PREDICTING PROTEIN LOCALIZATION SITES IN *ESCHERICHIA COLI* BACTERIA

Latha Parthiban[1], Rangasamy Parthiban [2]

[1]SSN College of Engineering, Kalavakkam - 603110
[2]Sri Venkateswara College of Engineering, Sriperumbudur - 602 105
Email: [1]lathap@ssn.edu.in and lathaparthiban@yahoo.com

Abstract

In this paper, three different neural network structure which are Self Organizing Map (SOM), Probablistic Neural Network (PNN) and Radial Basis Function (RBF) were applied to the *Escherichia coli* benchmark and their efficiency in classifying the dataset has been obtained. Then the dataset is applied to the proposed coactive neuro-fuzzy inference system (CANFIS) model integrated with genetic algorithm and better classification with less MSE is obtained when tested using replicative testing.

Key words: Self Organizing Map, Probabilistic Neural Network, Radial Basis Function, CANFIS, *Ecoli*.

I. INTRODUCTION

The *Ecoli* bacteria data have a distinctive characteristic as a classification benchmark because it contains eight different classes and each class contains different number of test and train data. Two of the data classes have only two samples. The data is divided into two sets, one for training and one for testing. It is very hard to classify a test data if network is trained by using only one training data. The previous results for the ecoli benchmark [1,2] have been obtained by ad hoc structured probability model [3,4,5], binary decision tree and Bayesian classifier methods. In this work, the same ecoli data were classified by artificial neural networks like PCA, SOM and SVM and then by CANFIS and best classification result of 87.23% has been obtained.

II. *ESCHERICHIA COLI* DATA BENCHMARK

Escherichia coli is a bacterium and some kinds of E.coli have powerful toxic for human and animal health. The dataset [1] in this paper were taken from Nakai and maintaned by Horton in 1996. E.coli dataset has 336 instances divided into 8 classes. Each instance is identified by a sequence name and eight attributes (Table 1).

The first attribute is not for classification and hence only seven attributes were used. The eight classes and their data number of the ecoli bacteria are given in Table 2.

Table 1. Attributes in E-Coli dataset

Attribute	Description
Sequence name	Accession number for the SWISS-PROT database
mcg	McGeoch's method for signal sequence recognition
gvh	Von Heijne's method for signal sequence recognition
lip	Von Heijne's Signal Peptidase II consensus sequence score
chg	Presence of charge on N-terminus of predicted lipoproteins
aac	Score of discriminant analysis of the amino acid content of outer membrane and periplasmic proteins
alm1	Score of the ALOM membrane spanning region prediction program
alm2	Score of ALOM program after excluding putative cleavable signal regions from the sequence

Table 2. Classes and their Data Distributions

Class	Description	Total Data
cp	Cytoplasm	143
im	inner membrane without signal sequence	77
pp	Periplasm	52
imU	inner membrane with uncleavable signal sequence	35
om	outer membrane	20
omL	outer membrane lipoprotein	5
imL	inner membrane lipoprotein	2
imS	inner membrane with cleavable signal sequence	2

In the past usage of the dataset, a probabilistic classification system by Horton and Nakai [3], an expert system and a knowledge based system by Nakai and Kanehisa [4,5] are applied to classify the dataset. Nakai and Kanehisa also applied binary decision tree, and Bayesian classifier methods for the ecoli dataset classification but this work was not published [1]. Avci and Yildirim [6] used PNN for classifying E-coli database. The most accurate result of ecoli classification problem is 86.3% using evolutionary computation [7] and a new evaluation method of the dataset is introduced in [2]. The accuracy results on E- coli dataset cited in the literature is presented in Table 3.

Table 3. Results on E-coli database

Author	Network Type	Accuracy (%)
Avci and Yildirim [6]	PNN	82.97
Horton and Nakai [3]	k-nearest neighborhood	86
Watkins and Boggess [7]	Evolutionary Computation	86.3
Horton and Nakai [8]	Adhoc probability	81

III. APPLIED ANN STRUCTURES

A. SOM

Self-organizing maps learn the topology of their input vectors and neurons next to each other in the network learn to respond to similar vectors. The layer of neurons can be imagined to be a rubber net that is stretched over the regions in the input space where input vectors occur. Self-organizing maps allow neurons that are neighbors to the winning neuron to output values. Thus the transition of output vectors is much smoother than that obtained with competitive layers, where only one neuron has an output at a time.

