

# DESIGN AND IMPLEMENTATION OF BINARY NEURAL NETWORK LEARNING WITH FUZZY CLUSTERING

Sachin Bhandari<sup>1</sup> and Dr. Aruna Tiwari<sup>2</sup>

<sup>1</sup>Department of Computer Engineering, SGSITS, Indore, India  
er.bhandari04@gmail.com

<sup>2</sup>Department of Computer Engineering, SGSITS, Indore, India  
atiwari@sgsits.ac.in

## ABSTRACT

*In this paper, Design and Implementation of Binary Neural Network Learning with Fuzzy Clustering (DIBNNFC), is proposed to classify semisupervised data, it is based on the concept of binary neural network and geometrical expansion. Parameters are updated according to the geometrical location of the training samples in the input space, and each sample in the training set is learned only once. It's a semisupervised based approach, the training samples are semi-labelled i.e. for some samples, labels are known and for some samples data labels are not known. The method starts with classification, which is done by using the concept of ETL algorithm. In classification process various classes are formed. These classes classify samples in to two classes after that considers each class as a region and calculates the average of the entire region separately. This average is centres of the region which is used for the purpose of clustering by using FCM algorithm. Once clustering process over labelling of semi supervised data is done, then whole samples would be classify by (DIBNNFC). The method proposes here is exhaustively tested with different benchmark datasets and it is found that, on increasing value of training parameters number of hidden neurons and training time both are getting decrease. The result reported, using real character recognition data set and result will compare with existing semi-supervised classifier, the proposed approach learned with semi-supervised leads to higher classification accuracy.*

## KEYWORDS

*Semisupervised classification, Geometrical Expansion, Binary Neural Network, Fuzzy C-means algorithm, ETL algorithm.*

## 1. INTRODUCTION

Recently, the back propagation learning (BPL) algorithm has been applied to many binary-to-binary mapping problems [6], [2]. However, since the BPL algorithm searches the solution in continuous space, the BPL algorithm applied to binary-to-binary mapping problems results in long training time and inefficient performance. Typically, the BLTA algorithm require an extremely high number of iterations to obtain even a simple binary-to-binary mapping [3]. Also, in the BLTA algorithm, the number of neurons in the hidden layer required to solve a given problem is not known a priori. Since the numbers of neurons in the input and the output layer are determined by the dimensions of the input and output vectors, respectively, the abilities of three-layer neural networks depend on the number of neurons in the hidden layer. Therefore, one of the most important problems in application of three-layer neural networks is to determine the necessary number of neurons in the hidden layer. It has been widely recognized that Stone-Weierstrass's theorem does not give a practical guideline in determining the required number of

neurons [4]. In this paper, the geometrical learning algorithm called expand-and-truncate learning (ETL) is proposed with fuzzy c-means algorithm to train a three-layer binary neural network (BNN) for the generation of a semisupervised classifier.

In this paper “Design and implementation of Binary Neural Network learning with fuzzy clustering (DIBNNFC)” an approach to classify semisupervised data is proposed. The work present in this paper is based on binary neural network framework proposed by Jung H. Kim and Sung-Kwon Park, Member, IEEE in “The Geometrical learning of binary neural networks and weiling cai, Sangaon Chen, Daoqiang Zhang in Robust Fuzzy Relational classifier Incorporating the Soft class Labels [5]. ETL algorithm is an approach to cover all true vertices in a neural network. By using this algorithm a linearly inseparable problems is divided into a set of linearly separable problems. In this firstly used ETL algorithm for classify the semi supervised data in to classes, after that these classes considered as a region then for calculating the centre of these region, taking average of the region then this average is considered as a centre of the region which is used in FCM algorithm for Clustering. The second method is clustering which is done by FCM. FCM done labelling of all samples and show that which sample is belong to which region through this the classification of semi supervised data is completed. Existing experimental results character data set is proposed to compare with semi-supervised classifier to be implemented.

