Evolving Neuro-Fuzzy Rule Generation: Survey in Data Mining of Medical Diagnose Framework

Conference Paper · January 2008

DOI: 10.13140/2.1.1794.5605

Dr. Samaher Al-Janabi
University of Babylon

Evolving Neuro-Fuzzy Rule Generation: Survey in Data Mining of Medical Diagnose Framework

Dr.Abbas Mohsin Al-Bakry, Samaher Hussein Ali

Department of Computer Science, University of Babylon, Iraq

E-mail: (abbasmoh67, Samaher_hussein)@yahoo.com

Abstract: This paper presents a methodology for knowledge discovery in data mining of medical data with the use of hybrid Evolving Fuzzy Neural Networks (EFuNNS). EFuNNs are five layer sparsely connected networks. EFuNNs contain dynamic structures that evolve by growing and pruning of neurons and connections. EFuNNS merge three supervised classification methods: connectionism, fuzzy logic, and case-based reasoning. By merging these strategies, this new structure is capable of learning and generalising from a small sample set of large attribute vectors as well as from large sample sets and small feature vectors. After classification has been made through EFuNNs , one can inspect each class of the patterns acquired. There are several methods of inspections. The easiest one is Statistical Analysis (SA) of each class. Using central tendency and dispertion statistical measures one can form several rules that govern each class attributes. The proposed methodology provides fast and accurate adaptive learning for generated rules from data mining. It is also applicable for classification problem.

Key words: Evolving Fuzzy Neural Networks, Data Mining, Knowledge Discovery, Rule Generation, Medical Diagnose.

INTRODUCTION

Data mining is the process of automatic extraction of useful, novel, and understandable patterns in very large databases. One of the main goals of data mining is association rule mining. The problem of mining association rules consists of two sub problem. The first one is how to find the set of large or frequent item sets. The second sub problem is how to generate and test the interestingness of generation rules [1].

Data Base Management Systems (DBMSs) were initially optimized to process queries that may touch a small part of the database and transactions that deal with insertion or updates of a few tuples per relation to process. The usual activities of databases (insertion, deletions, updates and also supporting information query requirements) are called traditional database application or on-line transaction processing. Knowledge discovery in databases includes all pre-processing steps in stored data, discovering interesting patterens on the data, and post-processing of the result found on the data.

Data mining refers to the discovery of interesting and frequent patterns from the data in knowledge discovery process. These interesting patterns may be in the form of associations, deviations, regularities, etc. association rules are the main patterns in the form of associations that we will focus our attention on through this research.

Data mining algorithms (especially association rules algorithms) are not well-integrated with database management systems because of there are several limitations. Some of them are as follows:

- The first limitation is that the existing manipulate data file in specific format [2].
- The second limitation is that all the miners use loosely-coupled or tightly-coupled approaches for establishing their connection to database. These approaches have many drawback such as slowness of the miners; increase of I/O system traffics, bad integration with DBMSs , and their dependence on cache-mine architecture.
- The available rule miners can only mine rules from one file or table. It is very often required to combine data from more than one data source; therefore, it is important to develop a miner having the ability to connect many files or tables to generate a suitable data set for mining.

The term rule generation encompasses both rule extraction and rule refinement. Note that rule extraction here refers to extracting knowledge from the artificial neural network, using the network parameters in the process. Rule refinement, on the other hand, pertains to extracting refined knowledge from the artificial neural network that was initialized using crude domain knowledge, rules learned and interpolated for fuzzy reasoning and fuzzy control can also be considered under rule generation. It covers, in a wider sense, the extraction of domain knowledge (say, for initial encoding of an artificial neural network) using non connectionist tools like fuzzy sets and rough sets.

Section 2 provides a brief review on knowledge discovery in databases. Section 3 characteristics that from the basis of EFUNNs are identified. Section 4 explains the rule generation phases of EFNNs. Section 5 highlight a case study of evolving neuro-fuzzy rule generation algorithm with application to medical diagnosis. In section 6 discussion and concludes the article.

1. Knowledge Discovery in Database(KDD)

Knowledge discovery differs from traditional information retrieval from databases. In traditional DBMS, database records are returned in response to a query; while in knowledge discovery, what is retrieved is not explicit in the database. Rather, it is implicit patterns. The Process of discovering such patterns is termed data mining. Data mining finds these patterns and relationships using data analysis tools and techniques to build models. There are two main kinds of models in data mining. One is predictive models, which use data with known results to develop a model that can be used to explicitly predict values. Other is descriptive models, which describe patterns in existing data. All the models are abstract representations of reality, and can be guides to understanding business and suggest actions..

