© 2018 IJSRSET | Volume 4 | Issue 7 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology

Neuro Fuzzy Inference Approach : A Survey

Tigilu Mitiku, Mukhdeep Singh Manshahia*
Department of Mathematics, Punjabi University Patiala, Punjab, India
Corresponding author email : mukhdeep@gmail.com*

ABSTRACT

Fuzzy Logic is an extension of classical logic which provides an effective mathematical tool to represent information in a way that resembles natural human reasoning and deals with system uncertainty and vagueness. ANN is a biologically inspired computational structure comprised of densely interconnected adaptive simple processing elements that are capable of performing massively parallel computations for data processing and knowledge representation. The combination of fuzzy inference system and artificial neural network have attracted the researchers and scholars in various scientific and engineering areas to the growing need of adaptive intelligent systems. Artificial neural network are not good at explaining how they reach their decisions whereas fuzzy systems, which can reason with imprecise information, are good at explaining their decisions but they cannot automatically acquire the rules they use to make those decisions. Due to these limitations an intelligent hybrid systems where two or more techniques are combined in a manner that overcomes the problems of individual techniques are created. Any type of systems that combine these two techniques can be called Neuro-Fuzzy systems. Neuro-Fuzzy systems are systems which utilize fuzzy logic to construct a complex model by extending the capabilities of Artificial Neural Networks. This type of system is characterized by a fuzzy system where fuzzy sets and fuzzy rules are adjusted using input output patterns. There are several types of neuro-fuzzy systems where each author defined its own model. This survey paper describes the most known hybrid neuro-fuzzy techniques, with their advantages and Limitations.
Keywords: Fuzzy Logic, Neural Networks, Neuro Fuzzy Systems

I. INTRODUCTION

Soft computing is a collection of artificial intelligence (AI) methodologies aiming to exploit the tolerance for imprecision and uncertainty that is inherent in human thinking and in real life problems, to deliver robust, efficient and optimal solutions and to further explore and capture the available design knowledge [1]. Soft computing utilizes computation, reasoning and inference to reduce computational cost by exploiting tolerance for imprecision, uncertainty, partial truth and approximation. In contrast to traditional computing (hard computing) that require exact mathematical model and lot of computation time, soft computing deals with approximate models and gives solutions to complex real-life problems. For such problems, methods which are computationally intelligent that possess human like expertise and adapt to the changing environment can be used effectively and efficiently. AI methodologies are accomplished by studying how human brain thinks , learn, decide and work while trying to solve a problem, and then using the outcomes of this study as a basis of developing intelligent software and systems. The core AI methodologies comprising of soft computing are: Fuzzy Logic (FL), Neural Computing (NC), Evolutionary Computing (EC), Probabilistic Computing (PC), Chaotic Computing (CC), rough set theory and Machine Learning (ML). Where PC and FL systems are based on knowledge-driven reasoning, whereas, NC and EC are data-driven search and optimization approaches. The basic tenet of soft computing is that, better results can be obtained through the use of constituent methodologies of soft computing in combination rather than in a stand-alone mode. A combination which has attained wide visibility and importance is that of neuro-fuzzy systems.

1.1 Fuzzy Logic

Most often fuzzy ideas are utilized in our routine life that nobody even pays attention to them. Fuzzy Logic is an extension of classical (conventional) logic which provides an effective mathematical method to represent information in a way that resembles natural human reasoning and deals with system uncertainty and vagueness. Uncertainty can be caused by imprecision in measurement due to imprecision of tools or other factors. It can also be caused by vagueness in language objects and situations. The idea of fuzzy set was introduced by Lofti Zadeh where the behavior of the system is described by fuzzy rules [2]. The behavior of such systems is described through a set of fuzzy rules, like: *if <premise> then <consequent>* that uses linguistics variables with symbolic terms. The fuzzy logic can be described simply as computing with words rather than numbers. Each term represents a fuzzy set. For instance, we use a linguistic variable like short, tall, very tall for HEIGHT or may be young, old, and very old for AGE. A fuzzy logic system is an expert system that uses a collection of fuzzy membership functions and fuzzy IF-THEN rule base to reason about data. The rules in a fuzzy logic system are of a form as following:

IF (x is LOW) AND (y is HIGH) THEN (z is MEDIUM),

where x and y are input variables for known data values, z is an output variable for an output data to be computed, LOW is a membership function (fuzzy subset) defined on the set of x, HIGH is a membership function defined on the set of y, and MEDIUM is a membership function defined on the set of z. The antecedent (the rules premise, between IF and THEN) describes to what degree the rule applies, while the consequent (the rules conclusion, following THEN) assigns a membership function to each of one or more output variables. The set of rules in a fuzzy logic system is known as the rule base or knowledge base.