B. PNN

The PNN introduced by Specht [9] is essentially based on the well-known Bayesian classifier technique commonly used in many classical pattern-recognition problems. Consider a pattern vector with m dimensions that belongs to one of two categories K_1 and K_2. Let $F_1(x)$ and $F_2(x)$ be the probability density functions for the classification categories K_1 and K_2, respectively. From Bayes' discriminant decision rule, x belongs to K_1 if

$$\frac{F_1(x)}{F_2(x)} > \frac{L_1}{L_2}\frac{P_2}{P_1} \qquad ...(1)$$

Conversely, x belongs to K_2 if

$$\frac{F_1(x)}{F_2(x)} < \frac{L_1}{L_2}\frac{P_2}{P_1} \qquad ...(2)$$

where L_1 is the loss or cost function associated with misclassifying the vector as belonging to category K_1 while it belongs to category K_2, L_2 is the loss function associated with misclassifying the vector as belonging to category K_2 while it belongs to category K_1, P_1 is the prior probability of occurrence of category K_1, and P_2 is the prior probability of occurrence of category K_2. In many situations, the loss functions and the prior probabilities can be considered equal. Hence the key to using decision rules given by equations (1) and (2) is to estimate the probability density functions from the training patterns.

C. RBF

RBF's are embedded in a two layer neural neural network where each hidden unit implements a radial activated function. The output unit implement a weighted sum of hidden units. The input to the RBF

network is non-linear while the output is linear. Due to their non-linear approximation properties, RBF networks are able to model complex mappings which perceptron neural network can model with multiple intermediary layers [10].

IV. CLASSIFICATION USING CANFIS

The CANFIS model integrates adaptable fuzzy inputs with a modular neural network to rapidly and accurately approximate complex functions. Fuzzy inference systems are also valuable, as they combine the explanatory nature of rules (MFs) with the power of neural networks. These kinds of networks solve problems more efficiently than neural networks when the underlying function to model is highly variable or locally extreme [11].

The fundamental component of CANFIS is a fuzzy axon, which applies membership functions to the inputs. The output of a fuzzy axon is computed using the following formula:

$$f_j(x, w) = \min \forall i \quad (MF(x_i, w_{ij})) \qquad ...(3)$$

where i = input index, j = output index, xi = input i, wij = weights (MF parameters) corresponding to the jth MF of input i and MF = membership function of the particular subclass of the fuzzy axon. This system can be viewed as a special three-layer feed forward neural network. The first layer represents input variables, the middle (hidden) layer represents fuzzy rules and the third layer represents output variables. The CANFIS architecture used in this study is shown in Fig 1.

A. CANFIS architecture

Consider a CANFIS structure with n inputs and one output. For model initialize, suppose a common rule set with n inputs and m IF-THEN rules as follows [12]

Rule 1: If z_1 is A_{11} and z_2 is A_{12} ... and z_n is A_{1n} then $u_1 = p_{11}z_1 + p_{12}z_2 + ... + p_{1n}z_n + q_1$

Rule 2: If z_1 is A_{21} and z_2 is A_{22} ... and z_n is A_{2n} then $u_2 = p_{21}z_1 + p_{22}z_2 + ... p_{2n}z_n + q_2$

……..

Rule m: If z_1 is A_{m1} and z_2 is A_{m2} . . and z_n is A_{mn} then $u_m = p_{m1}z_1 + p_{m2}z_2 + ... + p_{mn}z_n + q_m$

The corresponding CANFIS structure is illustrated in Fig. 2. All layers in CANFIS structure are either adaptive or fixed. The function of each layer is described as follows:

Layer 1 (Premise Parameters): Every node in this layer is a complex-valued membership function (μ_{ij}) with a node function:

$$O_{1,ij} = |\mu A_j(Z_i)| \angle \mu A_{ij}(Z_i)$$

for $(1 \le i \le n, 1 \le j \le m)$ ...(4)

Each node in layer 1 is the membership grade of a fuzzy set (A_{ij}) and specifies the degree to which the given input belongs to one of the fuzzy sets.