KDD[3],[4] is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in the data.

KDD in fact, aims at discovering unexpected, useful and simple patterns, and it is an inter-disciplinary research area. It is interest to researchers in machine learning, pattern recognition, databases, statistics, artificial intelligent, expert systems, graph theory, and data visualization.

KDD process is an interactive and iterative multi-step process which uses six step to extract interesting knowledge according to some specific measures and thresholds. We can describe the steps of knowledge discovery as follows [5]:

- Data selection: once you have formulated your information requirements, the next logical step is to collect and select the data you need. In most cases, this data will be stored in operational databases used in organization.
- Cleaning: when the data have been collected, the next stage is cleaning. A user is probably not aware of the pollution that exists in this data, nor of the potential of data mining in general. It is therefore a good idea to spend some time in the examining the data in order to obtain a feeling for the possibilities, although this may in practice be difficult with large data sets. The best way to avoid polluted data is to organize the input of the data properly. Checking domain consistency needs to be carried out by programs that have deep semantic knowledge of the attributes that are being checked. Most forms of pollution are produced via the method in which the data is gathered in the field.
- Enrichment: once the data is cleaned, it may be required to be enriched . in most countries access to many additional databases is available on a commercial basis, and these can provide information on a variety of subjects, including demographic data such as the average prices of houses and cars. Types of insurance that people have, and so on.
- Coding: is a creative and mining-domain dependant process there can be infinite different codes that are related to any number of different potential patterns we would like to find.
- Data mining: refers to the overall process of discovering new patterns or building models from a given dataset. Typically, data mining has two high level goals of prediction and description. Two main forms of data mining can be identified [6]. In **verification-driven** data mining, the user postulates a hypothesis, and the system tries to validate it. The common verification-driven operations include query and reporting, multidimensional analysis, and statistical analysis. **Discovery-driven** mining on other hand automatically extracts new information. The typical discovery- driven tasks include: association rule, sequential patterns, classification , regression, clustering, and similarity search.
- Reporting: the result of data mining can take many forms. In general, any report write, traditional query tools for databases, or graphical tool can be used to make the results of the process accessible.

While, we can summarized the stages of data mining by three stages as fellows:

- Exploration: this stage usually starts with data preparation which may involve cleaning data, data transformations, selecting subset of records and in case of data sets with large numbers of variables (fields) performing some preliminary feature selection operations to bring the number of variables to a manageable range (depending on the statistical methods which are being considered).

- Model building and validation: this stage involves considering various models and choosing the best one based on their predictive performance(i.e., explaining the variability in question and producing stable results across samples). There are a variety of techniques developed to achieve that goal many of which are based on so called "competitive evaluation of models" that is, applying different models to the same data set and then comparing their performance to choose the best.

- Deployments: that final stage involves using the model selected as best in the previous stage and applying it to new data in order to generate predications or estimates of the expected outcome.

2. Fuzzy Neural Networks (FuNN) and Evolving Fuzzy Neural Networks(EFuNN)

2.1.Fuzzy Neural Networks

Fuzzy neural networks are neural networks that realise a set of fuzzy rules and a fuzzy inference machine in a connectionist way[7]. We shall use this term to cover also all fuzzified connectionist modules [8]. FuNN is a fuzzy neural network introduced in 14) and developed as FuNN/2 in. It is a connectionist feed-forward architecture with five layers of neurons and four layers of connections. The first layer of neurons receives the input information. The second layer calculates the fuzzy membership degrees to which the input values belong to predefined fuzzy membership functions, e.g. small, medium, large. The third layer of neurons represents associations between the input and the output variables, fuzzy rules. The fourth layer calculates the degrees to which output membership functions are matched by the input data, and the fifth layer performs defuzzification and calculates exact values for the output var iables. A FuNN has features of both a neural network and a fuzzy inference machine. The number of neurons in each of the layers can potentially change during operation through growing or shrinking. The number of connections is also modifiable through learning with forgetting, zeroing, pruning and other operations [9].

The membership functions (MF) used in FuNN to represent fuzzy values, are of triangular type, the centers of the triangles being attached as weights to the corresponding connections. The MF can be modified through learning that involves changing the centers and the widths of the triangles.

Several training algorithms have been developed for FuNN [10]and several algorithms for rule extraction from FuNNs have been developed and applied [9]. One of them represents each rule node of a trained FuNN as an IF-THEN fuzzy rule.