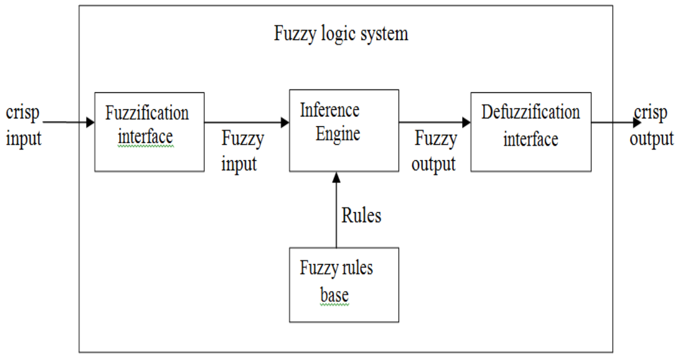

Fig.1 Architecture of a fuzzy logic system [3]

To implement fuzzy logic technique requires three steps as shown in figure1 above. In the first steps, the values of the numerical inputs are mapped by a function according to a degree of compatibility of the respective fuzzy sets; this operation can be called fuzzification. Fuzzification is always necessary in a fuzzy logic system as the input values from existing sensors are always crisp numerical values. The second step is fuzzy inference process includes rule base that contain a number of fuzzy IF- THEN rules and data base (fuzzy inference engine) which defines the membership functions of the fuzzy sets used in the fuzzy rules. The inference engine takes the fuzzy input and the fuzzy rule base to processes the rules in accordance with the firing strengths of the inputs to generates fuzzy outputs. The last processing element of a fuzzy logic system is defuzzification which transform the resultant fuzzy values again into numerical values (crisp output). It maps the fuzzy sets (the aggregated output fuzzy set) produced by the inference engine into crisp numbers using different methods to calculate each associated output. The choice of defuzzification methods usually depends on the application and the available processing power.

Fuzzy systems have the advantage that the fuzzy rules, which store the information, are easily interpretable. Furthermore they provide a simple interface for extending the system with new information (by adding new rules) or manipulating the existing rules. The problem with fuzzy systems lies in the fact that they completely depend on the experts who design them. It only uses the information which was encoded in the system and cannot learn on its own and also incapable of generalization. The described nature of fuzzy systems indicates that a fusion with ANNs may possibly lead to a new powerful computational model. Fuzzy systems are being applied in a wide range of industrial and scientific applications with the main application areas being fuzzy control, data analysis and knowledge based systems. For instance, fuzzy controllers model the control strategy of a human expert to control a system for which no mathematical or physical model exists. They employ a set of linguistic rules to describe the human behavior. The linguistic rules describe a control surface, which defines an appropriate output value for every vector of input values.

1.2 Artificial Neural Network

Artificial neural networks (ANN) or neural computing is one of the rapidly growing fields of research, attracting researchers from a wide variety of engineering disciplines, such as electronic engineering, control engineering, and software engineering [4]. ANN is a biologically inspired computational structure comprised of densely interconnected adaptive simple processing elements that are capable of performing massively parallel computations for data processing and knowledge representation. Neural networks aim to bring the traditional computers a little closer to the way human brain works. ANNS performs best when the relationship between the inputs and outputs are highly non-linear and highly suitable for solving problems where there are no algorithms or specific set of rules to be followed in order to solve the problem. ANN consist of a group of simple processing elements, units or nodes called "neurons" that are functionally analogous to nerve neurons. Each neuron is connected to each other with the weights. The processing ability of the network is stored in the internal unit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns. The information relevant to the input–output mapping of the net is stored in the weights. ANNs while implemented on computers are not programmed to perform specific tasks. Instead, they are trained with respect to data sets until they learn the patterns presented to them. Once they are trained, new patterns may be presented to them for prediction or classification.

1.2.1 Architectures of Artificial Neural Networks

The architecture of an ANN defines how its several neurons are arranged (placed) in relation to each other. ANN contains three types of layers i.e., input layers, hidden layers and output layers. The input neurons receive the data (information), signals, features, or measurement either from input files or directly from electronic sensors in real-time applications. These inputs are usually normalized within the limit values produced by activation functions. This normalization results in better numerical precision for the mathematical operations performed by the network. The output layer neurons send information directly to the outside world, to a secondary computer process, or to other devices such as mechanical control system. Between these two layers there may be many hidden layers. These internal layers contain many of the neurons in various interconnected structures. Hidden layer receives the signals from all of the neuron in a layer before it, which is an input layer. After a neuron performs its function it passes its output to all of the neurons in the layer after it. Once a network has been structured for a particular application, that network is ready to be trained. Considering the arrangement of neuron as well as how they are interconnected and how its layers are composed, the architectures of ANNs can be divided into three. Single layer feedforward network is a network that has just one input layer and a single output layer. The information always flows in a single direction (thus, unidirectional), which is from the input layer to the output layer. Multi-layer feedforward network is the second types of architecture having one or more hidden layers, whose computation nodes are called hidden neurons or hidden units. The number of hidden layers and their respective amount of neurons depend on the nature and complexity of the problem being mapped by the network, as well as the quantity and quality of the available data about the problem. Lastly, recurrent network is a feed forward neural network having one or more hidden layers with at least one feedback loop. The feedback may be a self feedback, i.e., where output of neuron is fed back to its own input. The advantage of ANN with respect to other models are its speed, simplicity, ability of modeling a multivariable problem to solve complex relationships between the variables and can extract the nonlinear relationships among these variables by means of training data. These systems are able to adapt-to learn how to deal with situations that they have not previously encountered and, in extreme cases, are able to learn to survive when the environment in which they operate changes. ANN is widely accepted as a technology offering an alternative way to tackle complex problems in actual situations. The disadvantages includes traditional neural networks often described as being like a "black box," in the sense that once it is trained, it is very hard to see why it gives a particular response to a set of inputs.