Layer 2 (Firing Strength): Every node in this layer is product of all the incoming signals. This layer

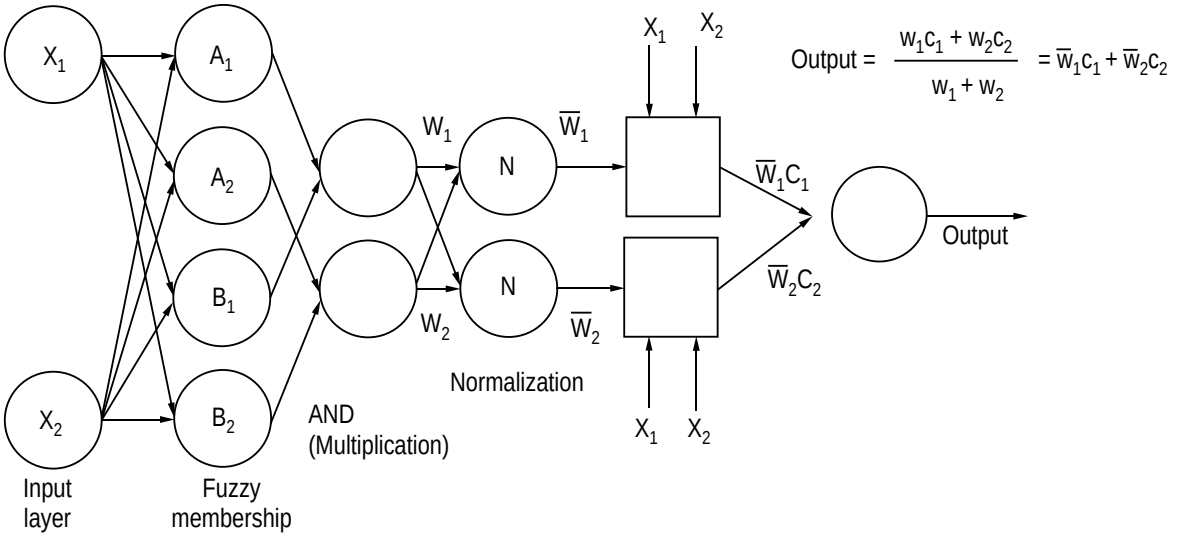

Fig. 1 A prototype two-input one-output CANFIS network and output calculation

receives input in the form of the product of all the output pairs from the first layer:

$$O_{2,j} = W_j = \mu A_{i1}(z_1)\, \mu A_{i2}(z_2), \ldots \mu A_{i1n}(z_{nl})$$

$$\text{for } (1 \le i \le m) \qquad ...(5)$$

Layer 3 (Normalized Firing Strength): **Every node in this layer calculates rational firing strength:**

$$O_{3,j} = \overline{wj} = \sum_{j='1}^{m} W_{ij} \text{ for } (1 \le j \le m0). \qquad ...(6)$$

Layer 4 (Consequence Parameters): Every node in this layer is multiplication of Normalized Firing Strength from the third layer and output of neural network:

$$O_{4,j} = \overline{wj}\, U_j = \overline{wj}\,(P_{j1} Z_1 + P_{j2} Z_2 + \ldots P_{jn} Z_{zn} + q_j)$$

$$\text{for } (1 \le j \le m). \qquad ...(7)$$

Layer 5 (Overall Output): The node here computes the output of CANFIS network:

$$O_{5,1} = \Sigma\, \overline{wj}\, Uj \qquad ...(8)$$

Basically, two membership function types can be used (Gaussian or generalized bell). The bell fuzzy axon used in this study is a type of fuzzy axon that uses a bell-shaped curve as its membership function. Each membership function takes three parameters stored in the weight vector of the bell fuzzy axon (Eq. 9):

$$MF(x, w) = \frac{1}{1 + \left| \frac{x - w2}{wo} \right| 2m} \qquad ...(9)$$

where x = input and w = weight of the bell fuzzy axon.

