



Exploration and Analysis of Computational Learning Algorithms on the Second Generation Neural Network

Amit Gupta^{1*} Bipin Kumar Tripathi² Vivek Srivastava³

¹*Department of Computer Science and Engineering,
Dr. A.P.J. Abdul Kalam Technical University, Lucknow, India.*

²*Department of Computer Science and Engineering,
Harcourt Butler Technical University, Kanpur, India*

³*Department of Computer Science and Engineering, Rama University, Kanpur, India*

* Corresponding author's Email: amitphd40@gmail.com

Abstract: Computation intelligence is an interesting field having ability to solve many complex problems that are exist in real world. Suitable collaboration of different type of computational learning intelligence techniques viz. fuzzy, evolutionary and neural methods can be more efficient for solving real word complex problems. This paper presents comparative analysis using some intelligence approaches viz. fuzzy c-means clustering (FCM), evolutionary fuzzy clustering with Minkowski distance (EFC-MD) and fusion of EFC-MD with functional modular neural network (EFCMD-FMNN) for determine their efficiencies in real domain (first generation neural network) and complex domain (second generation neural network) both. EFC-MD is introduced for pre-classification, specifying the optimal number of cluster that are assigned in training process. Rather than the Euclidean distance, Minkowski provides the flexibility to algorithm clustering in achieving any size for the cluster keeping the distance matrix in mind. A new approach, hilbert transformation is used for converting real datasets in complex datasets (complex number form) for the computation in complex domain. In this paper initially two existing real domain datasets (wine datasets and monk datasets) is selected and complexities these datasets in complex domain using hilbert approach. Next application of various computational algorithm separately in respective domain to evaluating performance in terms of accuracy. Exploration of these algorithms in complex domain (2-D) investigates improved results and better learning characteristics in the compression of real domain (1-D).

Keywords: Complex activation function, Fuzzy clustering, Hilbert transformation, Complex back propagation, Evolutionary strategy, FMNN (functional modular neural network).

1. Introduction

The conventional methodologies and approaches are used in earliest neural network like real valued neural network but these methodologies are ineffective with the extension of these ideas in the complex domain or high dimensional neural computing. The idea of high dimension neural computing starts with the devolvement of complex value neural network can be traced by N.Aizenberg in 1971[1]. Complex form of back propagation is proposed as complex least mean square (LMS) algorithm by Widrow, MacCool and Ball (1975) [2].

Later N.Aizenberg study the idea of neurons with phase dependent functions. In next the idea comes of multivalued neuron and universal binary neurons that is presented by I. Aizenberg [3-4, 14]. In early 1990s back propagation is presented in one dimensional (real) and two-dimensional (complex) plane. CVNN second generation neural network that is widely used in the field of such as ocean technology, radar imaging sonar waves, are intelligently published. Kim and guest (1990) [5] presented a complex valued learning algorithm in the field of signal processing. Lueng and haykin (1991) [6] was published a extended form of CBP in the idea of complex activation function that is extension of sigmoid

function. The dynamic nature of CVNN was presented by Hirose (1992). T Nitta (1997) [7] published an extensive idea of CBP along with complexity analysis. He also reported certain problems where conventional BP fails but CBP manage to solve. Later tripathi (2009) studied of complex activation function that is bounded in nature called split type activation function. Meanwhile, T. Adali (2003) [9] reported the issue related to unboundedness of function with fully complex valued network. Benvenuto and Piazza (1992) [8] published a different version of CBP with the extension of real activation function in complex domain. From 2009 to 2012, Tripathi proposed extensive study in the field of novel complex valued neural computing [10]. A. Hirose and S. Yoshida. Presented a detailed generalization characteristics of complex-valued feed forward neural networks in 2012 [11]. T. Nitta (2013) extended his research in the field of complex valued neural network supporting hierarchical structure of complex valued neural network and focusing on local minima theory [12]. Next Hirose (2013) presented a comprehensive coverage of conventional complex valued neural network [13].

All these works in the field of complex domain or high dimensional neural computing for analysing wide applicability for the performance evaluation. Various researchers presented and compiled these idea in the field of number theory as real number and extended the concept in two dimension as complex number. The real valued neural network is based on the real number (one dimensional) and complex value neural network (two dimension).

Next purposed research focus on the field of computation intelligence in complex domain and real domain for analysing performance in terms of accuracy.

Bhattacharjee [15] has proven the superiority of computational intelligence techniques over conventional statistical methods. In order to solve real domain problems in complex domain, we introduced a combination of two algorithm viz. evolutionary fuzzy clustering with functional modular neural networks. This combination is worked as a novel hybridization approach. In this approach functional modular neural network proven their improved conditions over conventional neural network in terms of decision making and input patterns processing and used as a classifier. Next Hybridisation of neural network with fuzzy clustering is introduced by Ozbey [16]

On the other hand, Mingoti and Lima [17] has proposed a proof the superiority of fuzzy c means clustering over k-mean clustering in real domain. The

big disadvantages of FCM is that it creates different partition of the same algorithm in different eposes Hrushkea in 2009 [18]. To overcome this drawback, evolutionary algorithm generates the best solution for partitioning when combine with fuzzy clustering. T seng and yang in 2001 assessed the optimal number of clusters that are produces in different runs of fuzzy clustering that are deal with fitness function and some validity criteria [19].

Relationship between different types of distances and structures are figured in given datasets with the application of FCM clustering evolved by Yuan [20]. The proposed research strategy dealing with evolutionary fuzzy clustering with Minkowski distance (EFC-MD) improving the different classification conditions managing different shapes of clusters in their respective domains. Further, for training of the network functional modular neural network has been proposed in real and complex domain.

The output of each FMNN is similar to the number of neurons Members in each cluster. Maximum output for all FMNN holder forward, one final decision-making unit, where the output produced by FMNN is compared with the threshold provided in the integrator

The whole organisation of this proposed research is based on the performance analysis in their respective domains taking variety of computational learning algorithms viz. real domain (FCM, EFC-MD, EFCMD-FMNN) and complex domain (CFCM, CEFC-MD, CEFCMD-FMNN) is computed over many bench mark problem datasets that are available from different repositories in real form.