1.2.2 Training of Artificial Neural Networks

One of the most relevant features of ANN is their ability of learning from the presentation of samples (patterns), which expresses behavior of the system. All neurons from one layer are connected to all neurons in the next layer. Once a network has been structured for a particular application, it is ready to be trained. To start this process, the initial weights are chosen randomly and then the training or learning begins. During the training process of ANNs, each complete presentation of all the samples belonging to the training set, in order to adjust the synaptic weights and thresholds, will be called training epoch. There are three basic approaches to training:

a) Supervised training: In this type of training, both the inputs and the outputs data are provided to the network; in other words, each training sample is composed of the input signals and their corresponding outputs. The network then processes the inputs and compares its resulting outputs against the desired outputs. Errors propagated back through the system and cause the system to adjust the weights, which control the network. This process occurs over and over as the weights are continually adjusted. This training is considered complete when the neural network reaches a user defined performance level. When no further training is necessary, the weights are frozen for the underlying application. The set of data, that enables the training, is called the "training set." During the training of a network, the same set of data is processed many times, as the connection weights are ever refined.

b) Unsupervised or Adaptive Training: In this type of learning, the network is provided with inputs but not with the desired outputs. The system itself must then decide what features it will use to group the input data. This is often referred to as self-organization or adaption. The networks use no external influences to adjust their weights. Instead, they internally monitor their performance. These networks look for regularities or trends in the input signals, and makes adaptations according to the function of the network. Even without being told whether it's right or wrong, the network still must have some information about how to organize itself. This information is built into the network topology and learning rules. An unsupervised learning algorithm might emphasize cooperation among clusters of processing elements. When some external input activated any node in the cluster, the cluster's activity as a whole could be increased. Likewise, if external input to nodes in the cluster was decreased, that could have an inhibitory effect on the entire cluster.

C) Reinforcement Learning: Methods based on reinforcement learning are considered a variation of supervised learning techniques, as they continuously analyze the difference between the response produced by the network and the corresponding desired output [5]. The reinforcement learning algorithms adjusts the internal neural parameters depending on any qualitative or quantitative information acquired through the interaction with the system (environment) being mapped and use this information to evaluate the learning performance. The network learning process is usually done by trial and error because the only available response for a given input is whether it was satisfactory or unsatisfactory. If satisfactory, the synaptic weights and thresholds are gradually incremented to reinforce (reward) this behavioral condition involved with the system. Several learning algorithms used by reinforcement learning are based on stochastic methods that probabilistically select the adjustment actions, considering a finite set of possible solutions that can be rewarded if they have chances of generating satisfactory results. During the training process, the probabilities associated with action adjustment are modified to enhance the network performance.

1.3 Neuro Fuzzy Systems

The design of control systems is currently carried by a large number of requirements posed by increasing competition, environmental requirements, energy and material costs and the demand for robust, fault-tolerant systems. These considerations introduce extra needs for effective process modeling techniques. Many real life systems are not suitable for conventional modeling approaches due to the lack of precise, formal knowledge about the system, due to strongly nonlinear behavior, high degree of uncertainty or time-varying characteristics. In recent years, fuzzy logic control has played an increasing and significant role in the development and design of real-time control applications. In FIS, the relationships between variables are represented by means of IF–THEN rules with imprecise (ambiguous) predicates. The common model types applied for fuzzy systems are Mamdani, Takagi and Sugeno and Tsukamoto model. They differ in the consequents of their fuzzy rules, and thus their aggregation and defuzzification procedures also differ accordingly. In fuzzy system, the determination of membership function type, number of rules of fuzzy controller and selection of parameters of fuzzy controller are made by means of trial and error method and by using the specialization (expert) knowledge. As there are no formal methods to determine its parameters (fuzzy sets and fuzzy rules), the implementation of a fuzzy system can be very time consuming. Therefore, it is necessary to combine fuzzy system with algorithms which can learn fuzzy systems automatically from data. A combination of these two technologies endows systems with a twofold advantage.

A neuro-fuzzy system is a neural network that is functionally equivalent to a fuzzy inference model trained by neural network learning algorithm. In a neuro-fuzzy system, neural networks extract fuzzy rules automatically from numerical data and, the membership functions are adjusted adaptively through the learning process. Training helps the system to develop fuzzy IF-THEN rules and determine membership functions for input and output variables of the system. Three types of combinations between neural networks and fuzzy systems can be distinguished that have the goal of tuning or learning a fuzzy system. Cooperative neuro-fuzzy system is system where neural networks used only in an initial phase to determine sub-blocks of fuzzy system i.e., fuzzy sets and /or fuzzy rules using training data. Then neural networks are removed and only the fuzzy system is executed. In concurrent neuro- fuzzy system neural network helps the fuzzy system continuously (or vice versa) to determine the required parameters, especially when the input variables of the controller cannot be measured directly. Hybrid neuro fuzzy systems are systems for which more than one technology is employed to solve the problem. The hybridization defines a homogeneous architecture, usually similar to the structure of a neural network. Both cooperative and concurrent neuro-fuzzy systems do not optimize the fuzzy system but only aids to improve the performance of the overall system. The hybrid neuro fuzzy system combines the learning, parallel computation and adapting abilities of neural networks with the human-like knowledge representation and explanation abilities of fuzzy logic system to build intelligent system. The main aim of the hybridization has been to overcome the weaknesses in one technology during its application, with the strengths of the other by appropriately integrating them. Advantages of hybrid neuro-fuzzy system are fast and accurate learning ability, good generalization capabilities, excellent explanation facilities in the form of meaningful *fuzzy* rules, and the ability to accommodate both data and existing expert knowledge about the problem. The majority of the researchers use the neuro-fuzzy term to refer only hybrid neuro-fuzzy system. The need of using hybrid neuro-fuzzy systems is growing rapidly with successful applications in many areas including process control, engineering design, financial trading, credit card evaluation, medical diagnosis, and cognitive simulation etc. There are several different types of hybrid neuro-fuzzy systems developed in literature. Nine types of hybrid neuro fuzzy system were given by the authors Jose and their colleagues namely: Fuzzy Adaptive Learning Control Network (FALCON), Adaptive Network based Fuzzy Inference System (ANFIS), Generalized Approximate Reasoning based Intelligence Control (GARIC), Neural Fuzzy Controller (NEFCON), Fuzzy Inference and Neural Network in Fuzzy Inference Software (FINEST), Fuzzy Net (FUN), Self Constructing Neural Fuzzy Inference Network (SONFIN), Dynamic/Evolving Fuzzy Neural Network (EFuNN and dmEFuNN) [6]. These differ from each other with regard to the shape of the consequent part of the fuzzy rules, the type of inference rules and the shape of the membership functions.