## 2. OVERVIEW OF DIBNNFC

### 2.1 Basic Concept

ETL algorithm is a geometrical approach and it is a supervised learning algorithm works for labelled dataset. In this paper, the learning algorithms called expand and-truncate learning (ETL) is proposed to train multilayer binary neural networks (BNN) with guaranteed convergence for any binary-to-binary mapping. By using this algorithm we make a classifier which classifies a semi supervised data, with the help of using FCM algorithm which is an unsupervised learning algorithm.

Boolean functions have the geometrical property which makes it possible to transform non linear representation to linear representation for each hidden neuron. We consider a Boolean function with  $n$  input and one output,  $y = f(x_1, x_2, \dots, x_n)$ , where  $y \in (0, 1)$  and  $x_i \in (0, 1)$ ,  $i = (1, \dots, n)$ .

These  $2^n$  binary patterns  $(0, 1)^n$  can be considered as a  $n - 1$  dimensional unit hypercube.

$$net(X, T) = (w_1x_1 + w_2x_2 + \dots + w_nx_n - T) = 0$$

### 2.2 Working of DIBNNFC

In the system given below the input data is labelled and unlabelled both. Labelled data are those in which label is defined for each sample and unlabelled data are those in which label is not defined for samples. The datasets from the different domain like Iris, Balance Scale, Bupa, and Wine etc are taken for system. As we seen in fig.1 an overview of the system is given, in this the semi labelled data is behave like an input and firstly train by ETL algorithm which is a supervised learning technique and the clustering process is done by FCM algorithm which is an unsupervised technique by the help of these two, an algorithm is proposed to make a binary neural network with fuzzy clustering which work as a semi supervised classifier. This is an overview of the proposed problem of system The goal of semi-supervised learning is to understand how combining labelled and unlabeled data may change the learning behaviour, and design algorithms that take advantage of such a combination

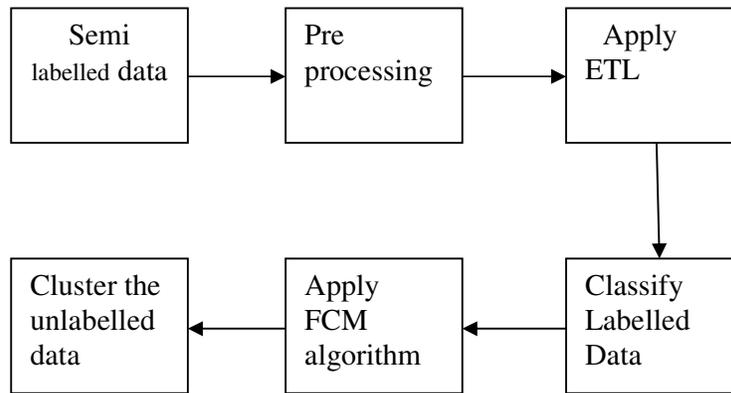


Figure 1. An Overview of the System

### 3. PROPOSED METHOD: DIBNNFC

Suppose that a set of  $n$ -bit training input vectors is given and a binary desired output is assigned to each training input vector. An  $n$ -bit input vector can be considered as a vertex of an  $n$ -dimensional hypercube. The two classes of training input vectors (i.e., vertices) can be separated by an  $(n - 1)$ -dimensional hyperplane which is expressed as a net function,

$$\text{net}(X, T) = (w_1x_1 + w_2x_2 + \dots + w_nx_n - T) = 0$$

where the  $w_i$ 's and  $T$  are constant. The set of training inputs is said to be linearly separable (LS), and the  $(n - 1)$ -dimensional hyperplane is the separating hyperplane. The  $(n - 1)$ -dimensional separating hyperplanes can be established by an  $n$ -input neuron with a hard-limiter activation function as below,

$$y = \begin{cases} 1, & (w_1x_1 + w_2x_2 + \dots + w_nx_n) \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

where,  $Y$  is output of the neuron,  $w_i$  is connection weight between the  $i^{\text{th}}$  input and neuron,  $x_i$   $i^{\text{th}}$  input to the neuron,  $T$  is threshold of neuron.

