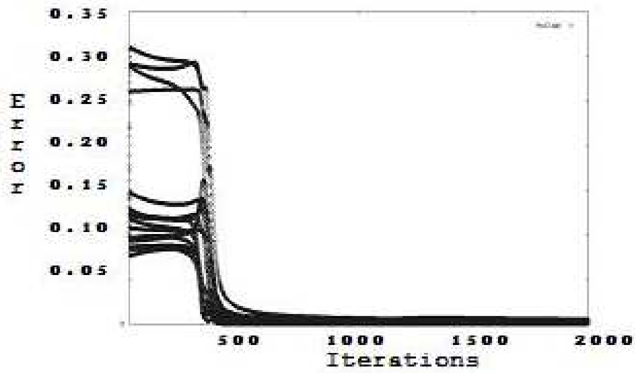

Fig. 7. Error convergence with Complex Quadratic

constellation diagram, phase jitter shows up by the signal states being shifted about their coordinate origin.

1) Model Description: The input had 3 input nodes. The first hidden layer had four generalized mean neural function units as depicted earlier. The output from these functions was then processed further using another generalized mean neural function. The output from this node was observed and studied as with intrinsic details.

2) Result and Discussion: We worked on 4QAM and 16QAM data. Our aim was to reduce noise drastically in minimum number of iterations. The presence of a generalized mean neural function system greatly enhanced the error convergence rate. The model achieved an error of .000001 within 12,000 iterations. This also depended on the value of learning rate, weights, the error function taken, the number of inputs taken and the value of generalization parameter. We tested our model on a number of error functions also. The error function that gave us the best result was the complex quadratic error function.

TABLE V
ERROR CONVERGENCE WITH COMPLEX QUADRATIC

Iterations	Error
1	0.32
250	0.2
1000	0.05
2000	0.0009
12000	0.000001

V. CONCLUSION AND FUTURE WORK

We have shown the effectiveness of a complex valued version of back propagation model with generalized mean neuron model in signal processing Furturemore, we have investigated the fundamental characteristics of the complex- Back Propagation algorithm and found this algorithm better than general or real valued BP algorithm. The average convergence speed is much superior to that of a Real Back Propagation. The updating rule of the Complex-Back Propagation is such that the probability for a standstill in learning is reduced. Also the presence of an GMN unit in a complex neural model is well suited for signal processing and studying constellation diagrams.

Indeed several interesting applications of the complex valued neural network architectures can be extended in the following areas: Optoelectronics, Imaging, Optical computing, Remote sensing, Quantum Neural devices and systems, Intelligent transport systems, Spatiotemporal analysis of Physiological Neural Systems, Artificial Neural Information Processing, Communication system design (Mobile channel equalizer design), Direction of Arrival Estimation (Signal Processing), Traffic Control, Robotics, Neuron Dynamics, Chaos in the complex domain.

VI. ACKNOWLEGMENT

A grateful thanks to Mr Bipin Kumar Tripathi for providing the oppurtunity to work in the field of neural network. Had it not been for his immense support this paper would have not seen the light of the day.

REFERENCES

[1] A. Piegat, Fuzzy Modeling and Control, Physica-Verlag, Heidelberg, New York, 2001.
[2] M.H.Housan, Fundamental of Artificial NEURAL network, Princeton Hall Of India, 1998.
[3] Nitta T., "An Analysis of the Fundamental Structure of Complex-valued Neurons", Neural Processing Letters, Vol.12, No.3, pp.239-246
[4] Nitta T., "Redundancy of the Parameters of the Complex-valued Neural Networks", Neurocomputing, Vol.49, Issue 1-4, pp.423-428 (2002).
[5] Nitta T., "Solving the XOR Problem and the Detection of Symmetry Using a Single Complex-valued Neuron", Neural Networks, Vol.16, No.8, pp.1101-1105 (2003)
[6] Nitta T., "An Extension of the Back-Propagation Algorithm to Complex Numbers", Neural Networks, Vol.10, No.8, pp.1391-1415 (1997)
[7] P. J. Werbos, Back propagation through time: what it does and how to do it Proc. Of the IEEE 78(10):1550-1560, 1990.
[8] R.N. Yadav, P.K. Kalra and J. John, Neural network learning with generalized mean based neuron model,, Springer Verlag
[9] http://www.blondertongue.com/QAM-Transmodulator/QAM_defined.php
[10] http://www.psych.utoronto.ca/users/reingold/courses/ai/cache/InvSlides.html
[11] http://www.lycos.com/info/quadrature-amplitude-modulation.html

Anupama Pande Ms Anupama Pande completed her undergraduation with a computer science major from HBTI, Kanpur, India. She currently works as a software engineer for the IBM India Software Labs. Ms Pande is the author of various papers published in international conferences and journals as ICAS'07, and IJCSE'07.

Ashok K Thakur Mr Ashok K Thakur completed his undergraduation with a computer science major from IIT Kanpur, India. He currently works as a software engineer for the IBM India Software Labs. Mr Thakur is the author of various papers published in international conferences and journals as ICAS'07, BIBE 07, and IJCBDD.

Swapnoneel Roy Mr Swapnoneel Roy completed his graduation with a computer science major from IIT Madras, India. He currently works as a software engineer for the IBM India Software Labs. Mr Roy is the author of various papers published in international conferences and journals as ICAS'07, BIBE 07, and IJCBDD.