A. Adaptive Neuro Fuzzy Inference System

FIS is based on expertise expressed in terms of fuzzy "if-then" rules and can be employed to represent the human reasoning process, and to make decisions based on uncertain and imprecise environments. FIS is the core of ANFIS. ANFIS combines the self-learning ability of NN with the linguistic expression function of fuzzy inference system. They possess human-like expertise within a specific domain; they adapt themselves and learn to do better in changing environments. In ANFIS, neural networks recognize patterns and help the adaptation of environments. The ANFIS architecture is composed of five layers as shown in figure 2 below and implements Takagi-Sugeno fuzzy inference system [7]. The first layer is responsible for the mapping of the input variable relative to each membership functions. The T-norm operator is applied in the second layer to calculate the antecedents of the rules which is called firing strengths of the rules. The third layer normalizes the rules strengths obtained from the previous layer and output of this layer is called normalized firing strength followed by the fourth layer where the consequents of the rules are determined. The output layer calculates the final output as the summation of all the signals that arrive to this layer. ANFIS uses either back-propagation learning algorithm alone to determine the membership functions parameters or in combination with a least squares method. Some of the advantages of ANFIS are fast convergence due to hybrid learning, computationally efficient, refines fuzzy if-then rules to describe the behavior of a complex system and ability to adjust the shape of input membership functions. It has better tracking and adaptive capabilities than any other controller. Most of the time, the ANFIS controller mimicked another working controller, the controller being mimicked is an experienced human operator who can control the plant satisfactorily.

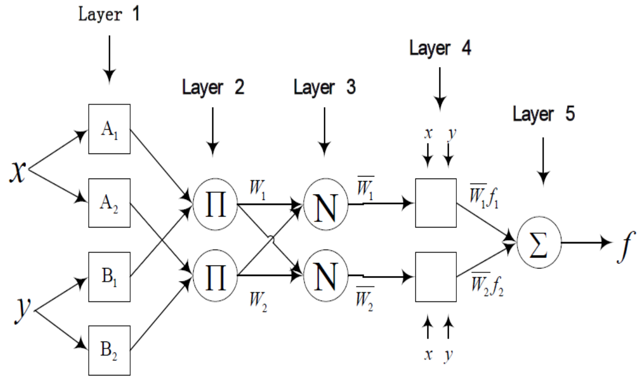

Figure 2. ANFIS architecture [8]

B. Generalized Approximate Reasoning based Intelligent Control

Generalized Approximate Reasoning based Intelligent Control (GARIC) architecture is an extended version of Berenji's Approximate Reasoning based Intelligent Control(ARIC) that implements a neuro-fuzzy controller using two neural network modules, ASN (Action Selection Network) and AEN (Action State Evaluation Network) [9]. AEN is an adaptive critic that evaluates the ASN and provides advice to the main controller. ASN don't use any weighted connections, but the learning process modifies parameters stored within the units of the network. ASN of GARIC is feed forward network with five layers which is responsible for selecting an action based on the current state of the system using fuzzy inference system. The first layer is the input layer consisting of real-valued input variables. Inputs are passed on to the second layer. The second layer represents the fuzzy rules nodes, which determine the degree of fulfillment of a rule using a softmin operation. The third layer represents the linguistic values of the control output variable. Each node determines the degree of fulfillment (or firing strength) of a rule. The conclusions of each rule are calculated depending on the strength of the rules antecedent calculated in the rule nodes. Nodes of the forth layer correspond to consequent labels. The fifth layer's nodes calculate the real output values based on the rules' firing strength and the forth layer's outputs. GARIC uses a mixture of gradient descending and reinforcement learning to adjust its node parameters. The hybrid learning stops if the output of the AEN ceases to change.

