ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

An Effective Fuzzy Set Theory with Neural Networks for using Feature Selection and Classification

Dr. N. Balakumar[1], A. Vaishnavi[2]

Assistant professor, Dept. Computer Applications, Pioneer College of Arts and Science, Coimbatore, India[1,2]

Abstract: In fuzzy set theory, Fuzzy set theory defines set membership as a possibility distribution. The fuzzy set theory can be used in a wide range of domains in which information is incomplete or imprecise, such as bioinformatics. The uncertainty may arise due to partial information about the problem, or due to information which is not fully reliable, or due to inherent imprecision in the language with which the problem is obtained, or due to receipt of information from more than one source about the problem which is conflicting. Fuzzy set theory is an excellent mathematical tool to handle the uncertainty and vagueness inherent to human perception, speech, thinking and decision making. In this paper used to how to find the error and make the analysis will be made up with the help of neural networks

Keywords: Fuzzy Rules, Fuzzy Classifier, nueral networks, artifial nueral networks.

I.INTRODUCTION

Fuzzy set is categorized based on the different functions of data mining that are modeled. The membership of elements in a set is assessed in binary terms according to a condition an element if either belongs to or does not belong to the set. By contrast, fuzzy set theory permits the gradual assessment of the membership of elements in a set; this is described with the aid of a membership function valued in the real unit interval [0, 1]. Fuzzy sets generalize classical sets, since the indicator functions of classical sets are special cases of the membership functions of fuzzy sets, if the latter only take values 0 or 1.[3] In fuzzy set theory, classical double sets are usually called crisp sets. The fuzzy set theory can be used in a wide range of empire in which information is incomplete or imprecise, such as bioinformatics.[4]

1.1 FUZZY SETS & MEMBERSHIP FUNCTION

A fuzzy set A has a membership function μ_A defined as a function from a well defined universe X into the unit interval as $\mu_{A:}$ X→ [0, 1]. A fuzzy set A in X is also expressed as a set of ordered pairs:

$$A=\{(x,\mu_A(x))\,|\,x\in X\}$$

The degree of membership in a set expressed by a number between 0 and 1; 0 means entirely not in the set,1 means completely in the set, and a number between 0 to 1 means partially in the set [6].

1.2 FUZZY RULE BASED SYSTEM

Fuzzy rules are the cornerstone of the fuzzy logic systems also it has the potential to add human-like subjective reasoning capabilities to machine intelligences, which are usually based on bivalent Boolean logic. A fuzzy if-then rule is a scheme for capturing knowledge that involves imprecision. A Fuzzy Inference System (FIS) is a way of mapping an input space to an output space using fuzzy logic as shown in Figure 1. A FIS tries to formalize the reasoning process of human language by means of fuzzy logic (that is, by building fuzzy IF-THEN rules). FIS are used to solve decision problems. The matching degree is combined with the consequent of the rule to form a conclusion inferred by the fuzzy rule [6]. The Rule base and Data Base are defined in Figure 2.

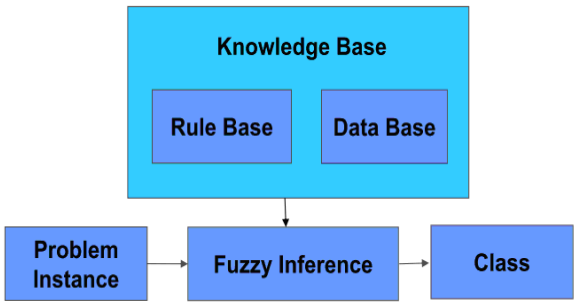


Fig. 1 Fuzzy Inference

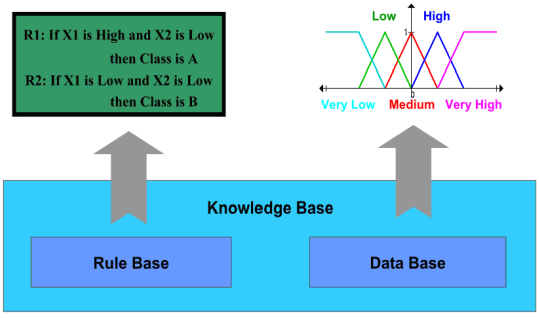


Fig. 2 Fuzzy Inference system components

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

1.3 FUZZY CLASSIFIER

A classifier is an algorithm for that assigns a class label to an object, based on the object description. The object description comes in the form of a vector containing values of the features are (attributes) deemed to be relevant for the classification task. Typically, the classifier learns to predict class labels using a training algorithm and a training data set. When this type of a training data set is not available, a classifier can be designed from prior knowledge and expertise.