2. Second generation neural network

Second generation neural network is an extension of first generation neural network. The first-generation neural network presented a single dimension analysis in real domain. Here we take input as a real number and training the network with the help of real back propagation algorithm (RBP) but second generation neural network take the input data as a complex number as $y=f(u,v)$ including both amplitude and phase. In this proposed research for training of the complex network a complex back propagation algorithm is applied to a multilayer (complex-valued) neural network. In second generation, neural network all the signal related to input, weight, threshold and output are in the form of complex number.

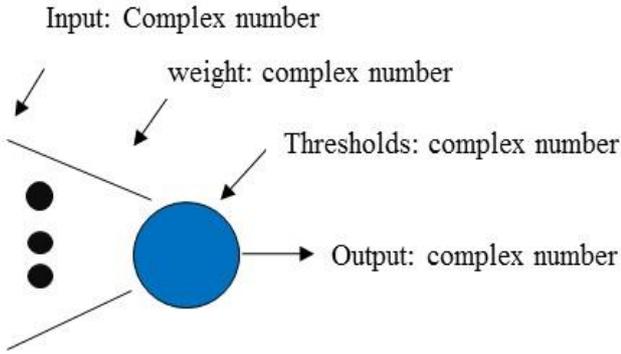


Figure. 1 Second generation neural network

The output Signal O complex to in the second-generation neural network is defined as:

$$O_{complex} = \sum I_{complex} W_{complex} + V_{complex} \quad (1)$$

where w is the weight, I is the input signal and V is the threshold value of a neuron. The output signal $O_{complex}$ is a combination of real and imaginary component exist in the form com complex number.

3. Back propagation in complex domain

In this section we shall present a back propagation algorithm in complex domain. Here we consider three-layer network as shown in Fig. 2. Let $X_e = (x_{e1}, x_{el}, \dots, x_{eQ})$ be input vector is applied to the input layer of network. Detailed explanation of the algorithm given as following steps and some other notations that is used in this algorithm as follows:

- h : hidden unit, O : output unit, w : weight factor
- R : real part, I : imaginary part, δ : error term
- F_n : Complex activation function (general form: $F(x) = f(x_R) + jf(x_I)$), f : real activation function, tot : Net input values to the hidden layer and each output layer. R and I represent the real and imaginary parts.
- θ : Bias term.

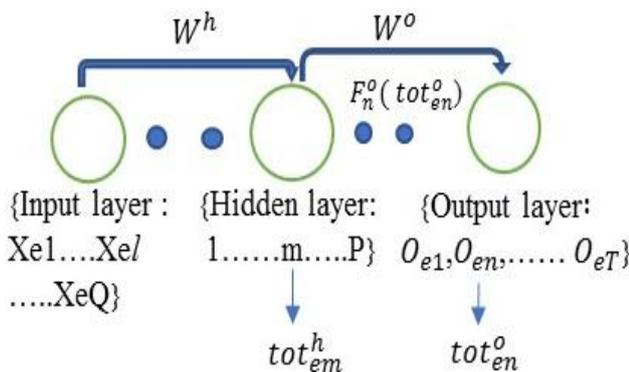


Figure. 2 Complex back propagation learning

The error term for output units

$$\delta_{en}^o = f_n^o(tot_{en,R}^o)Re(D_{en} - O_{en}) + jf_n^o(tot_{en,I}^o)Im(D_{en} - O_{en}) \quad (2)$$

where D_{en} is the desired output value, and O_{en} is the actual output from the k^{th} unit. The subscript “ e ” refers to the e^{th} training vector, and “ n ” refers to the n^{th} output unit.

the error term for hidden units

$$\delta_{em}^h = f_j^h(tot_{em,R}^h)Re(\sum_{n=1}^T \delta_{en}^o w_{nm}^{o*}) + jf_m^h(tot_{em,I}^h)im(\sum_{n=1}^T \delta_{en}^o w_{nm}^{o*}) \quad (3)$$

Update weight on the output layer according to

$$w_{nm}^o(t+1) = w_{nm}^o(t) \eta \delta_{en}^o i_{em}^* \quad (4)$$

$$w_{nm,R}^o(t+1) = w_{nm,R}^o(t) + \eta < \delta_{en,R}^o i_{em,R} - \delta_{en,I}^o i_{em,I} > , \quad (5)$$

$$w_{nm,I}^o(t+1) = w_{nm,I}^o(t) + \eta < \delta_{en,I}^o i_{em,R} - \delta_{en,R}^o i_{em,I} > \quad (6)$$

Update weight on the hidden layer according to

$$w_{ml}^h(t+1) = w_{ml}^h(t) \eta \delta_{em}^h x_{el}^* \quad (7)$$

Or

$$w_{ml,R}^h(t+1) = w_{ml,R}^h(t) + \eta < \delta_{em,R}^h x_{el,R} - \delta_{em,I}^h x_{el,I} > , \quad (8)$$

$$w_{ml,I}^h(t+1) = w_{ml,I}^h(t) + \eta < \delta_{em,I}^h x_{el,R} - \delta_{em,R}^h x_{el,I} > \quad (9)$$

Calculate the tot (net/total input values) on hidden layer and output layer:

At hidden layer:

$$tot_{em}^h = tot_{em,R}^h + jtot_{em,I}^h = \sum_{l=1}^Q w_{ml}^h x_{el} + \theta_m^h \quad (10)$$

At output layer:

$$tot_{en}^o = tot_{en,R}^o + jtot_{en,I}^o = \sum_{m=1}^p w_{nm}^h x_{em} + \theta_n^h \quad (11)$$

4. Learning with back propagation algorithm in real domain

Back propagation error algorithm or back propagation algorithm (BP) is an optimize method for training of artificial neural network. Networks without hidden units are very limited in the input – output mappings they can model. Adding a Layer of hand-coded features (as in a perceptron) makes them much more powerful but the hard bit is designing the features. Propagation and weight update are the two-phase cycles in which the back-propagation algorithm repeats. The following mathematics steps are given.

x_j^l : Input to note j of layer l

w_{ij} : Weight from layer $(l - 1)$ node to l to layer l node j

$\sigma(x) = \frac{1}{1+e^{-x}}$: Sigmoid Transfer Function

θ_j^l : Bias of node j of layer l

O_j^l : output of note j in layer l

t_j : Target value of note j of output layer

The error calculation between training set of data point t_j and output layer output O_j^l

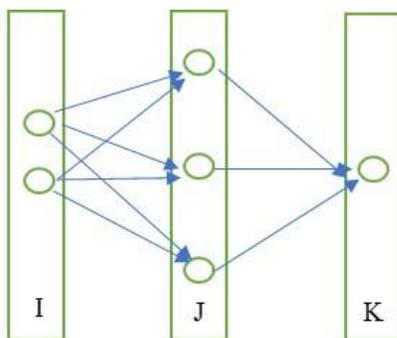
We can write the error as

$$E = \frac{1}{2} \sum_{k \in K} (O_j^l - t_j)^2 \tag{12}$$

We let the error of the network for a single training iteration be denoted by E . we want to calculate $\frac{\partial E}{\partial w_{jk}^l}$, the rate of change of the error with respect to the given connective weight, so we can minimize it. Now we consider two cases: the node is an output node, or it is in a hidden layer.