C. Fuzzy Adaptive Learning Control Network

Fuzzy Adaptive Learning Control Network (FALCON) model is a connectionist system with five layers and implements a mamdani type FIS [10]. Input nodes are located in the first layer. The first hidden layer is responsible for the fuzzification of each input variable. Each node in this layer can be a single node representing a simple membership function (MF) or composed of multilayer nodes that compute a complex MF. The Second hidden layer defines the membership functions for the preconditions (antecedents) of the rule followed by the consequents in the third hidden layer. Each node of third hidden layer acts as a fuzzy rule. Finally the fifth layer is the output layer; here for every output there are two nodes: one is for training data which is the desired output and the other is for decision signal which is the actual output of FALCON. FALCON uses a hybrid learning algorithm comprising of unsupervised learning to define the initial membership functions and rule base parameters and it uses a learning algorithm based on the gradient descent to optimize/adjust the final parameters of the membership functions to produce the desired output. Training is done by a two-phase-algorithm. The first phase is responsible for finding the initial membership functions by a self-organized learning scheme. In the second phase the parameters of the membership functions are adjusted using supervised learning. During the training nodes, links connecting the network can be deleted or combined reforming the structure of the network.

D. Neural Fuzzy Controller

Neural Fuzzy Controller (NEFCON) architecture has three layers and is designed to implement Mamdani type FIS [11]. The first layer consists of the input nodes which represent input variables. The nodes in the second layer represent the fuzzy rules and the third layer holds the output nodes, which is responsible for the defuzzification interface. In contrast to neural networks, the links connecting the nodes in NEFCON are weighted with fuzzy sets instead of real numbers. The process of learning in NEFCON architecture is based on a mixture of reinforcement learning with backpropagation algorithm and it is carried out in two stages: learning the structure (i.e., learning the rules) and learning the parameters (i.e., learning the MFs). When learning the parameters, it is assumed that the structure is already known. Other two systems were developed based on NEFCON which are specialized versions of the original architecture. NEFCLASS [12] is specialized in classification problems and NEFPROX [11] which was created for function approximation.

E. Fuzzy Neural Networks

Fuzzy Neural Networks (FuNNs) are neural networks that acquire a set of fuzzy rules and fuzzy inference machine in a connectionist way [13,14,15,16,2]. It is feed-forward network architecture with five layers of neurons and four layers of connections. The input layer receives the input information and transfer to the second layer. The first hidden layer determines the membership degrees to which the input values belong to predefined fuzzy membership functions. Neurons in the second hidden layer represent associations between the input and the output variables, fuzzy rules. The degree to which output membership function matched by the input data are calculated by the third hidden layer. The output neuron performs defuzzification and calculates exact values for the output variables. A FuNN has features of both a neural network and a fuzzy inference machine. The number of neurons in 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 [15,17]. The MF used in the structure 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.

Different algorithms for training, rule insertion, rule extraction and adaptation have been developed for FuNN [15,17]. FuNNs have several advantages when compared with the traditional connectionist systems, or with the traditional fuzzy systems: i) they are statistical and knowledge engineering tools; ii) they are relatively robust to catastrophic forgetting, i.e. when they are further trained on new data, they keep a reasonable memory of the old data; iii) they interpolate and extrapolate well in regions where data is sparse; iv) they accept both real input data and fuzzy input data represented rule nodes input outputs as singletons (centers of the input membership functions). The above listed features of FuNNs make them universal statistical and knowledge engineering tools.

F. Evolving Fuzzy Neural Networks

Evolving Fuzzy Neural Networks (EFuNNs) are FuNN structures having five layered architectures that evolve according to the ECOS (Evolving COnnectionist Systems) principles for adapting intelligent systems formed because of evolution and incremental, hybrid (supervised/unsupervised), online learning [18,19]. They can accommodate new input data, including new features, new classes, and etc. through local element tuning. ECOS are systems that evolve in time through interaction with the environment, *i.e.*, an ECOS adjust its structure with a reference to the environment. The input layer passes input variable/data to the second layer that represents fuzzy quantification of each input variable space. Any input variable is represented by a group of spatially arranged neurons to represent a fuzzy quantization of this variable. Nodes representing membership functions can be modified during learning. The third layer contains rule nodes that evolve through supervised/unsupervised learning. Rule nodes are defined by two vectors of connection weights, which are adjusted through a hybrid learning technique. The fourth layer of neurons represents fuzzy quantification for the output Variables and the fifth layer carries out the defuzzification and calculates the numerical value for the output variable. The evolving process can be based on two assumptions. Either no rule nodes exist prior to learning and all of them are created during the evolving process or there is an initial set of rule nodes that are not connected to the input and output nodes and become connected through the learning (evolving process). The latter case is more biologically plausible.

G. Dynamic Evolving Fuzzy Neural Networks

Dynamic Evolving Fuzzy Neural Networks (dmEFuNN) [18] is a modified version of the EFuNN developed with the idea of not only the winning rule node's activation is propagated but also a group of rule nodes that is dynamic selected for every new input vector, and their activation values are used to calculate the dynamical parameters of the output function. While EFuNN implements Mamdani type fuzzy rules, dmEFuNN implements Takagi-Sugeno fuzzy rules. The first, second and third layers of dmEFuNN have exactly the same structures and functions as the EFuNN. The fourth layer which is called the fuzzy inference layer selects m rule nodes from the third layer which have the closest fuzzy normalized local distance to the fuzzy input vector, and then, a Takagi-Sugeno fuzzy rule will be formed using the weighted least square estimator. The last layer calculates the output of dmEFuNN. The number m of activated nodes used to calculate the output values for a dmEFuNN is not less than the number of the input nodes plus one. Like the EFuNNs, the dmEFuNNs can be used for both offline learning and online learning thus optimizing global generalization error, or a local generalization error. In dmEFuNNs, for a new input vector (for which the output vector is not known), a subspace consisted of m rule nodes are found and a first order Takagi Sugeno fuzzy rule is formed using the least square estimator method. This rule is used to calculate the dmEFuNN output value. In this way, a dmEFuNN acts as a universal function approximator using m linear functions in a small m dimensional node subspace. The accuracy of the approximation depends on the size of the node subspaces, the smaller the subspace is, the higher the accuracy. It means that if there are sufficient training data vectors and sufficient rule nodes are created, a satisfying accuracy can be obtained.