Fuzzy axons are valuable because their MF can be modified through backpropagation during network training to expedite the convergence. A second advantage is that fuzzy synapses help in characterizing inputs that are not easily discretized. The powerful capability of CANFIS stems from the pattern-dependent weights between the consequent layer and the fuzzy association layer.

The second major component of CANFIS is a modular network that applies functional rules to the inputs. The number of modular networks matches the number of network outputs and processing elements in each network corresponding to the number of MFs. Two fuzzy structures are mainly used: the Tsukamoto model and the Sugeno (TSK) model. Finally, a combiner is used to apply the MF outputs to the modular network outputs. The combined outputs are then channeled through a final output layer, and the error is backpropagated to both the MF and the modular network [13].

B. Genetic optimization

In order to improve the learning of the *CANFIS*, quicker training and enhance its performance, we use genetic algorithms to search for the best number of MF for each input, and optimization of control parameters such as learning rate, and momentum coefficient. This approach also is useful to select the most relevant features of the training data which can produce a smaller and less complicated network, with the ability to generalize on freshly presented data, due to the removal of redundant variables.

The GA combines selection, crossover, and mutation operators with the goal of finding the best solution to a problem by searching until the specified criterion is met. The solution to a problem is called a chromosome, which is composed of a collection of genes. In hybrid neuro-fuzzy-genetic applications, genes are the *CANFIS* parameters to be optimized. The GA creates an initial population and then evaluates

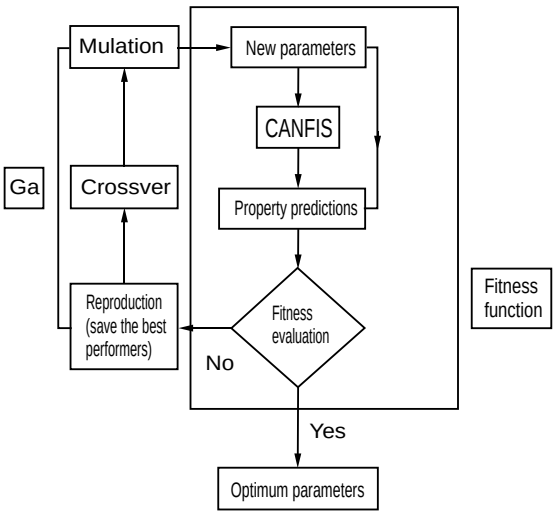

Fig. 2 The CANFIS /GA cycle for Optimization

this population by training a network for each chromosome. It then evolves the population through multiple generations in the search for the best network parameters.

GAs cause the initial population to evolve towards a population that is expected to contain the best solution [14]. We use the following reproduction evaluation cycle for each iteration-referred to as a generation. Chromosomes (individuals) from the current population are selected with a given probability; and copies of these chromosomes (individuals) are created. The selection of chromosomes is based on their fitness relative to the current population; that is, the stronger chromosomes will have a higher probability of being copied. The fitness is a function of the *CANFIS* model's response. Selected chromosomes are subjected to mutation and to crossover. Figure 2 shows the CANFIS/genetic algorithm cycle for search of optimum parameters of the model

These mathematical chromosomes could be operated upon by quasi-genetic processes of crossing over and mutation. To implement crossovers, chromosomes were randomly paired, and segments of paired chromosomes between two randomly determined breakpoints were swapped. Crossovers could be implemented either across genes, so that gene boundaries might potentially be breached by the exchange of genetic material; or within genes, so that gene boundaries would be preserved. Inversions could also be modeled, so that exchanged genetic material could be inverted before becoming incorporated into the recipient chromosome.