Classifier

A classifier is a function D: $R^s \rightarrow \Omega$ where $\Omega = \{\omega_1, \omega_2, ...\omega_n\}$ is a set of class labels. D takes as input an object data i.e., a feature vector in s dimension and assigns a class label to it.

Why fuzzy classifier

The fuzzy classifier is being used in the following situations:

- If there is insufficient information to properly implement classical pattern recognition methods.
- The user needs not only the class label of an object but also some additional information is needed (i.e., how typical this object is, how severe the disease is).
- Fuzzy set theory gives a mathematical tool for including and processing expert opinions about classification decisions, features and objects.

II. NEURAL NETWORKS

Neural Network represents the brain structure and operating with varying degrees of sophistication. This chapter provides an introduction to neural networks and description about the neuron, the network architecture and back propagation algorithm. The processing of input-output network architecture of human brain are shown in Figure 3.

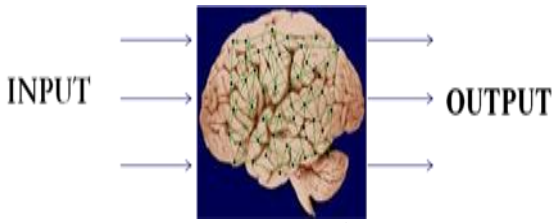


Fig. 3 Network architecture of Human Brain

Neural Networks adopt various learning mechanisms. Learning methods can be broadly classified into three basic types

- Supervised learning
- Unsupervised learning
- Reinforced learning

In Supervised learning, every input pattern that is used to the network is associated with an output pattern, which is the target or the desired pattern. During the learning process, when a comparison is made between the network's computed output and the expected output, to determine the error.

2.1 MODEL OF AN ARTIFICIAL NEURON

The behavior of a neuron can be captured by a simple neuron model as shown in Figure 4.

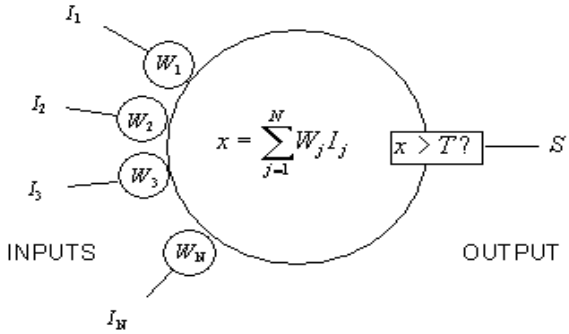


Fig. 4 The Computation of a Neuron

where, $I_1, I_2, I_3, ..., I_n$ are the n inputs to the artificial neuron $W_1, W_2, W_3 ..... W_n$ are the weights attached to the input links. The net sum is

$$x = \sum_{j=1}^{n} w_j I_j$$

To generate the final output S, the sum is passed on to a non-linear filter ϕ called activation function or transfer function, or squash function which releases the output S= $\phi(x)$

The activation adopted in this work is sigmoid function. A sigmoid function is a continuous function that varies gradually between the asymptotic values 0 and 1 or -1 and +1 and is given by $\phi(x) = \frac{1}{1+e^{-\alpha I}}$

where α is the slope parameter, which adjusts the abruptness of the function as it changes between the two asymptotic values. Sigmoidal functions are differentiable, which is an important feature of neural network.

2.2 NEURALNETWORK ARCHITECTURES

An Artificial Neural Network(ANN) is defined as a data processing system consisting of a large number of simple highly interconnected processing elements in an architecture inspired by the structure of the cerebral cortex of the brain.ANN structure can be represented using a directed graph.

The neurons in a network are usually organized into fields or layers. Inputs to the network are presented to the input layer; the signal from the input layer is passed through one or more intermediate or hidden layer which transforms the signals depending upon the neuron signal functions, the outputs of the network are generated in the output layer. The two different classes of neural network are FeedForward networks, Feedback Networks [7].

2.2.1 FEEDFORWARD NETWORK

The network comprises of two layers, namely the input layer and the output layer**.** The input layer neuron receives the input signals and the output layer neurons receive the

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

output signals. The synaptic links carrying the weight connect every input neuron to the output neuron but not vice-versa. Such a network is said to be feedforward.