H. Self Constructing Neural Fuzzy Inference Network

Self Constructing Neural Fuzzy Inference Network (SONFIN) [20] [21] [22] implements a Takagi-Sugeno type fuzzy inference system which consists of six layers. Fuzzy rules are created and adapted as online learning procedure via a simultaneous structure and parameter identification. The SONFIN architecture is in fact similar to the ANFIS. Layer 1 up to 4 and 6 are functioning the same as they are in the ANFIS architecture. The fifth which is the consequent layer can hold two types of nodes. The first type represents the fuzzy sets by membership functions while the second type is optional and gains its inputs from the first and the fourth layer. Constructing of SONFIN happens concurrently both by a structure and a parameter learning method. The structure learning identifies both the precondition and consequent parts of the rules by minimizing the number of rules and membership functions for the input and by optimally generating new membership functions for the output variables. Parameter learning uses Least Mean Squares [LMS] or Recursive Least Squares [RLS] algorithms to adjust consequent parameters and backpropagation for precondition parameters. To enhance knowledge representation ability of SONFIN, a linear transformation for each input variable can be incorporated into the network so that much fewer rules are needed or higher accuracy can be achieved. Proper linear transformations are also learned dynamically in the parameter identification phase of SONFIN.

I. Fuzzy INference Environment Software with Tuning

Fuzzy INference Environment Software with Tuning (FINEST) [23] also called Fuzzy Inference and Neural Network in Fuzzy Inference Software is a software environment designed to tune the fuzzy inference itself. FINEST has capable of tuning two kinds of process: the tuning of fuzzy predicates and combination functions and the tuning of an implication function. FINEST has the following three important features.

(1) Improved generalized modus ponens:

FINEST framework provides a mechanism based on the improved generalized modus ponens for fine tuning of fuzzy predicates and combination functions and tuning of the implication function.

The generalized modus ponens is improved in the following four ways:

a) Aggregation operators that have synergy and cancellation nature
b) A parameterized implication function
c) A combination function that can reduce fuzziness
d) Backward chaining based on generalized modus ponens.

(2) Mechanism which can tune the inference method as well as fuzzy predicates:

The tuning mechanism is based on the improved generalized modus ponens. That is, aggregation operators with synergy and cancellation nature are defined using some parameters, indicating the strength of the synergistic effect, the area influenced by the effect, etc., and the tuning mechanism is designed to tune also these parameters. In the same way, the tuning mechanism can also tune the implication function and combination function. In short, the tuning mechanism of FINEST can be used to tune not only fuzzy predicates as conventional systems can do, but also the various parameters of the improved generalized modus ponens. Moreover, inner parameters of functions which are expressed as algorithmic representation of fuzzy data can be tuned if the derivative functions with respect to these parameters are given.

(3) Software environment for debugging and tuning:

The software environment is designed for carrying out forward and backward-chaining based on the improved generalized modus ponens and for tuning the various parameters of a system.

FINEST uses backpropagation learning algorithm for the fine-tuning of the parameters and provides a framework to tune any parameter which appears in the nodes of the network representing the calculation process of the fuzzy data if the derivative function with respect to the parameters is given.

1.4. Applications areas of Neuro-Fuzzy Systems

The use of NFS is proliferating into many sectors in our social and technological life. Based on the scope of collected articles on NFS applications, Samarjit Kar et.al [24] in their survey paper classified NFS applications into different categories such as student modeling system, medical system, economic system, electrical & electronics system, traffic control, image processing & feature extraction, manufacturing & system modeling, forecasting & predictions, NFS enhancements and social sciences for different research and problem domains.

i) NFS in student modeling: Neuro fuzzy system has a wide range of applications in the educational field and new directions are constantly given in educational research. According to Stathakopoulou et al. [25] student modeling is consisted of two components: the student model and the diagnostic module. The student model is one of the components of an intelligent tutoring system (ITS) which provides a description of student related information such as his knowledge level, skills or even preferences while diagnosis is the inference process which results in the end updates of the student model. Student modeling includes student classification, monitoring students' actions, processing intelligent learning environment

(ILE), assessing students' knowledge, evaluating students in intelligent tutoring system, modeling students in web based ITS etc.

ii) NFS in medical system: A medical system (also sometimes called health care system) is the organization of people, institutions and resources to deliver health care services to meet the health needs of target populations. Presently diseases in developing countries like Ethiopia have emerged as number one killer in both urban and rural areas of the country due to the increase of population from time to time. It will be of greater value if the diseases are diagnosed in its early stage. Correct diagnosis of the disease will decrease the death rate due to different diseases. Many clinical tests are being done to find the presence of the disease. In last decade neuro fuzzy applications in medical system are getting huge attention and that is why much relevant research has been conducted. NFS are being used for various typical disease diagnoses like brain disorder, cardiac disease, breast cancer, alzheimer, thyroid disorder, leukemia, hypotension, heart disease etc.