Mutations were implemented by flipping a bit at a binary locus, so that a "0" bit was converted to a "1," or a "1" bit was converted to a "0." In this paper, for the optimization of the CANFIS model, GA used the serial method of binary type, roulette-wheel in the selection operator, tow-point crossover in the crossover operator, and boundary in the mutation operator. Automatically determination of the chromosomes length used to optimal search is one of the most important capabilities of the NeuroSolution software. Thus, all the chromosomes were automatically set in this software so that they consisted of the number of input neurons and membership functions, learning rate, and momentum. NeuroSolution also automatically produced their initial values

IV. SIMULATION RESULTS

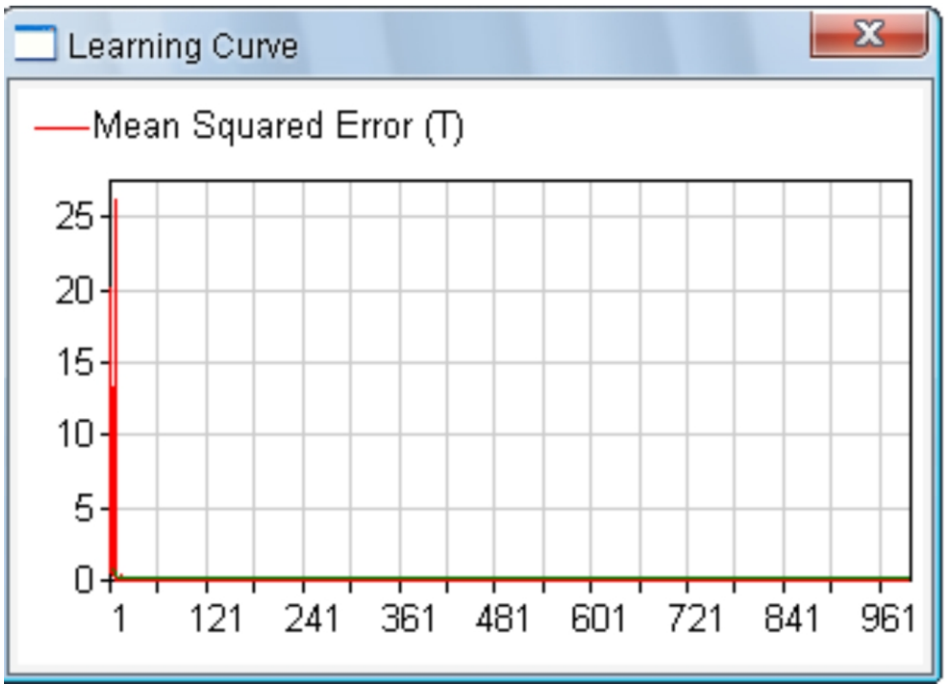

Fig. 3 Learning Curve of CANFIS

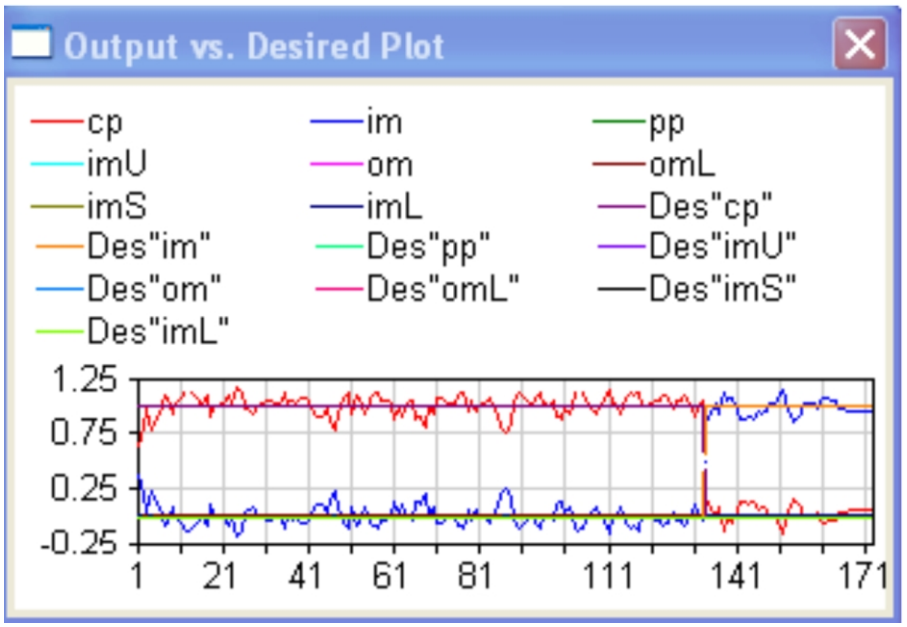

Fig. 4 Output Performance of CANFIS

The simulations were realized by using neurosolution software. The neural networks discussed were tested on E- coli database using replicative testing and the Mean Square Error (MSE) for CANFIS network is 0.00192, whereas MSE for SOM, PNN and RBF were 0.01623, 0.01871 and 0.00873 respectively. The learning curve for CANFIS is shown in Fig. 3 and the output performance of CANFIS is shown in Fig. 4

V. CONCLUSION

While encouraged by the performance of our methods, we believe further work is likely to yield much more comprehensive and accurate models of the Protein Localization Sites. CANFIS gives accuracy of 86.9% on E-coli which is the best accuracy% for this dataset reported till date. The main issue with CANFIS is that it consumes more time, which will be focused in our future work.

ACKNOWLEDGEMENTS

The authors would like to thank Computer Society of India for providing necessary support for this work.

REFERENCES

[1] www.ics.uci.edu/mlearn/MLRepository.html.

[2] Sejong Oh, 2011 A new dataset evaluation method based on category overlap, Computers in Biology and Medicine, vol. 41, Issue 2, pp.115-122.

[3] Horton P. and Nakai K, 1996, A probabilistic classification system for predicting the cellular localization sites of proteins, Intelligent system in molecular biology, vol. 4, pp. 109-115.