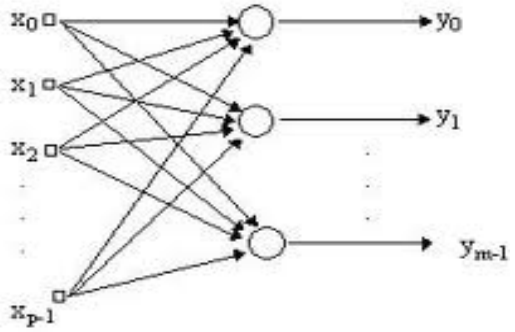


Fig.5 Single layer feedforward

Single-layer FeedForward Networks

In a layered neural network the neurons are organized in the form of layers. In the simplest form of a layered network, we have an input layer of source nodes that projects onto an output layer of neurons but not vice versa. Perceptrons(neurons) can be trained by a simple learning algorithm that is usually called the delta rule. Single-unit perceptrons are only capable of learning linearly separable patterns. Figure 5 & Figure-6 illustrates the single layer feedforward network and its flow graph.

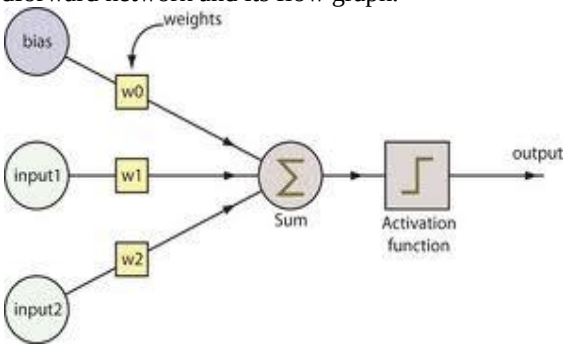


Fig 6 Flow graph

where $x_1, x_2, ..., x_{p-1}$ are the input

$y_0, y_1, ..., y_{m-1}$ are the output

w_0, w_1, w_2 are the weights.

The weights here are multiplicative factors of the inputs to account for the strength of the synapse. To generate the final output, the sum is passed on to a non-linear filter (activation function) which releases the output.

Multilayer FeedForward Networks

The second class of a feedforward neural network distinguishes itself by the presence of one or more hidden layers, whose computation nodes are correspondingly called hidden neurons or hidden units.

The hidden layer aids in performing useful intermediary computations before directing the input to the output layer. The input layer neurons are linked to the hidden layer neurons and the weights on these links are referred to as input-hidden layer weights.

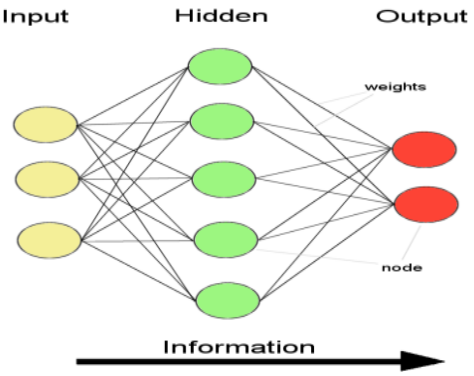


Fig.7 Multilayer perceptrons

The hidden layer neurons are linked to the output layer neurons and the corresponding weights are referred to as hidden-output layer weights. Multilayer perceptrons and its Flow graph are shown in Figure 7 and Figure 8.

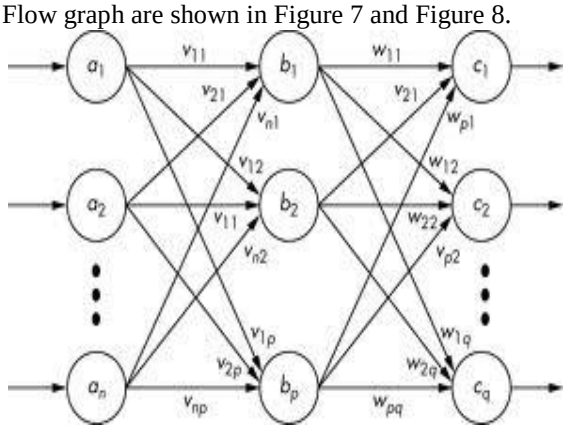


Fig.8 Flow graph

2.2.2 FEEDBACK NETWORKS

A Feedback networks (Recurrent networks) distinguishes itself from a feedforward network in that it has atleast one feedback loop. Recurrent network may consist of a single layer of neurons with each neuron feeding its output signal back to the other neurons. The recurrent network has no hidden neurons.Feedback networks thus possess a rich reperotoire tasks such as pattern completion, topological feature mapping and pattern recognition.