iii) NFS in economic system: An economic system can be defined as an organization where a person, country or area makes, distributes, consumes, buys or sells services and goods. This type of system has a direct impact on various governments and also on public activities. NFS can be applied in various field of economic system like state economic, stock market, toll collection, gas condensate, energy consumption, electric load forecasting, price prediction, supply chain management etc. Since the last decade, Neuro fuzzy applications in economic systems are attaining huge attention of many researchers and a number of relevant researches have been conducted. Starting from stock market to supply chain network, NFS has a wide range of applications in the economic systems.

iv) NFS in traffic control: Neuro fuzzy system has a wide range of applications in the traffic control since last decade. Recently a number of researchers are paying their attention in this category. Road traffic control is the process which is used to describe how councils and highway authorities control use of the road network in order to achieve improvements in road safety and efficiency. Network traffic control is the process of managing, prioritizing, controlling or reducing the network traffic to reduce congestion, latency and packet loss.

V) NFS in image processing and feature extraction: Neuro fuzzy system has a wide range of applications in the imaging analysis. Imaging analysis is the process of extraction data or information from images by means of image processing techniques. Computers are indispensable for the analysis of large amounts of data which contains the fields of computer or machine vision and medical images for tasks that require complex computation for the extraction of quantitative information and makes use of pattern recognition, digital geometry, and signal processing. Some applications related to image processing and feature extraction include emotion recognition, image stage analysis, noisy image processing, face recognition and image compression.

Vi) NFS in forecasting and prediction: Forecasting and prediction is the process to predict future events and conditions and should be key decision-making elements for management in service organizations. The term 'forecasting' is sometimes reserved for estimates of values at certain specific future times, while the term prediction is used for more general estimates of values over a long period of times.

Vii) NFS in manufacturing and system modeling: Manufacturing system includes equipment, products, people, information, control and support functions for the competitive development to satisfy market needs. The term may refer to a range of human activity, from handicraft to high tech, but is most commonly applied to industrial production in which raw materials are transformed into finished goods on a large scale. System modeling concerns modeling the operation of an unknown system from a set of measured input output data and has a wide range of applications in various areas such as control, power systems, communications, and machine intelligence. Systems modeling may be used in different ways as part of a process for improving and understanding of a situation, identifying problems or formulating opportunities and supporting decision making. In business and IT development the term "systems modeling" has multiple meaning such as functional modeling, business process modeling, enterprise modeling etc. Applications in this category includes autonomous vehicles, gear industry, underwater robotics, anti lock braking system, supply chain management, unmanned flight control, pneumatic system, software development time estimation, time varying system etc.

Viii) NFS in electrical and electronics system: Impact of electrical and electronics system in our daily life is increasing day by day. Electrical systems differ around the world both in voltage and less critically frequency. It is used to connect one or more pieces of equipment to or part of a structure and designed to provide a service such as heat or electricity or water or sewage disposal. Electronic systems are groupings of electronic circuits and components that focus on the higher abstraction level concerns first and foremost, used to accomplish one or more complex functions. Both electrical and electronics systems enhance the overall operation and also improve the operator's safety, through various safety circuits and applied methods. Some of the applications implemented by NFS in the field of electrical and electronics system are thermal process, electrical drives, transformer currents, circuit theory, power system, servo system and signal processing.

1.5 Comparative Analysis

Neuro-Fuzzy computing which combines the merits of neural and fuzzy logic systems enables one to build more intelligent decision-making systems. This incorporates the generic advantages of artificial neural networks like massive parallelism, robustness, and learning in data-rich environments into the system. The modeling of imprecise and qualitative knowledge as well as the transmission of uncertainty is possible through the use of fuzzy logic. Besides these generic advantages, the neuro-fuzzy approach also provides the corresponding application specific merits. The features of FL, NN and NF are given in table 1 below. It can be seen that they work at different levels of abstraction and individually provide rich functionality, which when brought together in a cohesive manner, results in an intelligent system.

TABLE 1 Comparison Based on the Features of FL, NN and NF

S.No	**Features**	**FL**	**NN**	NF
1.	Mathematical model	SG	Bad	Good
2.	Learning ability	Bad	Good	Very Good
3.	Knowledge representation	Good	Bad	Very Good
4.	Expert knowledge	Good	Bad	Good
5.	Nonlinearity	Good	Good	Very Good
6.	Optimization ability	Bad	SG	Good
7.	Fault tolerance	Good	Good	Very Good
8.	Uncertainty tolerance	Good	Good	Very Good
9.	Real-time operation	Good	SG	Very Good

Note: SG- Slightly Good

1.6. Conclusion and future work

This survey paper reviewed the concept of Fuzzy Logic Systems and Artificial Neural Networks as computational models and why neuro-fuzzy systems are created. As it was discussed this fusion can combine the learning and adaptation capabilities of Neural Networks with the easy interpretability and high expressive power of fuzzy rules in an effective way. Nine different neuro fuzzy architectures were presented and it can be concluded that these are the most important ones although there are other structure variations too. Usually each architecture organizes its nodes in a slightly different way and consequently they use specific learning algorithms which are adapted to the different structures. Most NF models use gradient descent techniques to learn the membership function parameters. For faster learning and convergence, it will be interesting to explore other efficient neural network learning algorithms (e.g. conjugate gradient search) instead of backpropagation. The different Neuro-Fuzzy models were also compared and presented as a table summarizing the advantages and limitations of each presented architecture. All in all it can be said that ANFIS architecture is the most popular and widespread among the Neuro-Fuzzy systems for various applications. This is mainly because the ANFIS model has higher accuracy than the other Neuro-Fuzzy model types which compensates its less interpretable structure. Due to the lack of a common framework it remains often difficult to compare the different neuro-fuzzy models conceptually and evaluate their performance comparatively. As a future work it remained to show the application of the discussed neuro fuzzy systems to real situations to show that ANFIS is more accurate than the rest.