Output layer node:

$$\frac{\partial E}{\partial w_{jk}^l} = \frac{\partial}{\partial w_{jk}^l} \frac{1}{2} \sum_{k \in K} (O_j - t_j)^2 \tag{13}$$



Figurer. 3 A neural network

or

$$\frac{\partial E}{\partial w_{jk}^l} = (O_k - t_k) O_k (1 - O_k) O_j \tag{14}$$

For notation purpose in above equation consider δ_k to be the expression $(O_k - t_k) O_k (1 - O_k)$ So we can write the above equation as

$$\frac{\partial E}{\partial w_{jk}^l} = O_j \delta_k \tag{15}$$

where $\delta_k = (O_k - t_k) O_k (1 - O_k)$ Hidden layer node

$$\frac{\partial E}{\partial w_{ij}^l} = O_j (1 - O_j) O_i \sum_{k \in K} (O_k - t_k) O_k (1 - O_k) w_{jk} \tag{16}$$

But recalling our definition of δ_k we can write this as

$$\frac{\partial E}{\partial w_{ij}^l} = O_j (1 - O_j) O_i \sum_{k \in K} \delta_k w_{jk} \tag{17}$$

Similar to before we will now define all terms besides O_i to be δ_j . So, we have

$$\frac{\partial E}{\partial w_{ij}^l} = O_i \delta_j \tag{18}$$

What about the bias? If we incorporate the bias term θ into the equation you find that

$$\frac{\partial O}{\partial \theta} = 1 \tag{19}$$

Therefore, we view the bias term as output from a node which is always one. This hold for any layer l we are consider with, a substitution into the previous equations gives us that $\frac{\partial E}{\partial \theta} = \delta_l$ (because the O^l IS replacing output from the previous layer)

Update the weight and biases as follows:

Given $\Delta W = -\eta \delta_l O_{l-1}$, η is a small number.

$$\Delta \theta = \eta \delta_l \tag{20}$$

Apply

$$W + \Delta W \rightarrow W \tag{21}$$

$$\theta + \Delta \theta \rightarrow \theta \tag{22}$$

5. FCM clustering

In proposed research a new clustering approach presented in complex domain: complex fuzzy c-

means (CFCM) algorithm. But in case of fuzzy c-means clustering deals this algorithm in real domain (real numbers). The CFCM uses a complex vector (complex center) to generalize this type of algorithm. In 1981 Bezdek [21] described this algorithm. Let U be the $C \times N$ membership matrices where $\mu_{ik} \in [0,1]$, $1 \leq i \leq c$, $1 \leq k \leq n$, is the degree of membership of data x_k to the cluster with prototype v_i .

$$U_{c \times n} = \begin{bmatrix} \mu_{11} & \mu_{12} & \dots & \mu_{1n} \\ \mu_{21} & \mu_{22} & \dots & \mu_{2n} \\ \dots & \dots & \dots & \dots \\ \mu_{c1} & \mu_{c2} & \dots & \mu_{cn} \end{bmatrix} \quad (23)$$

Dunn further studied fuzzy objective function:

$$J_D(U, V) = \sum_{i=1}^c \sum_{j=1}^n \mu_{ij}^2 \|x_j - v_i\|^2 \quad (24)$$

where v_i is cluster center of i set, Then Bezdek (1981) extended it to:

$$J_m(U, V; X) = \sum_{i=1}^c \sum_{j=1}^n \mu_{ij}^m \|x_j - v_i\|^2, \quad 1 \leq m \leq \infty \quad (25)$$

Using Lagrangian objective function and getting the following minimum (U, V)

$$\min \{ J_m(U, V, \lambda; X) = \sum_{i=1}^c \sum_{j=1}^n \mu_{ij}^m \|x_j - v_i\|^2 + \sum_{k=1}^n \lambda_k \sum_{i=1}^c (\mu_{ik} - 1) \} \quad (26)$$

by differentiating $\left(\frac{dJ_m}{dv_i} = 0, \frac{dJ_m}{d\mu_{ij}} = 0, \frac{dJ_m}{d\lambda_k} = 0 \right)$, by solving we get:

$$v_i = \frac{\sum_{j=1}^n (\mu_{ij})^m x_j}{\sum_{j=1}^n (\mu_{ij})^m}, \quad 1 \leq i \leq c \quad (27)$$

and

$$\mu_{ij} = \left(\sum_{k=1}^c (\|x_j - v_i\|^2 / \|x_j - v_k\|^2)^{\frac{1}{m-1}} \right)^{-1}, \quad 1 \leq i \leq c, 1 \leq j \leq n \quad (28)$$

the prototype (centroid) v_i is a real center that is characterized by each cluster i and the fuzzy membership vector μ_{ij} .

5.1 Complex centers

The previous mathematical formulas defines that the v_i real center) is obtained by minimizing an objective function J_m . It should be noted that the

value of J_D at v_i is not zero but the minimum positive possible one. For complex centers (z_i), the value of J_m at z_i is always zero. The complex centers are derived based on following concept. The complex center can be written as:

$$z_i = v_i + j\sigma_i \quad (29)$$

where v_i is the real center and σ_i is the standard deviation.