2.3 BACKPROPAGATION NETWORKS

Backpropagation(BP) is the most widely used algorithm for training multilayer feedforward network. The algorithm uses gradient descent technique to adjust the connections of weights between neurons in order to minimize the system error between the actual output and the desired output. Backpropagation updates the weights iteratively to map a set of input vectors $(x_1, x_2, ..., x_p)$ to a set of corresponding output vectors $(y_1, y_2, ..., y_p)$.

The input is presented to the network and multiplied by the weights.

All the weighted inputs fed to each neuron in the upper layer are summed up, and produce output governed by the following equation

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

$$o_{kp} = f(net_{kp}) = f\left(\sum_j \omega_{kj} o_{jp} + \theta_k\right)$$

Where o_{kp} is the output of neuron 'k' for p-th pattern, o_{jp} is the output of neuron 'j' at the lower layer, ω_{kj} is the weight between the neurons 'k' and 'j'. net_{kp} is the net input feeding to neuron 'k' from the lower layer for pattern. 'p', θ_k is the bias for unit 'k' and $f(\cdot)$ is the activation function of the neurons.

The cost function to be minimized in back propagation i.e., the sum of squared error is measured at the output layer are defined as $E = \frac{1}{2}\sum_p \sum_k (t_{kp} - o_{kp})^2$ where t_{kp} - target for pattern 'p' o_{kp} actual outputs of neuron 'k' for pattern 'p'. To calculate error in the output layer, the error back propagation algorithm iteratively changes the weight relative to error size and then propagate back to previous layer. The training of a neural network involves two passes, forward pass and reverse pass.

In the forward pass, the input signals propagate from the network input to the output. In the reverse pass, the calculated error signals propagate backwards through the network to adjust the weights. The calculation of output is carried out layer by layer in the forward direction. The output of one layer in weighted manner will be the input to the next layer. These processes get repeated until the error is acceptably low.

2.4 NEURO-FUZZY SCHEME

Neural networks and fuzzy logic are two complementary technologies. However, understanding the knowledge or the pattern learned by the neural networks has been difficult. More specifically, it is difficult to develop an insight about the meaning associated with each neuron and each weight. Hence, neural networks are often viewed as a "black box" approach. i.e., we can understand what the box does, but not how it is done conceptually. Fuzzy rule-based models are easy to comprehend because it uses linguistic terms and the structure of if-then rules. However, fuzzy logic does not come with a learning algorithm. The learning and identification of fuzzy models need to adopt techniques from other areas (e.g. statistics, linear system identification, etc) [7].

The integration of neural networks and fuzzy systems can yield systems, which are capable of learning and decision making. These systems are called neuro fuzzy systems, and the neural networks are used to improve the performance of fuzzy systems. In the ordinary neural networks, nodes have the same functionality and are fully connected to the nodes in the neighboring layers. But in a neuro fuzzy system, nodes have different functionalities and are not fully connected to the nodes in the neighboring layer.

NNs with fuzzy capabilities, thereby increasing the network's expressiveness and flexibility to adapt to uncertain environments. One merit of the neuro fuzzy systems over the ordinary neural networks is the easiness of adding the expert knowledge before learning. The neuro fuzzy systems are usually built from the given fuzzy systems which are based on the expert or prior knowledge. Thus, the neuro fuzzy systems can embed the knowledge at the beginning. Hence, the neuro fuzzy scheme is more suitable for our work.

III. FEATURE SELECTION & CLASSIFICATION

This chapter deals with the system study by explaining a methodology for simultaneous feature analysis and system identification in a four-layered neuro-fuzzy framework. The network has a methodology to select the important features in the given data set, which forms an important phase for any classification process. All features that characterize a data point may not have the same impact with regard to NN classification. i.e., some features may be redundant and also some may have derogatory influence on the classification task. The selection of a proper subset of features from the available set of features is important for designing efficient classifiers. Features (variables) are used to describe the objects numerically. Feature values for a given object are arranged as an n-dimensional vector $X = [x_1,..x_n]^T \in R^n$. The feature selection is not done in an online-manner. The network method can do feature selection simultaneously with designing the classifier [5], it would be able to select the most appropriate set of features for the task and to produce good results.