FuNNs are universal statistical and knowledge engineering tools. Many applications of FuNNs have been developed and explored; such as pattern classification.

2.2. Evolving Fuzzy Neural Networks

EFuNNs are FuNN structures that but here all nodes in an EFuNN are created during (possibly one-pass) learning. The nodes representing MF (fuzzy label neurons) can be modified during learning. As in FuNN, each input variable is represented here by a group of spatially arranged neurons to represent a fuzzy quantisation of this variable. For example, three neurons can be used to represent "small", "medium", and "large" fuzzy values of the variable. Different membership functions (MF) can be attached to these neurons (e.g. triangular, or Gaussian). New neurons can evolve in this layer if, for a given input vector, the corresponding variable value does not belong to any of the existing MF to a degree greater than a membership threshold. A new fuzzy input neuron, or an input neuron, can be created during the adaptation phase of an EFuNN.

3. Rule Generation Phases of EFuNN

There are different fuzzy, neural and neuro-fuzzy models for rule generation, infencing, and querying, along with their salient features as explained in [11]. But in this paper, we suggest a methodology for generation rule of medical database based on evolving fuzzy neural network (EFuNNs). EFuNNs merge three AI paradigms: connectionism; fuzzy rule based systems; and case base reasoning. Then the results of EFuNNs are treatment by statistical analysis (SA). The methodology phases has been illustrated in Figure 1.

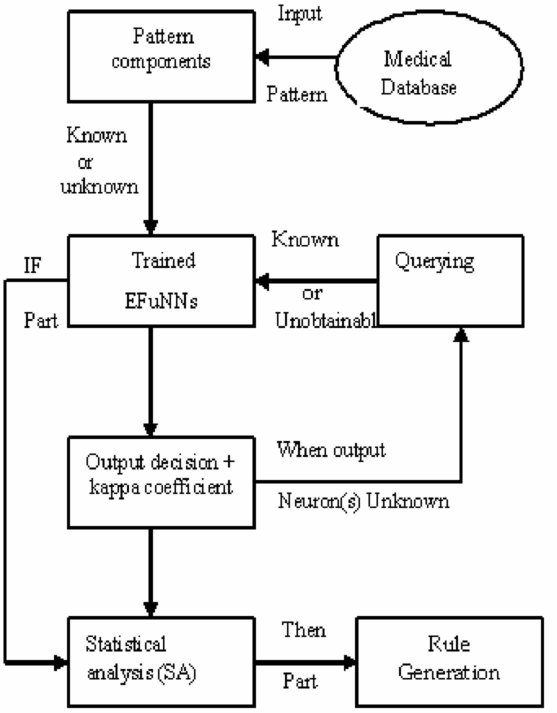

Figure (1): Block diagram of rule generation phases of EFuNNs

The model is capable of

- Inferencing based on complete and/ or partial information
- Querying the user for unknown input variables that are key to reaching a decision
- Producing justification for inferences in the form of TF-THEN rules.

3.1. The EFuNN Learning Algorithm

Here, the EFuNN evolving algorithm is given as a procedure of consecutive steps

step1: Initialise an EFuNN structure with a maximum number of neurons and zero-value connections. Initial connections may be set through inserting fuzzy rules in a FuNN structure. FuNN is an open architecture that allows for insertion of fuzzy rules as an initialisation procedure thus allowing for prior know ledge to be used before training (the rule insertion procedure for FuNNs can be applied). If initially there are no rule (case) nodes connected to the fuzzy input and fuzzy output neurons with non-zero connections, then *connect* the first node rn=1 to represent the first example EX=x 1 and set its input W1 (rn) and output W2 (rn) connection weights as follows:

<Connect a new rule node rn to represent an example EX>:

W1(rn)=EX
W2(rn)=TE
Where TE is the fuzzy output vector for the (fuzzy) example EX

step2: WHILE <there are examples> DO
Enter the current, example xi, EX being the fuzzy input vector (the vector of the degrees to which the input values belong to the input membership functions). If there are new variables that appear in this example and have not been used in previous examples, create new input and/or output nodes with their corresponding membership functions.

step3: Find the normalised fuzzy similarit y between the new example EX (fuzzy input vector) and the already stored patterns in the case nodes *j=1, 2,…, rn*.

$$D(j) = \sum abs(EX - W1(j)) / \sum (W1(j)) + EX) \qquad (1)$$

step4: Find the activation of the rule (case) nodes j, *j=1:rn*. Here radial basis activation function.