Real distance versus complex center It should be noted that the real distance of a datum x_k to a complex center z_i is given by:

$$d(x_k, z_i) = \frac{1}{\sqrt{((x_k - v_i)^2 - \sigma_i^2)^2 + (2\sigma_i(x_k - v_i))^2}} \quad (30)$$

This can be derived using the following complex distance:

$$d_c(x_k, z_i) = (x_k - z_i)^2 = (x_k - v_i - j\sigma_i)^2 = (x_k - v_i)^2 - \sigma_i^2 + j(c(x_k - v_i)) \quad (31)$$

The relationship between the complex distance and the real distance is given by:

$$d(x_k, z_i) = \text{abs}(d_c(x_k, z_i)) \quad (32)$$

$$\text{where } \text{abs}(z) = \sqrt{(\text{real}(z))^2 + (\text{imag}(z))^2}$$

The derivation of the complex center for the FCM is similarly derived as follows:

$$j(v_i) = \sum_{k=1}^n \mu_{ki}^m (x_k - z_i)^2 = 0 \Rightarrow z_i^2 - \frac{2}{\sum_{k=1}^n \mu_{ki}^m} (\sum_{k=1}^n \mu_{ki}^m x_k) z_i + \frac{1}{\sum_{k=1}^n \mu_{ki}^m} \sum_{k=1}^n \mu_{ki}^m x_k^2 = 0 \quad (33)$$

The complex center is given by:

$$z_i = \frac{1}{\sum_{k=1}^n \mu_{ki}^m} \sum_{k=1}^n \mu_{ki}^m x_k + j \sqrt{\left(\frac{1}{\sum_{k=1}^n \mu_{ki}^m} \sum_{k=1}^n \mu_{ki}^m x_k \right)^2 - \frac{1}{\sum_{k=1}^n \mu_{ki}^m} \sum_{k=1}^n \mu_{ki}^m x_k^2} \quad (34)$$

where ($z_i = v_i + j\sigma_i$).

The derivation of the membership vector for the CFCM is derived as follows:

$$j(v_i) = \sum_{j=1}^n \mu_{ji}^m (x_j - z_i)^2 = \sum_{j=1}^n \mu_{ji}^m (d(x_j, z_i))^2 \tag{35}$$

Now setting $\frac{dJ}{d\mu_{ji}} = 0$ will give the following formula for the membership vector:

$$\mu_{ij} = \left(\sum_{k=1}^c \left\{ \frac{d(x_j, z_i)}{d(x_j, z_k)} \right\}^{\frac{2}{m-1}} \right)^{-1}, \quad 1 \leq i \leq c, 1 \leq j \leq n \tag{36}$$

6. Evolutionary fuzzy clustering with minkowski distances

To achieve optimal partition among different runs of FCM, the proposed rerserch employed evolutionary search strategy. Optimized partitioning refers to the well fitted data objects into the clusters. If the distance between objects and prototypes of clusters is minimum, then higher quality partition can be achieved. It provides more generalized nature of algorithm and algorithm perform well any type shapes of cluster. In this proposed research we called this technique as evolutionary fuzzy clustering with Murkowski distances (EFC-MD). This evolutionary fuzzy clustering with Murkowski distances (EFC-MD) is based on the real data sets problem or other word we can say real domain. For complex domain means complex data that have both phase and amplitude we use this algorithm as complex evolutionary fuzzy clustering with Murkowski distances (CEFC-MD).

The propose technique of this algorithm is defined in the following steps as the computation membership function, initialization of population, fitness function and best population for selection process and suitable validity criteria.

6.1 Computation of membership function in complex domain

Let $X = \{x_1, x_2, x_3, \dots, x_n\}$ is M objects each containing elements of different dimensions. EFC-MD clustering algorithm divides M data into fuzzy partition matrix U (size $C \times M$) having C clusters. The complex membership function in U is defined as μ_{ji} which satisfies the following equation.

$$\mu_{ij} = \left(\sum_{k=1}^c \left\{ \frac{d(x_j, z_i)}{d(x_j, z_k)} \right\}^{\frac{2}{m-1}} \right)^{-1}, \quad 1 \leq i \leq c, 1 \leq j \leq n \tag{37}$$

6.2 Initialization of population

At the beginning, the population (Fuzzy Partition Matrix U) is started randomly. Therefore, in each part different elementary population is produced. It is equivalent to Random Start of Fuzzy Partition Matrix of Traditional fuzzy c- means clustering.

6.3 Objective function computation

The objective function basically reduces the Minkowski distance between the object Centroids of the clusters. As much as this minimization occurs, the distance between the object and prototype of clusters decreases which yield more accurate partitioning. Let v_i represented the center of each cluster then objective function is computed as:

$$J = \sum_{i=1}^c \sum_{j=1}^n \mu_{ij}^m \|x_j - v_i\|^2, \quad 1 \leq m \leq \infty \tag{38}$$

Now the Minkowski distance is given by the following relation:

$$d(x_j, v_i) = \left(\sum_{k=1}^q \|x_{jk} - v_{ik}\|^p \right)^{\frac{1}{p}}, \quad 1 \leq p < \infty \tag{39}$$

The degree fuzziness of the given relation depends on the value of weighting exponent m. Basically, the value of m lies between one and infinity ($1 \leq m \leq \infty$) which impact the performance of CEFC-MD. Here p is mentioned as generalization parameter which generate the different shapes of cluster like boxes, ellipse. Sphere. The main purpose for selecting Minkowski distance is to provide the freedom to proposed algorithm for generating different shape of clusters that is not best suit in Euclidian distance measure. Centre of cluster and member function deals with CEFC-MD algorithm for minimizing the objective function with the applicability of some necessary conditions.

$$v_i = \frac{\sum_{j=1}^n (\mu_{ij})^m x_j}{\sum_{j=1}^n (\mu_{ij})^m}, \quad 1 \leq i \leq c \tag{40}$$

$$\mu_{ij} = \left(\sum_{k=1}^c \left\{ \frac{d(x_j, z_i)}{d(x_j, z_k)} \right\}^{\frac{2}{m-1}} \right)^{-1}, \quad 1 \leq i \leq c, 1 \leq j \leq n \tag{41}$$

Let be the value of objective function at t^{th} iteration is $J(\mu, v)^t$ the termination condition followed by the EFC-MD algorithm:

$$\|J(\mu, v)^{t+1} - J(\mu, v)^t\| < \xi \tag{42}$$

where ξ is threshold.

6.4 Selection

Let U^0, U^1, \dots, U^n , be the n populations generated by the n runs of proposed algorithm. As per our requirement when the value of objective function is minimum then higher the fitness function for selecting best U of propose algorithm, the final output of this proposed algorithm is optimized fuzzy partition matrix U .