For learning the hidden layer the geometrical learning is used to decompose an arbitrarily linearly inseparable function into multiple LS functions [1]. For any binary-to-binary mapping, ETL will determine the required LS functions, each of which is realized by a neuron in the hidden layer [8]. Check if there are true and false vertices are at hamming distance equal to one and If only true vertices at hamming distance less than 'd' [7]. The hyperplane can be represented by the equation below.

$$(w_1x_1 + w_2x_2 + \dots + w_nx_n - T) = 0$$

where

$$w_i = 1 \quad \text{if } V_c^i = 1$$

$$w_i = -1 \quad \text{if } V_c^i = 0$$

i.e. 
$$T = w_1 V_c^1 + w_2 V_c^2 + \dots + w_n V_c^n - (d - 1)$$

The first hyperplane is found. The SITV includes the core vertex and the true vertices which are separated by the first hyperplane [12]. Consider a function:  $\{0, 1\}^n \Rightarrow \{0, 1\}$ . Denotes the n-bit input vector and  $\{0, 1\}$  denotes the desired output corresponding to the n-bit input vector. The value of f divides the  $2^n$  points of n-tuple (i.e.,  $2^n$  vertices of n-cube) into two classes, those for which f is zero and those for which it is one. A function f is linearly separable if and only if there exists a hypersphere such that all true vertices lie inside or on the hypersphere and all false vertices lie outside or vice versa. Consider a reference hypersphere and an n-dimensional hypersphere which has its centre at  $(c_1/c_0, c_2/c_0, c_3/c_0, \dots, c_n/c_0)$  and its radius r.

$c_0$  = number of elements in SITV including trial vertex.

$$c_i = \sum_{k=1}^{c_0} V_k$$

where,  $V_k$  is an element in SITV  
 $V_{k,i}$  is  $i^{\text{th}}$  bit of vertex  $V_k$ .

The point  $(c_1/c_0, c_2/c_0, c_3/c_0, \dots, c_n/c_0)$  in the n-dimensional space represents the centre of gravity of all elements in SITV. The separating hyperplane can be represented as

$$(2c_1 - c_0)x_1 + (2c_2 - c_0)x_2 + \dots + (2c_n - c_0)x_n - T = 0$$

where T is a constant called threshold. That is, if there exists a separating hyperplane

For learning an output layer if required separating hyperplanes are found with only one core vertex, the weights and threshold of one output neuron are set as follows. The weight of the link from the odd-numbered hidden neuron to the output neuron is set to 1. The weight of the link from the even-numbered neuron to the output neuron is set to -1. For calculating the center of the class average method is used. Taking an average of entire region of classes, this existing center is used for clustering as an input centers in FCM algorithm.

Fuzzy c-means (FCM) [9], [10], [11] is a data clustering technique in which a data set is grouped into n clusters with every data point in the dataset belonging to every cluster will have a high degree of belonging or membership to that cluster and another data point that lies far away from the center of a cluster will have a low degree of belonging or membership to that cluster. It is based on the concept of fuzzy C-partition which was introduced by Ruspini and developed Dunn and generalized by Jim Bezdek in 1981 as an improvement over earlier clustering methods. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C (u_j(x_i))^m \|x_i - c_j\|^2, 1 \leq m < \infty$$

where  $m$  is any real number greater than 1,  $u_{ij}$  is the degree of membership of  $x_i$  in the cluster  $j$ ,  $x_i$

is the  $i^{\text{th}}$  of  $d$ -dimensional measured data,  $c_j$  is the  $d$ -dimension center of the cluster, and  $\|*\|$  is any norm expressing the similarity between any measured data and the center. The FCM algorithm is composed of following steps:

**Step 1:** The FCM algorithm starts with a set of randomly initializing cluster membership matrix and with a set of predefined number of cluster center.