$$A1(j) = radbas(D(j)) \qquad (2)$$

step5: Update the local parameters defined for the rule nodes, e.g. age, average activation as predefined.

step6: Find all case (rule) nodes j with an activation value *A1(j)* above a sensitivity threshold (*Sthr).*

step7: If there is no such case node (adaptive the knowledge base), then <*Connect a new rule node*> using the procedure from step 1.
ELSE
step8: Find the rule node *inda1* that has the maximum activation value (*maxa1).*

step9: (a) in case of one-of-n EFuNNs, propagate the activation *maxa1* of the rule node *inda1* to the fuzzy output neurons. Saturated linear functions are used as activation functions of the fuzzy output neurons:

$$A2 = satlin(A1(inda1) * W2) \qquad (3)$$

(b) in case of many-of-n mode, only the activation values of case nodes that are above an activation threshold of *Athr* are propagated to the next neuronal layer.
step10: Find the winning fuzzy output neuron *inda2* and its activation *maxa2.*

step11: Find the desired winning fuzzy output neuron *inda2* and its value *maxt2.*

step12: Calculate the fuzzy output error vector:

$$Err = A2 - TE \qquad (4)$$

step13: IF (inda2 is different from inda2) or (abs(Err (inda2)) > Errthr) <Connect a new rule node>
ELSE

step14: Update: (a) the input, and (b) the output connections of rule node k=inda1 as follows:

(a) Dist=EX-W1(k);
W1(k)=W1(k) + lr1*Dist,
where lr1 is the learning rate for the first layer;

(b) W2(k) = W2 (k) + lr2* Err.*maxa1,
where lr2 is the learning rate for the second layer.

step15: Prune rule nodes j and their connections that satisfy the following fuzzy pruning rule to a predefined level representing the current need of pruning:
IF (node (j) is OLD) and (average activation A1av(j) is LOW) and (the density of the neighbouring area of neurons is HIGH or MODERATE) and (the sum of the incoming or outgoing connection weights is LOW) and (the neuron is NOT associated with the corresponding "yes" class output nodes (for classif ication tasks only)) THEN the probability of pruning node (j) is HIGH

The above pruning rule is fuzzy and it requires that all fuzzy concepts such as OLD, HIGH, etc., are defined in advance. As a partial case, a fixed value can be used e.g. a node is old if it has existed during the evolving of a FuNN from more than 1000 examples.

step16: END of the while loop and the algorithm

step17: Repeat steps 2-16 for a second presentation of the same input data or for training if needed.

EFuNNs are very efficient when the problem space is not adequately represented by the available training data. In these cases, where online learning is required, estimates of the acceptance diagnoses in the problem space must be adaptable in time.

3.2. Certainty Measures

Generation rules were determined by the kappa coefficient, $\hat{k}$. The coefficient of agreement called (kappa) measures the relationship of beyond chance agreement to expected disagreement. It uses all the cells in the confusion matrix, not just the diagonal elements. The estimate of kappa(K) is the proportion of agreement after chance agreement is removed from consideration. The kappa value for class i(k_i) is defined as

$$K_i = \frac{s.s_{ic} - s_i . s_i^{-}}{s.s_i^{-} - s_i . s_i^{-}} \qquad (5)$$

Let S be an l*l matrix whose (i,j)th element s_{ij} indicate the number of patterns actually belonging to class i, but classified as class j, where s_i^{-} is equal to the number of points in class i and s_{ic} of these points are correctly classified.

The numerator and denominator of overall kappa are obtained by summing the respective numerators and denominators of k_i separately over all classes.

In this paper, we deal with this measure. But actually there are some other measures you can used in this field such as accuracy, user's accuracy, fidelity, confusion, cover, rule base size, computational complexity and certainty [12].

3.3. Statistical Analysis

After classification has been made thought EFuNN and determined the desired output with the kappa coefficient. We can inspect each class of patterns acquired. There is several methods of inspections. The easiest one is statistical analysis of each class. Using central tendency and desperation statistical measures we can form several rule that govern each class attributes [13],[14].

3.3.1. Measures of central tendency:

Measures of central tendency are measures which are representative of a sample. They enable one to be more objective when drawing conclusions or making inferences. These measures identify the center or middle of a set of values and best characterize the distribution. The typical measures of central tendency are: mode, median and mean. In this paper, we deal with mean that represented the arithmetic average of values.