6.5 Fitness function

The fitness function is holds to require the following condition:

Reasonable and consistent, less computation, well versatility, non-negative and maximum. The fitness function $y(\cdot)$ can be determine by the objective function $J(\mu, v)$.

$$y = \frac{1}{J(\mu, v) + \zeta} \quad (43)$$

where ζ is a constant, taking $\zeta = 1$ as per design requirement the fitness function is defined by following equation:

$$y = \frac{1}{J(\mu, v) + 1} \quad (44)$$

Fitness function is inversely proportion of objective function. Higher value of y gives the superior fittest population.

6.6 Best number of cluster selection with the help of validity criteria (validity index)

Primarily we choose a determinable number of cluster C . Apply the population initialization and computation of objective function which are given in section 6.4 to 6.5, are done recursively. Selecting different values of C Repetitively and measure the validly criteria by the following mathematical function:

$$\tau = \sum_{j=1}^l \Psi_{(ir)}(C_j) \eta - \Psi_{(ia)}(C_j) \quad (45)$$

Where $\Psi_{(ia)}(C_j)$ denotes the intra distance and $\Psi_{(ir)}(C_j)$ denotes inter distance of cluster C_j . Intra measure the distance within same cluster and inter different clusters. η Represents user defined quantity. The higher value of validity index (τ) predicts the best number cluster selection.

7. Hybrid model (CEFCMD-FMNN)

Tradition hybrid model (EFCMD-FMNN) is based on the real word clarification problem. But in complex domain the clusters are in the form of complex number (having phase and amplitude). we use combination of these two-computation algorithm in the form of (CEFCMD-FMNN).

8. Functional modular neural network

A modular neural network is a combination of several independent unit of neural networks. These independent parts are served as separate module to perform a part of complete task with the help of different sets of inputs. The main aim of modular neural network is to reduce a larger neural network into a smaller and more suitable for training the input patterns.

FMNN classifies the applied input set based on the fuzzy distribution of training patterns. The proposed function modular neural network has developed an efficient system with single hidden layer. With FMNN, this system requires neural networks which are not only compact but more efficient than a system using a neural network. Let x_i be the input and w_i be the corresponding weight then each Functional modular neural network (FMNN) is characterized by 5-tuples-

$$F = (q, N, C, \pi, y) \quad (46)$$

Here where q is the number of inputs, N is the number of classes, C is the number of functional modules, π is the cluster allocation matrix, the output of FMNN computed as:

$$y_{ij} = \delta \left(\sum_{k=1}^q x_k w_k \right)_{\pi_{ij}}, \quad \forall 1 \leq i \leq C, 1 \leq j \leq z \quad (47)$$

Here δ is an activation function and z is a membership grade element of a cluster. Let the number of functional module is as M_1, M_2, \dots, M_c . Then the output of corresponding FMNN M_i is

$$OP(M_i) = \max \left((y_{ij})_{\pi_{ij}} \right), 1 \leq j \leq z \quad (48)$$

Schematic representation Functional Modular Neural Network of is shown in Fig. 4

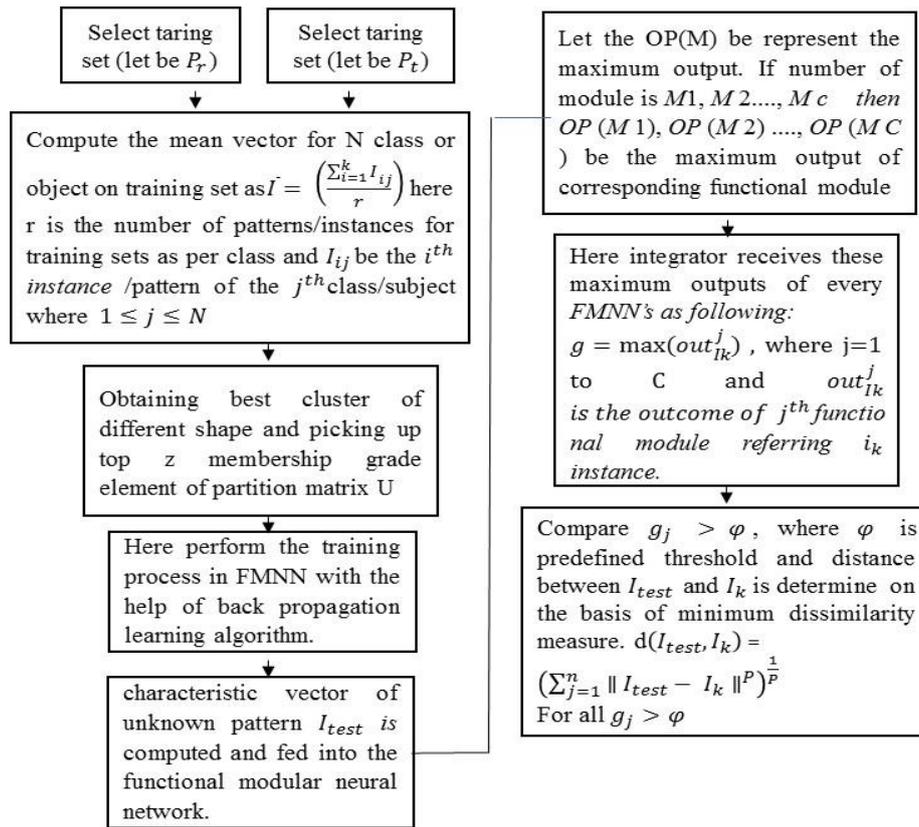


Figure. 5 EFCMD-FMNN algorithm

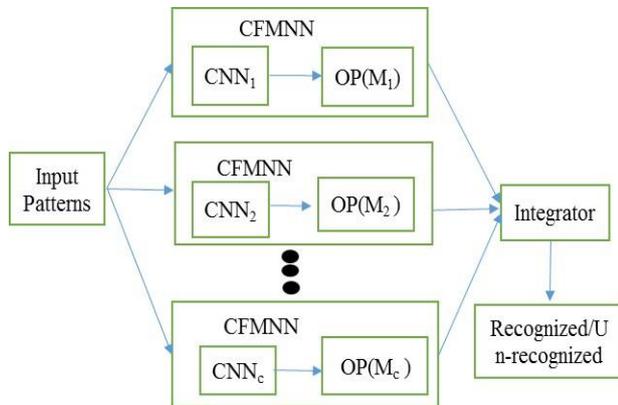


Figure. 4 Complex functional modular neural network

In the above description of given Fig. 4 represent as the complex neural network and the corresponding network used as complex modular neural network (CFMNN) in complex domain but in real domain we use simply neural network(NN) and corresponding network as a functional modular neural network (FMNN). Here the final output comes in the form of recognize and un recognize pattern of a given input pattern after a small code running on the output side termed as an integrator (decision unit) for recognition of input pattern.