**Step 2:** Using the randomly initialize cluster membership matrix, the cluster center are calculated.

$$c_j = \frac{\sum_{i=1}^N [u_j(x_i)]^m x_i}{\sum_{i=1}^N [u_j(x_i)]^m}$$

**Step 3:** Using the obtained cluster center, the cluster membership matrix is updated by taking the fractional distance from the point to the cluster center.

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

The fuzzy c-means algorithm imposes a direct constraint on the fuzzy membership function i.e. the total membership for a point in sample or decision space must add to 1. The Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership  $u_{ij}$  and the cluster centers  $c_j$ . This iteration will stop when  $\|U^{k+1} - U^k\| < \epsilon$  where  $\epsilon$  is a termination criterion between 0 and 1, whereas  $k$  are the iteration steps. This procedure converges to a local minimum or a saddle point of  $J_m$ .

### 3.1 Steps for proposed approach as follows:

1. Firstly select a core vertex initially
2. Calculate number of bit difference between core vertex and rest of the vertex
3. Select shortest distance vertex as final core vertex
4. Core vertex tends two SITV and rest of vertex are put in to rest
5. Calculate distance between SITV and rest vertex
6. Select trial vertex whose output is 1 from rest whose distance from core vertex is less than other vertex in rest
7. Find hyperplane equation by using this formula:-  
 $(w_1 x_1 + w_2 x_2 + \dots + w_n x_n - T) = 0$

here If  $V_{c,i} = 1$  then  $w_i > 0$  else  $w_i < 0$

where

$$\begin{aligned} w_i &= 1, \text{ if } F(V_i) = 1 \text{ and } V_{e_i} = 1, \\ w_i &= -1, \text{ if } F(V_i) = 1 \text{ and } V_{e_i} = 0, \\ w_i &= 1, \text{ if } F(V_i) = 0 \text{ and } V_{e_i} = 0, \\ w_i &= -1, \text{ if } F(V_i) = 0 \text{ and } V_{e_i} = 1, \end{aligned}$$

8. Calculate  $t_{\min}$  and  $f_{\max}$

$$T = t_{\min} + f_{\max} / 2$$

9. If  $t_{\min} > f_{\max}$  hyperplane is exist then SITV = trial vertex, SITV removed trial vertex from rest save equation & calculate threshold  
else select new trial vertex until all true output vertices are covered Goto 6 until all true vertices are covered.
10. If rest contain vertex whose output is 1 then convert true in to false & false into true  
Goto step 6.  
else find weight of output layer by putting the weight of the link from the odd numbered hidden neuron to the output neuron is set to 1. The weight of the link from the even-numbered neuron to the output neuron is set to -1
11. Calculate the average of the classes to find the centers of the region, and these centers are used in FCM
12. Initialize  $U = [u_{ij}]$  matrix,  $U^{(0)}$
13. At  $k^{\text{th}}$ -step: calculate the centers vectors  $C^{(k)} = [c_j]$  with  $U^{(k)}$

$$c_j = \frac{\sum_{i=1}^N [u_j(x_i)]^m x_i}{\sum_{i=1}^N [u_j(x_i)]^m}$$

14. Update  $U^{(k)}, U^{(k+1)}$

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

15. If  $U^{(k+1)} - U^k \leq \epsilon$  then STOP; otherwise return to step 13.

#### 4. EXPERIMENT AND RESULTS

The proposed approach is implemented in Matlab 7.8.0(R2009a) and applied on various data sets. Results of these experiments are summarized in Table 1. In this section the proposed DIBNNFC neural network have been tested on dataset and the results are obtained. Datasets used for training are Iris, Balance Scale, Bupa, and Wine data set. The input data sets files are pre processed, which provide require input set for the learning. The input data set files is converted into Binary data by performing Normalisation. Proposed approach determines number of iterations in obtaining cluster center. Furthermore we use Output of ETL to decide that which cluster goes to which class after classification and calculate the accuracy of the classifier.