3.1 THE NETWORK STRUCTURE

There be S input features and c classes where $X \in R^s$, the neural-fuzzy system deals with fuzzy rules of the form R_i : if x_1 is A_{1i} and x_2 is A_{2i} ,…,and x_s is A_{si} then X belongs to class t_l with certainty $d_l, (1 \le l \le c)$ where A_{ji} is the i th fuzzy set defined on the domain of x_j [1]. The Neural-Fuzzy System is realized using a Four-Layered Network.

- The First layer is the Input layer.
- The Second Layer is the Fuzzification and Feature Analysis layer.
- The Third Layer is the Antecedent layer.
- The Fourth Layer is the Consequent layer.

Layer 1 (Input nodes)
Layer 2 (Fuzzification and feature analysis)
Layer 3 (Antecedent node)
Layer 4 (Consequent layer)

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

3.2 LEARNING PHASES

The learning phases are being categorized into three phases, as phase-I, phase-II and phase-III.

PHASE I Feature selection and rule detection

The concept of back propagation is used to minimize the error function

$$e = \frac{1}{2}\sum_{i=1}^{N} E_i \quad (\text{i.e.}) \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{l=1}^{c}(y_{il} - z_{il})^2$$

where c is the number of nodes in layer 4 and y_{il} and z_{il} are the target and actual outputs of node l in layer 4. In phase I, learnable weights between layer 3 and layer 4 and the parameters $\beta_p s$ in layer 2 are updated based on gradient descent search [4]. Learning rules are derived by the instantaneous error function E_i. The delta value δ of a node in the network is defined as the influence of the node output on E. The derivation of the delta values and the adjustment of the weights and β_p s are presented layer wise next.

LAYER 2

The δ_n for layer 2 is

$$\delta_n = \sum_{m \in R_n} \delta_m \left(\frac{z_m z_n^{q-1}}{\sum_{n \in P_m} z_n^q} \right)$$

where R_n - set of indexes of nodes in layer 3 connected with node in layer 2. The weight update equation and the equation for updating β_p are

$$\frac{\partial E}{\partial g_{lm}} = \begin{cases} \sum_{l \varepsilon Q_m} 2\delta_l z_m g_{lm}, if \\ 0, \quad \text{otherwise} \end{cases}$$

Similarly,

$$z_m g_{lm}^2 = \max_{m'}\{z_{m'} g_{lm'}^2\}$$

$$\frac{\partial E}{\partial \beta_p} = -\sum_{n \varepsilon R_p} \delta_n (2\beta_p e^{-\beta_p^2} z_n)\left(\frac{z_p - \mu_n}{\sigma_n}\right)^2$$

R_p- set of indexes of nodes in layer 2 connected to node p of layer 1.

The update equation for the weights g_{lm} and β_p are $g_{lm}(t+1) = g_{lm}(t) - \eta\left(\frac{\partial E}{\partial g_{lm}}\right)$

$$z_m g_{lm}^2 = \max_{m'}\{z_{m'} g_{lm'}^2\}$$

$$\beta_p(t+1) = \beta_p(t) - \nu\left(\frac{\partial E}{\partial \beta_p}\right)$$

where η and ν are learning coefficients.

LAYER 3

The delta for this layer is

$$\delta_m = \sum_{l \varepsilon Q_m} \delta_l g_{lm}^2, if$$

0, otherwise, where Q_m - set of indexes of the nodes in layer 4 connected with node m of layer 3.

LAYER 4

The output of the nodes in this layer is given by z_l and the δ value for this layer δ_1 will be

$$\delta_1 = \frac{\partial E}{\partial z_l} = -(y_l - z_l)$$

3.3.2 PHASE II

Pruning of redundant nodes and further training

The network in layer 3 and layer 4 represent all possible rules. But all the rules are never needed to represent a system. Hence some of the nodes present in the network may be redundant. The presence of these redundant nodes will decrease the readability/interpretability of the network and add to its computational overhead.

Since each node in layer 3 is connected with all nodes in layer 4, it gives rise to incompatible rules that need to be removed. Consequently, there may be some rules which are never fired by the training data. Such rules which are not supported by the training data could be harmful. The certainty factors of such rules can be erratic and they can lead to bad generalization. So it is necessary to get rid of redundant nodes, incompatible rules and less used rules.