3.3.2. Measures of dispersion:

Another important characteristic of a data set is how it is distributed, or how far each element is from some measure of central tendency (average). There are several ways to measure the variability of the data. Although the most common and most important is the stander deviation. Which provides an average distance for each element from the mean; several others are also important such as range and variance. In this paper, we deal with a standard deviation is the most and useful measure because it is the average distance of each score from the mean. The formula for stander deviation is as follows:

$$\sigma = \sqrt{\frac{\sum(\mu - x_i)^2}{S}} \qquad (6)$$

3.4. Rule Generation

After calculate these stastical measures for each class, we can form one production rule depending on one or more measures.

The input is in quantitative, linguistic, or set forms or a combination of theses. It is represented as a combination of membership values to the three primary linguistic properties low, middle, high.

The user can ask the system why it inferred a particular conclusion. The system answers with an IF-THEN Rule applicable to the case at hand. Note that these IF-THEN Rules are not represented explicitly in the knowledge base; they are generated by the inferencing system, by backtracking, from the connection weights as needed for explanation.

The complete IF part of the rule is obtained by ANDING clauses corresponding to each of the features, e.g.,

IF F_1 is more or less high and F_2 is not high and ……. and F_n is very high

The consequent part of the corresponding IF-THEN rule is generated using the statistical measures. A simple rule, in terms of features F_1 and F_2, is as follows:

IF F_1 is very low AND F_2 is high THEN likely class1.

4. Application and Result

Medical diagnosis or more specifically, the results of tests involve imprecision, noise and individual difference. Often one cannot clearly distinguish the difference between normal and pathological values.

This data hepato consists of 536 patterns (patient cases) of various hepatobiliary disorders. The nine input features are the results of different biochemical tests: Glutamic Oxalacetic Transaminate (GOT, Karmen unit), Glutamic Pyruvic Transaminase (GPT, Karmen Unit), Lactate Dehydrase (LDH, iu/l), Gamma Glutamyl Transpeptidase (GGT, mu/ml), Blood Urea Nitrogen (BUN, mg/dl), Mean Corpuscular Volume of red blood cell (MCV, fl), Mean Corpuscular Haemoglobin (MCH, pg), Total Bilirubin (TBil, mg/dl) and Creatinine (CRTNN, mg/dl). The 10 th feature corresponds to the sex of the patient and is represented in binary mode as(0,1). The Hepatobiliary disorder Alcoholic Liver Damage (ALD), Primary Hepatoma (PH), Liver Cirrhosis (LC) and Cholelithiasis (C) constitute the four output classes[15],[16].

The rule generation by application the rule generation phases of EFuNN to medical diagnoses data is depicts below, where the number of rules equal seven and the value of kappa=73.05.

IF (Mean-low-LDH=333.15 OR Standard deviation -low-LDH=140.00) AND (Mean-middle-MCV=17.27 OR Standard deviation -middle-MCV=7.18) AND (NOT (Mean-low-GOT=52.47 OR Standard deviation -low-GOT= 62.44) THEN Class Alcoholic Liver Damage, Cf=0.857.

IF (Mean-low-LDH=336.15 OR Standard deviation -low-LDH=140.00) AND (Mean-low-MCV=12.15 OR Standard deviation -low-MCV=5.11) AND (Mean-low-GOT=52.47 OR Standard deviation -low-GOT= 62.44) THEN Class Alcoholic Liver Damage, Cf=0.800.

IF (Mean-low-GPT=24.04 OR Standard deviation -low-GPT=29.15) AND (Mean-middle-GPT=53.41 OR Standard deviation -middle-GPT=55.14) AND (Mean-middle-LDH=477.95 OR Standard deviation -middle-LDH= 254.44) THEN Class Primary Hepatoma, Cf=0.846.

IF (Mean-middle-MCV=17.27 OR Standard deviation -middle-MCV=7.18) THEN Class Primary Hepatoma, Cf=0.571.

IF (Mean-low-LDH=336.15 OR Standard deviation -low-LDH=140.00) AND (Mean-low-GPT=24.04 OR Standard deviation -low-GPT=29.15) AND (Mean-middle-LDH=477.95 OR Standard deviation -middle-LDH= 254.44) THEN Class Liver Cirrhosis, Cf=0.800.

IF (Mean-low-LDH=336.15 OR Standard deviation -low-LDH=140.00) AND (Mean-low-MCV=12.15 OR Standard deviation -low-MCV=5.11) AND (NOT(Mean-low-GOT=52.47 OR Standard deviation -low-GOT= 62.44) THEN Class Cholelithiasis, Cf=0.833.