9. Computational learning algorithm EFCMD-FMNN

To train the network and to recognize test patters, this section presents a procedural way to find computationally recognized output with the fusion of EFCMD and FMNN. The following systemic diagram describes the different steps of algorithm in Fig. 5.

10. Hilbert transformation

If working data sets are in the form of real number, then using Hilbert transform [22-25] convert the whole data in the form of complex numbers with same amplitude and phase sifted by $\pm 90^\circ$. Now in our proposed research we used Hilbert transform for analysis in complex domain of different computation learning algorithm. Hilbert transform a signal $B(t)$ is defined as the transform in which phase angle of all the components of the signal is sifted by $\pm \frac{\pi}{2}$.

Hilbert transform of signal $B(k)$ is represented with $\widehat{B}(t)$ is given by

$$\widehat{B}(t) = \frac{1}{\pi} \int_{-\infty}^{+\infty} \frac{B(k)}{t-k} dk \tag{49}$$

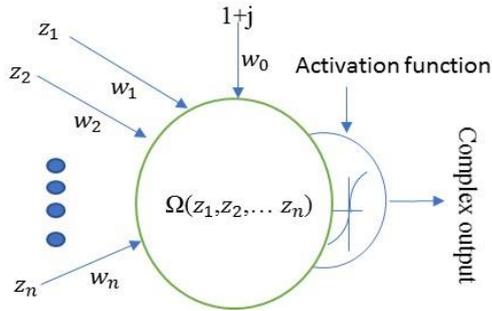


Figure. 6 Complex neural network with complex activation function

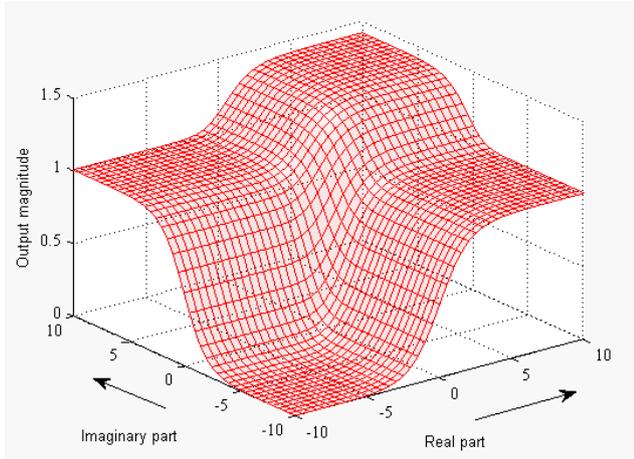


Figure. 7 Output magnitude of the A-P type activation function

$B(t)$ and $\widehat{B}(t)$ are orthogonal and if furrier transform exists then Hilbert transform is also existing for energy and power signals.

Syntax

$$d_r = Hilbert(d_r)$$

$$d_r = Hilbert(d_r, n)$$

11. Activation function in complex domain

For single dimensions the neural network uses the real domain activation functions. But in the case of two dimension or higher degree amplitude and phase type activation Function is used. To ensuring the analytic nature of a function is boundedness is required in its preliminaries for convergence analysis. Benevento and piazza [26] has given an idea of an alternative complex activation function to overcome the conflict between the requirements of an activation function. Liouville’s theorem explains that an analytical function cannot be bounded on all the complex plane unless it is constant.

Let be $f_c(z)$ be the complex function containing a separate real and imaginary part of weight inputs (A-P type (amplitude and phase type activation function))

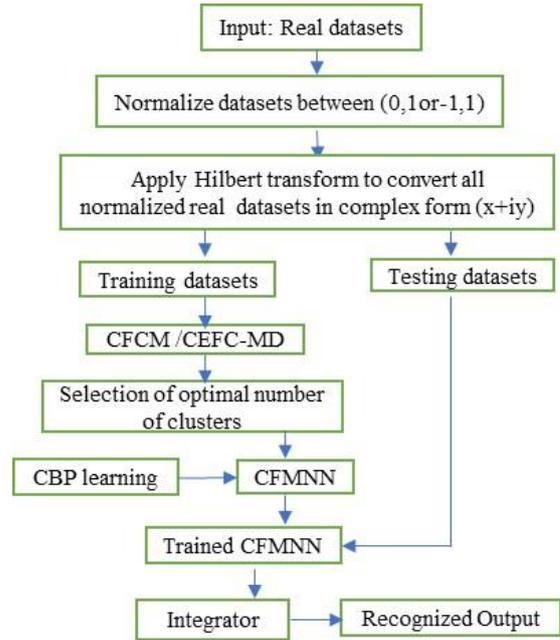


Figure. 8 Methodology

$$f_c(z) = f(x) + jf(y) \tag{50}$$

where $z = x + jy$ is a complex activity or variable The conventional real valued activation function is defined as follows:

$$f(x) = \frac{1}{1+e^{-x}} \tag{51}$$

Here, in the case of second generation neural network, we use A-P type activation. The mathematical representation as follows:

$$f(z) = \frac{1}{1+e^{-z}} \tag{52}$$

Now derivatives of a logistic neuron

$$\frac{df}{dz} = \frac{d}{dz} (1 + e^{-z})^{-1} = \frac{-1(-e^{-z})}{(1+e^{-z})^2}$$

$$= \left(\frac{1}{1+e^{-z}}\right) \left(\frac{e^{-z}}{1+e^{-z}}\right) = f(z)(1 - f(z)) \tag{53}$$

Two-dimensional activation function shows the output magnitude in the following Fig.7. Containing real and imaginary of the input for the complex function that is given in Eq. (52)

12. Methodology

In proposed research real datasets are selected from available repository. Normalize the datasets between (0,1 or 1, -1) for training and testing in real domain. In complex domain all the normalized real datasets converted in complex points using Hilbert transformation. Experiments are done on training datasets and selecting the computational algorithm viz. FCM and EFC-MD for getting the optimal

number of clusters in their respective domains. RBP (real back propagation / CBP (complex back propagation) learning algorithm is used with optimal number of clusters that are feeded as input in FMNN (real domain)/CFMNN (complex domain) respectively and implements (FMNN/CFMNN) as a trained (FMNN/CFMNN) which are shown in Fig. 8. Output of (FMNN/CFMNN) and testing datasets are given as a input in trained (FMNN/CFMNN). A decision neural activity Integrator interrogates these trainee (FMNN/CFMNN) and recognizes as desirable outputs.