Pruning of redundant nodes

The layer 1 of the network has s nodes, let the indexes of these nodes be denoted by $i (i = 1$ to s).

$$\gamma_p = 1 - e^{-\beta_p}$$

Let N_i be the set of indexes of the nodes in layer 2 and R be the set of 'r' indexes of the nodes, representing the 'r' bad features. Hence any node with index i in layer 1 such that $i \in R$ is redundant. After pruning of the redundant nodes a few epochs of training is required in the reduced architecture for the network to regain its performance [4].

A feature ρ is considered redundant, if

$$\gamma_p < th (th = 0.05)$$

$$th$$

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

where - threshold

sw

3.3.3 PHASE III

Pruning of Incompatible Rules, Less used Rules, and Zero Rules

The network is pruned and the certainty factor of the rules are again tuned to improve the performance. In phase III, the parameters of the membership function is also tuned.

Pruning of Incompatible Rules

As per construction of network, the nodes in layer 3 and layer 4 are fully connected and each link corresponds to a rule. The weight associated with each link is treated as the certainty factor of the corresponding rule. If there are c classes then layer4 will have c nodes and there will be c rules with same antecedent but different consequent, which are inherently inconsistent. The link connecting node m of layer 3 and node l of layer 4 has a weight ω_{lm} associated with it, which is interpreted as the certainty factor of the rule represented by the link. For each node m in layer 3, only one link is retained with a node in layer 4 that has the highest certainty factor associated with it.

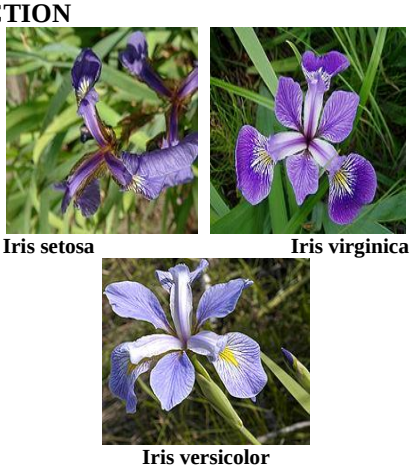
Pruning of Zero Rules and Less used Rules

After removal of the incompatible rules each node in layer 3 is connected with only one node in layer 4. Suppose node m in layer 3, which is connected to a node l in layer 4, has a very low weight

$$W_{lm} < \omega_{low}(\omega_{low} = 0.001).$$

The rule associated with the node pair m and l has a very low certainty factor and it does not contribute significantly in the classification process. Such rules are zero rules. These rules can be removed from the network. It means removing a node in layer 3 along with its links. There may be rules which are never fired or fired by only a few data points. Such rules are not well supported by the training data and will result in bad generalization. Such rules are to be removed and this is said to be less used rules.

3.4 FEATURE SELECTION AND RULE EXTRACTION



Iris setosa **Iris virginica**

Iris versicolor

Fig. 9 Classification of Iris flower

In this work, a feature selection and rule extraction has done on Iris data set of 150 samples which consists of four features and three classes. The four features were measured from each sample such as sepal length, sepal width, petal length and petal width, in centimeters. The three classes of Iris flower are Iris setosa, Iris virginica and Iris versicolor. The combinations of four features are used to distinguish those classes.

The network is processed until the target value is obtained. For instance[8], target value is fixed as 0.25 for Iris setosa, .25 for Iris versicolor, 2.25 for Iris irginica.The Iris data's have been processed in the network to identify the species. The third layer represents the antecedent layer and fourth layer represents the consequent layer. Therefore, the rules are extracted from layer 3 and layer 4. Initially, the antecedent layer has 9 nodes and the consequent layer has 3 nodes. The rules obtained from the network are, 9*3=27 rules.