[4] Nakai K. and Kanehisa M., 1991, Expert system for predicting protein localization sites in gram-negative bacteria, Proteins: Structure. function and genetics, vol 11, pp. 95-110.

[5] Nakai K. and Kanehisa M, 1992, A Knowledge Base for Predicting Protein Localization Sites in Eukaryotic Cells, Genomics, 14, pp. 897-911.

[6] Avci M and Yildirim T, 2002, Classification of Eschericia Coli Bacteria by Artificial Neural Networks, IEEE Intl. Symp. on Intelligent Systems, Bulgaria, Vol: 3, pp. 16-20.

[7] Watkins A. and L. Boggess, 2002, A Resource Limited Artificial Immune Classifier, In Proceedings of the 2002 Congress on Evolutionary Computation Special Session on Artificial Immune Systems, IEEE Press, vol 1, pp. 926-931.

[8] Horton P. and Nakai K, 1997, Better Prediction of Protein Cellular Localization Sites With the k Nearest Neighbors Classifier, Intelligent Systems for Molecular Biology, vol: 5, pp. 147-152.

[9] Specht D.F, 1990, Probabilistic neural networks and the polynomial ADALINE as complementary techniques for classification, *IEEE Transactions on Neural Networks,* Vol. 1, No. 1, pp. 111-121.

[10] Hyontai Sug, 2010, An objective method to find better RBF networks in classification, 5 IEEE conference on Computer Sciences and Convergence Information Technology (ICCIT), pp.373-376.

[11] Dubois D., Prade H., An introduction to fuzzy systems, Clin. Chim. Acta 270,3–29,1998.

[12] Kuncheva L.I., Steimann F, 1999, Fuzzy diagnosis, Artificial Intelligence in Medicine, vol. 16 , pp. 121–128.

[13] Nauck D., Kruse R., 1999, Obtaining interpretable fuzzy classi;cation rules from medical data, Artificial Intelligence in Medicine, Vol.16 , pp. 149–169.

[14] Jang, J.S.R, 1992, Self-learning fuzzy controllers based on temporal back-propagation, IEEE Transactions in Neural Networks 3 (5) ,pp. 714–723.

Dr. Latha Parthiban obtained her B.E from Madras University, M.E from Anna University and PhD from Pondic2herry central university. She has published 5 books and 9 international journal papers.

Dr. Rangasamy Parthiban completed his B.Tech from CIT M.Tech from REC and PhD from Anna University. He has published 7 books and over 28 international/ national publications.