13. Experimental results

Initially the proposed work is simulated as a real valued neural network (RVNN) and then extended in complex domain using Hilbert transformation for generating this RVNN in complex form that is called as complex valued neural network (CVNN). For supporting the complex domain (second generation neural network) analysis of various representation methods have been developed [1, 3, 5, 6, 8, 10, 12, 14]. Further the application of various computational learning algorithms which have been fully proved and introduced by many developers [15, 16, 20, 21, 27, 29]. Next the above whole ideas is extended in the practical domain and the experiment is done over the two real datasets problems (monk and wine) with the help of these computational learning methods and fusion with the application of functional modular neural network separately in respective domain (RVNN/CVNN) for evaluating the performance.

The idea of the different computation learning algorithm in real domain has been taken by a researcher [27] viz. fuzzy c-means clustering (FCM), evolutionary fuzzy clustering with Minkowski distance (EFC-MD) and fusion of EFC-MD with functional modular neural network (EFCMD-FMNN) and extended this in complex computational neural field with a new transformation logic called as Hilbert.

For evaluating the performance of proposed methods, the experiment is done on two slandered datasets using UCI machine learning repository [28].

Here we select the two type of existing real domain datasets which are explained in detail below in 13.1 and 13.2. Normalised the datasets in a predefined range as (0, 1or -1, 1). Next, select the sets of these data points for cluster formation in both the domain and apply different evolutionary approach viz. membership function, population initialisation, computation of objective function and selection for cluster optimization. Further, training of the neural network(RVNN/CVNN) is performed with the help

of FMNN/CFMNN in their respective domains and find out the testing and training accuracies for evaluating the performance in both domains (real and complex).

13.1 Monk data sets problem

The monk problem is formulated based on the international learning algorithm. Monk problems are existing in three kinds of problems in which third kind of problem [28] is the most challenging and latest one due to its noise and complexity. There are 554 patterns are available in the database with seven types of features assigned to two classes. To made fare comparison Yang [29] divides training and testing date in suitable forms. There are 122 patterns are selected for trading and 432 patterns for testing among 554 patterns. Different types of computational learning algorithm viz. evolutionary fuzzy clustering with Minkowski distance (EFC-MD). Collaboration of EFC-MD with functional modular neural network (EFCMD-FMNN) has been implemented along with fuzzy c-means clustering (FCM) in their respective domains. The comparative study is providing in the table .1. In complex domain CEFCMD-CFMNN yields maximum accurately over testing data.

Table 1. Monk data set accuracy comparison

Domain	Algorithms/Method	Accuracy (%)	
		Training Set	Test set
Real Domain	FCM	81.3	83.22
	EFC-MD	100	96.34
	EFCMD-FMNN	100	97.08
Complex Domain	CFCM	84.00	85.91
	CEFC-MD	100	97.67
	CEFCMD-CFMNN	100	98.02

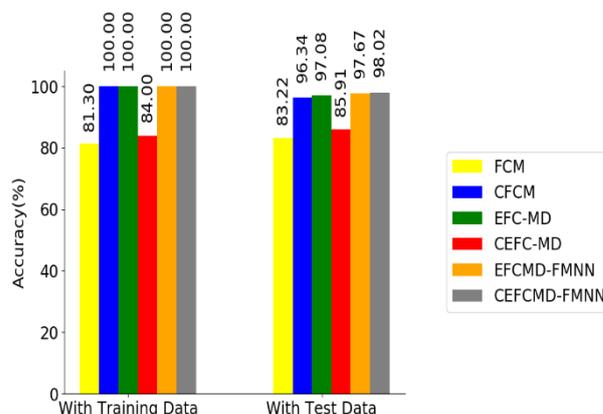


Figure. 9 Accuracy percentage between training data and test data

Table 2. Wine data set accuracy comparison

Domain	Algorithms/Method	Accuracy (%)	
		Training Set	Test set
Real Domain	FCM	86.54	85.36
	EFC-MD	97.04	97.32
	EFCMD-FMNN	100	97.38
Complex Domain	CFCM	88.13	89.89
	CEFC-MD	98.06	98.75
	CEFCMD-CFMNN	100	98.77

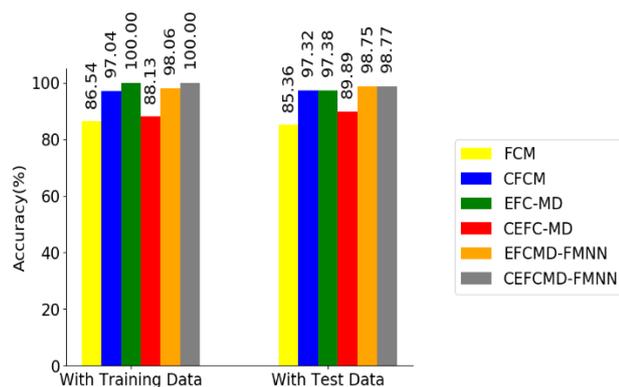


Figure. 10 Accuracy percentage between training data and test data

13.2 Wine datasets problem

Based on outcome of the chemical analysis of wine the wine data set containing 13 constituents in three different kinds of wine classes [28]. Data repository of wine are the collection of total 178 data points of all their classes. There are about 40% of data for training and rest 60 % data for testing on the proposed experiment. The comparative analysis for the performance FCM, EFC-MD and EFCMD-FMNN are given in the table 2. in their respective domains. The results show CEFCMD-CFMNN perform better in complex domain.