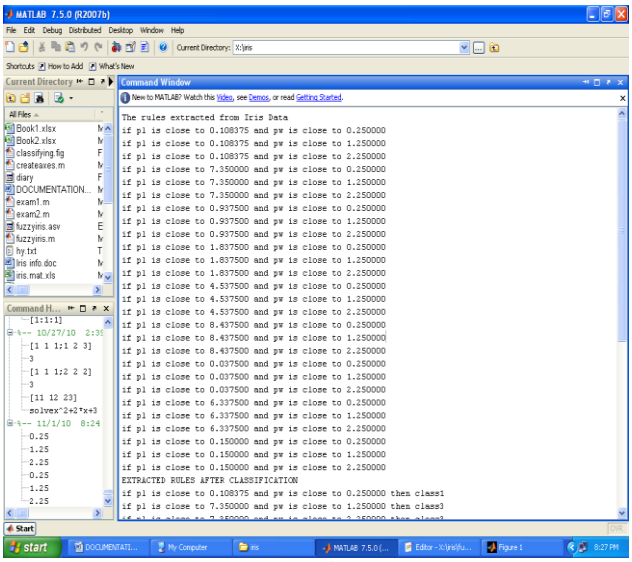


Fig. 10 Rules for Iris Data

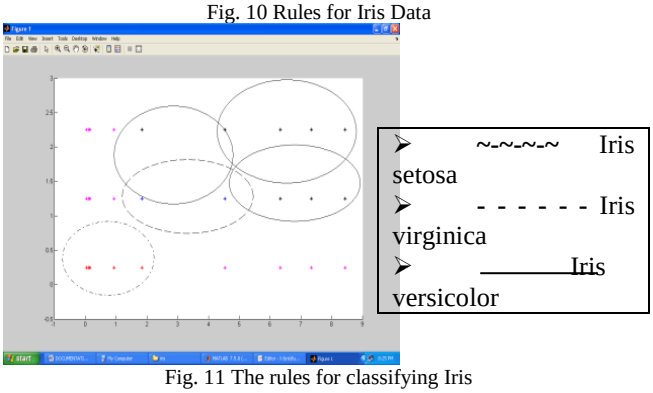


Fig. 11 The rules for classifying Iris

In Figure 3.4, the major axis represents the petal length and the minor axis represents the petal width. The ellipsoidal region represents the three classes and it is distinguished by the line style. After the identification of classes the rules obtained are,If pl is close to 1.5 and pw is close to 0.25 then class 1.If pl is close to 4.5 and pw is close to 1.25 then class 2If pl is close to 6.5 and pw is

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

close to 2.25 then class 3.Finally the classes are identified and the rules are extracted from the Iris dataset.

IV CONCLUSION

The novelty of the system lies in the fact that the network can select good features along with the appropriate rules in an integrated manner. The rules required for the classification task from the network are easily readable. The network starts with all possible rules and the training process retains only the rules required for classification, thus resulting in a smaller architecture of the final network. The final network has a lower running time than the initial network. Neuro fuzzy computing, which is a judicious integration of the merits of neural and fuzzy approaches, enables one to built more intelligent decision making systems. Hence the method is tested on Iris data sets, the network could select good features and extract a small but adequate set of rules for the classification tasks. The results obtained are comparable to the results reported in the literature.

REFERENCE

[1] D.Chakraborthy and N.R.Pal, "Integrated feature analysis and fuzzy rule based system identification in a neuro-fuzzy paradigm", IEEE Trans.Syst. Man Cybern.B., vol.31, pp, 391-400, 2001.

[2] S.Haykin, Neural Networks-"A Comprehensive Foundation", New York: proc.Conc., 1994.

[3] K.Pal, R.Mudi and N.R.Pal, "A new scheme for fuzzy rule based system identification and its application to self-tuning fuzzy controllers", IEEE Trans.Syst.Man Cybern.B., vol.13, pp.859-886, 1998.

[4] N.R.Pal, V.K.Eluri and G.K.Mandal, "Fuzzy Logic approaches to structure preserving dimensionality reduction", IEEE Trans.Fuzzy Syst., vol.10, pp.277-286, 2002.

[5] Ludmila I. Kuncheva, "Fuzzy Classifier Design", Physica-Verlag Heidelberg New York., vol.49, 2000.

[6] John Yen, reza Langari, "Fuzzy logic-Intelligence, Control and Information", Pearson Education, 2003.

[7] J.-S.R.Jang, C.-T.Sun and E.Mizutani, "Neuro-Fuzzy and Soft Computing-A computational Approach to Learning and Machine Intelligence",Prentice-Hall., 1997.

[8] Debrup Chakraborty and Nikhil R.Pal, "A Neuro-Fuzzy Scheme for Simultaneous Feature Selection and Fuzzy Rule-Based Classification", IEEE Transactions on Neural Networks., vol.15, no.1, January 2004.