IF (Mean-middle-GPT=53.41 OR Standard deviation -middle-GPT=55.14) AND (Mean-low-GPT=24.04 OR Standard deviation -low-GPT=29.15) AND (NOT(Mean-middle-LDH = 477.95 OR Standard deviation -middle-LDH= 254.44) THEN Class Cholelithiasis, Cf=0.833.

5. Conclusion

The paper suggests a methodology for generation rule of medical database based on evolving fuzzy neural networks (EFuNNs). EFuNNs merge three AI paradigms: connectionism; fuzzy rule-based systems; and case-based reasoning. The methodology has been illustrated on hepatobiliary disorders data. The methodology proves to be effective in terms of fast adaptive learning for rule generation.

EFuNNs have features that make them suitable for data mining enviroment when large feature space, and/or large databases are utilised. These features are [17]:

- Local element training and local optimization.
- Fast learning (possibly one pass).
- Achieving high local or global generalisation in an on-line learning mode.
- Memorizing exemplars for a further retrieval or system's improvement.
- Interpretation of the EFuNN structure as a set of fuzzy rules.
- Dynamical self-organisation achieved through growing and pruning.

The strength of EFuNNs centres(Mean) around tuning the sensitivity threshold. The sensitivity threshold acts as a minimum resolvable distance or radius (standard deviation) *r* around the rule nodes.

As the sensitivity increases, the boun dary surface separating two acceptance rules in the feature space reduces. If an example from Class 1 is defined as x and Class 2 as x', then d = x – x' < r for discrimination.

One obvious outcome of this research is the development of a dynamic architecture that can tune the sensitivity threshold without operator intervention. Lower sensitivity thresholds promote stability, generalisation, and computationally efficiency, while higher sensitivities increase discrimination.

REFERENCES

[1] Hussein K. AL-Kafajy "knowledge discovery in database by using data mining" P.DH.Thesis Dep. Of computer science & information systems, University of Technology, 2002.

[2] IMB, http//www.software.ibm.com/data/miner,2000.

Journal of Research and Development, 2, 1958.

[3]Usama F, Gregory P & Padhraic S," From data mining to knowledge discovery:An overview", Usama F, Gregory P, Padhraic S and Ramaswamy U., editors,Advances in Knowledge Discovery and Data Mining, p1-31,AAAI/MIT PRESS, 1996.

[4]Usama F, Gregory P & Padhraic S," *knowledge discovery* and Data mining: Towards a unifying framework", in processings of 2nd intl. conf. on *knowledge discovery and data mining*,P 82-88, Portland, Oregon, USA, August 1996.

[5]Mannila H., "Methodes and Proplems in data mining", in proc. Of 6th conf.on database theory, p 41-55, 1997.

[6]Michael G & Le G., "A survery of data mining and KDD Tools", Journal of UCS, 6(1), P 84-107, 2002.

[7]K.S. Ray and J. Ghoshal, " Neuro Fuzzy Approach to Pattern Recognition," Neural Networks, 10-1,161-182, 1997.

[8]M.M. Gupta and D.H. Rao, "On the Principles of Fuzzy Neural Networks," Fuzzy Sets and Systems, 61-1, 1-18, 1994.

[9]N. Kasabov, J. S. Kim, M. Watts and A. Gray, "FuNN/2- A Fuzzy Neural Network Architecture for Adaptive Learning and Knowledge Acquisition," Information Sciences – Applications, 101(3-4), 155-175, 1997.

[10]N. Kasabov, "Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid systems," Fuzzy Sets and Systems 82-2, 2-20, 1996.

[11] Sushmita M. and Yoichi H., "Neuro-Fuzzy Rule Generation: Survery in Soft Computing Framework",IEEE Transactions on Neural Networks, Vol 11, NO.3, May 2000.

[12] T.M. Lillesand and R.W. Kiefer, "Remote sensing and image interpretation," (3 rd ed), Thomas Wiley and Sons, New York, 1994.

[13] Freitas.A. Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer, 2002.

[14] Leena.L and Jorma.A,"Rock Image Classification Using Non-Homogenous Textures and Spectral Imaging",Tampere University of Technology, Finland, 2003.

[15] S. Mitra, ``Fuzzy MLP based expert system for medical diagnosis", Fuzzy Sets and Systems, Vol. 65, pp. 285-296, 1994.

[16] Y. Hayashi, ``Neural expert system using fuzzy teaching input and its application to medical diagnosis", Information Sciences Applications, Vol. 1, pp. 47-58, 1994.

[17] N. Kasabov, "Rule extraction, rule insertion, and reasoning in Evolving Fuzzy Neural Networks,"2000.