14. Conclusion

This paper has developed computational learning algorithms in complex domain (second generation neural network) and proven their superiority over existing real domain (first generation neural network). For the analysis of the datasets in complex domain, used a new approach Hilbert transform for converting real data in complex form. The experiments demonstrate the better learning characteristics in complex domain in comparison with the real domain in the various kind of benchmark problems. In the proposed research complex domain computational learning algorithm viz CFCM, CEFC-MD and CEFCMD-CFMNN yield efficacy over the real

domain computational learning algorithm viz. FCM, EFC-MD and EFCMD-FMNN. Further, it is observed that fusion of EFC-MD with FMNN maintains a supervised learning and good accuracy condition than a single EFC-MD as due to of unsupervised nature of EFC-MD in their respective domains. In future the proposed research further extended in three dimension (vector valued neural network) and four dimension (quaternion valued neural network) or higher dimensions.

References

- [1] N. N. Aizenberg, Yu. L. Ivaskiv, and D. A. Pospelov, *About one generalization of the threshold function*, Doklady Akademii Nauk SSSR (The Reports of the Academy of Sciences of the USSR), Vol. 196, No 6, 1971, pp. 1287-1290.
- [2] B. Widrow, J. McCool, and M. Ball, "The complex LMS algorithm", *Proceeding of the IEEE*, Vol.63, No.4, pp.719-720, 1975.
- [3] I. Aizenberg, N. Aizenberg, and J. Vandewalle, *Multi-valued and universal binary neurons: the ory, learning, applications*, Kluwer Academic Publishers, Boston/Dordrecht/London, 2000.
- [4] I. Aizenberg and C. Moraga, "Multilayer feed forward neural network based on multi-valued neurons (MLMVN) and a back-propagation learning algorithm", *Soft Computing*, Vol.11, No. 2, pp.169-183, 2007.
- [5] M. S. Kim and C. C. Guest, "Modification of backpropagation networks for complex-valued signal processing in frequency domain", In: *Proc. of the 1990 IJCNN International Joint Conference on Neural Networks*, 1990.
- [6] H. Leung and S. Haykin, "The complex backpropagation algorithm", *IEEE Transaction on Signal Processing*, Vol. 39, No.9, pp. 2101 – 2104, 1991.
- [7] T. Nitta, "An Extension of the Back-Propagation Algorithm to Complex Numbers", *Neural Networks*, Vol.10, No.8, pp. 1391-1415, 1997.
- [8] F. Piazza and N. Benvenuto, "On the complex Backpropagation Algorithm", *IEEE Transaction on Signal Processing*, Vol.40, No.4, pp.967-969, 1992.
- [9] T. Kim and T. Adali, "Approximation by fully complex multilayer perceptrons", *Neural Computation*, Vol.15, pp.1641-1666, 2003.
- [10] B. K. Tripathi and P. K. Kalra, "The Novel Aggregation Function Based Neuron Models in Complex Domain", *Soft Computing Springer*, Vol. 14, No. 10, pp. 1069 – 1081, 2010.

- [11] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feed forward neural networks in relation to signal coherence", *IEEE Transactions on Neural Networks and Learning Systems*, Vol.23, No.4, pp.541–551, 2012.
- [12] T. Nitta, "Local Minima in Hierarchical Structures of Complex-Valued Neural Networks", *Neural Networks*, Vol.43, pp.1-7, 2013.
- [13] A.Hirose, *Complex-Valued Neural Networks: Advances and Applications*, John Wiley & Sons Inc. 2013.
- [14] I. Aizenberg, *Complex-Valued Neural Networks with Multi-Valued Neurons*, Springer Publishing Company, Incorporated, ISBN 3662506319, 9783662506318, 2016.
- [15] D. Bhattacharjee, D.K. Basu, M. Nasipuri, and M. Kundu, "Human face recognition using fuzzy multilayer perceptron", *Soft Comput*, Vol.14, No.6, pp.559–570, 2010.
- [16] Y. Ozbey, R. Ceylan, and B. Karlik, "A fuzzy clustering neural network architecture for classification of ECG arrhythmias", *Computers in Biology and Medicine*, Vol.36, pp.376–388, 2006.
- [17] S.A. Mingoti and J.O.Lima, "Comparing SOM neural network with Fuzzy c-means, K-means and traditional hierarchical clustering algorithms", *European Journal of Operational Research*, Vol.174, No.3, pp. 1742–1759, 2006.
- [18] E.R. Hruschka, R.J.G.B. Campello, A.A. Freitas, and A.C.P.L.F. de Carvalho, "A survey of evolutionary algorithms for clustering", *IEEE Trans. System Man Cybernet Part C: Appl. Rev.*, Vol.39, No.2, pp.133–155, 2009.
- [19] L.Y. Tseng and S.B. Yang, "A genetic approach to the automatic clustering problem", *Pattern Recognition*, Vol. 34, pp.415–424, 2001.
- [20] B. Yuan, G.J. Klir, and J.F. Stone, "Evolutionary fuzzy c-mean clustering algorithm", In: *Proc. of International Conference on Fuzzy System*, Vol.35, pp. 2221–2226, 1995.
- [21] J.C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*, Plenum Press, New York, N.Y.1981.
- [22] N.E. Huang and N.O.A. Okine, *The Hilbert-Huang Transform in Engineering*, CRC Press, 2005.
- [23] S. L. Hahn, *Hilbert Transforms in Signal Processing*, Artech House Inc, Boston, London, 1996.
- [24] M. Klingspor, *Hilbert transform: Mathematical theory and applications to signal processing*, the Linköping University, LiTH - MAT - EX, November 19, 2015.
- [25] M.T. Lacey, E.T. Sawyer, C-Y. Shen, and I. Uriarte-Tuero, "Two weight inequality for the Hilbert transform: A real variable characterization, I", *Duke Math. J.* Vol.163, No.15, pp.2795–2820, 2014.
- [26] F. Piazza and N. Benvenuto, "On the complex Backpropagation Algorithm", *IEEE Transaction on Signal Processing*, Vol.40, No.4, pp.967-969, 1992.
- [27] V.Srivastava, "Computational Intelligence Techniques: Development, Evaluation and Applications", PhD Thesis (Thesis defended on 26 April, 2014), Uttar Pradesh Technical University, Lucknow, 2014.
- [28] UCI machine learning repository, URL:<http://archive.ics.uci.edu/ml/datasets>
- [29] X. Yang, G. Zhang, J. Lu, and J. Ma, "A Kernel Fuzzy c-Means Clustering-Based Fuzzy Support Vector Machine Algorithm for Classification Problems with Outliers or Noises", *IEEE Trans. on Neural Network*, Vol. 19, No. 1, pp. 105-115, 2011.