Table 1. showing results of DIBNNFC classifier for various Datasets

Dataset	Number of Instances	Number of Classes	Number of Features	Iterations	Training time in seconds	Classification Accuracy
Iris	150	3	4	33	0.175633	96%
Balance Scale	625	3	4	8	0.238572	44.55%
BUPA	345	2	6	166	1.007675	58.14%
Wine	178	3	13	56	0.69163	68.54%

## 5. CONCLUSIONS

This paper, “Design and implementation of Binary Neural Network learning with fuzzy clustering (DIBNNFC)” is proposed to classify semi supervised data and train multilayer BNN for any binary-to-binary mapping. It improves significant performance as compared to the previous framework. We have shown that for any generation of binary-to-binary mapping, the proposed DIBNNFC approach always converges and finds the three-layer BNN, by automatically determining a required number of neurons in the hidden layer. Fcm done Clustering for semi labeled data and provide iterations and classification accuracy as per Input data. Any semi supervised data can be classified with this classifier. The neuron in the proposed BNN employs a hard-limiter activation function, only integer weights and integer thresholds. It leads to a higher accuracy of classification. DIBNNFC is better approach to classify semi supervised data then others.

## REFERENCES

- [1] Jung H. Kim and Sung-Kwon Park, “The Geometrical Learning of Binary Neural Networks,” *IEEE T-mn. Neural Networks*, Vol. 6, No.1. Pp.237-247, Jun. 1995.
- [2] J. M. Zurada and W. Shen, “Sufficient condition for convergence of a relaxation algorithm in actual single-layer neural networks,” pp. 3W303, Dec. 1990
- [3] D. L. Gray and A. N. Michel, “A training algorithm for binary feed forward neural network,” *IEEE Trans. Neural Networks*, pp. 176194, Mar. 1992
- [4] N. E. Cotter, “The Stone-Weierstrass theorem and its application to neural networks,” *IEEE Trans. Neural Networks*, Dec. 1990.
- [5] Weiling Cai, Songcan Chen \*, Daoqiang Zhang, “Robust fuzzy relational classifier incorp- orating the soft class labels” Received 29 July 2006; received in revised form 3 June 2007 June 2007 Available online 6 August 2007
- [6] S. Oh and R. J. Mark 11, “Dispersive propagation skew effect in iteration neural networks ‘*IEEE Trans. Neural Networks*, pp. 16C-162, Jan. 1991
- [7] D. Wang and Narendra S. Chaudhari, “A Multi-core Learning algorithm for Boolean Neural Networks”, School of Computing Engineering, Block N4-2a-32, Nanyang Technological University, Singapore 639798-2003 IEEE
- [8] Narendra S. Chaudhari and (Mrs) Aruna Tiwari, “Extending ETL for multi-class output”, Nanyang Technological University, Singapore 639798, Neural Information Processing (ICONIPOZ), Vol. 4
- [9] M. Setnes and R. Babuska, “Fuzzy relational classifier trained by fuzzy clustering,” *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 29, no. 5, pp. 619–625, Oct. 1999.
- [10] Maedeh zirak Javanmard, “Fuzzy C-means algorithm and its application in case of public database of automobile property information. 2010[Online]. Available: [http:// www.authorstream.com/Presentation / aSGuest 80031-738704-fuzzy-c-means-clustering](http://www.authorstream.com/Presentation/aSGuest80031-738704-fuzzy-c-means-clustering).

- [11] W. Pedrycz and G.Vukovich, "Fuzzy clustering with supervision," Pattern Recognition., vol. 37, pp. 1229–1349, 2004.
- [12] Atsushi Yamamoto, Toshimichi Saito, "An Improved Expand-and-Truncate Learning", EEE Dept., HOSEI University, Aug 1997 IEEE.

### Authors

**Sachin Bhandari** received her Bachelor of Engineering degree in Computer Engineering from RGPV University, India in 2010. He is currently pursuing Master of Engineering in Computer Engineering from SGSITS, Indore, India. His research interests include Data mining, and Soft Computing.



**Dr. Aruna Tiwari** She is currently working as Associate Professors in Computer Engineering Department at SGSITS Indore, India. Her research interest areas are Data mining, Computational Learning and Soft Computing.