II. Acknowledgements

Authors would like to acknowledge Ethiopian Ministry of Education for the sponsor and Punjabi University, Patiala for providing adequate library and internet facility.

III. REFERENCES

[1]. LA. Zadeh, "Soft computing and fuzzy logic. ," IEEE software, vol. 11, no. 6, pp. 48-56, 1994.

[2]. Lotfi A.Zadeh L Zadeh, "Fuzzy sets," Information and Control, vol. 8, no. 3, pp. 338-353, 1995.

[3]. Jerry M., and George C. Mouzouris. Mendel, "Designing fuzzy logic systems.," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing., vol. 44, no. 11, pp. 885-895, 1997.

[4]. Dogan. Ibrahim, "An overview of soft computing.," Procedia Computer Science., vol. 102, pp. 34-38, 2016.

[5]. Andrew G., and Richard S. Sutton. Barto, "Reinforcement learning. ," Neural systems for control., pp. 7-29, 1998.

[6]. Jose, F. Morgado Dias, and Alexandre Mota. Vieira, "Neuro-fuzzy systems: a survey.," In 5th WSEAS NNA International Conference on Neural Networks and Applications, Udine, Italia. , vol. 3, pp. 414-419, March 2004.

[7]. JS. Jang, "ANFIS: adaptive-network-based fuzzy inference system. ," IEEE transactions on systems, man, and cybernetics., vol. 23, no. 3, pp. 665-685, 1993.

[8]. Chun-Tian, Jian-Yi Lin, Ying-Guang Sun, and Kwokwing Chau. Cheng, "Long-term prediction of discharges in Manwan Hydropower using adaptive-network-based fuzzy inference systems models.," in In International Conference on Natural Computation. , vol. 3612, Springer, Berlin, Heidelberg., 2005, pp. 1152-1161.

[9]. Hamid R., and Pratap Khedkar. Berenji, "Learning and tuning fuzzy logic controllers through reinforcements. ," IEEE Transactions on neural networks 3, no. 5 (1992): 724-740., vol. 3, no. 5, pp. 724-740, 1992.

[10]. C-T., and C. S. George Lee. Lin, "Neural-network-based fuzzy logic control and decision system.," IEEE Transactions on computers , vol. 40, no. 12, pp. 1320-1336, 1991.

[11]. Detlef, and Rudolf Kruse. Nauck, "Neuro-fuzzy systems for function approximation.," Fuzzy sets and systems., vol. 101, no. 2, pp. 261-271, 1999.

[12]. Detlauf, and Rudolf Kruse. Nauck, "NEFCLASSmdash; a neuro-fuzzy approach for the classification of data. ," in In

Proceedings of the 1995 ACM symposium on applied computing., 1995, pp. 461-465.

[13]. Tom M., and Bert Kappen. Heskes, "On-line learning processes in artificial neural networks. ," North-Holland Mathematical Library., vol. 51, pp. 199-233, 1993.

[14]. N. Kasabov, "A framework for evolving connectionist systems and the" eco" training method.," in In proceedings of ICONIP, pp. 1232-1235.

[15]. Nikola K. Kasabov, "Adaptable neuro production systems. ," Neurocomputing , vol. 13, no. 2, pp. 95-117, 1996.

[16]. Michael C., and Paul Smolensky. Mozer, "Skeletonization: A technique for trimming the fat from a network via relevance assessment.," In Advances in neural information processing systems., pp. 107-115, 1989.

[17]. Teuvo. Kohonen, "The self-organizing map. ," in proceedings of the ieee, 78., 1990.

[18]. Nikola K., and Qun Song. Kasabov, Dynamic Evolving Fuzzy Neural Networks with" m-out-of-n" Activation Nodes for On-line Adaptive Systems. Department of Information Science.: University of Otago, 1999.

[19]. Nikola. Kasabov, "Evolving fuzzy neural networks for adaptive, on-line intelligent agents and systems. ," Recent Advances in Mechatronics, pp. 27-41, 1999.

[20]. Ajith. Abraham, "Neuro Fuzzy Systems: state of Art Modelling Techniques," in In proceedings of the sixth international work conference on Artificial and Natural Neural Networks., 2001.

[21]. Ajith, and Baikunth Nath. Abraham, "Hybrid intelligent systems design: A review of a decade of research. ," IEEE Trans-actions on Systems, Man and Cybernetics (Part C)., vol. 3, no. 1, pp. 1-37, 2000.

[22]. Jose, F. Morgado Dias, and Alexandre Mota. Vieira, "Neuro-fuzzy systems: a survey.," In 5th WSEAS NNA International Conference on Neural Networks and Applications, Udine, Italia. , 2004.

[23]. Shun'ichi, Takuya Oyama, and Thierry Arnould. Tano, "Deep combination of fuzzy inference and neural network in fuzzy inference software友INEST. ," Fuzzy Sets and Systems., vol. 82, no. 2, pp. 151-160, 1996.

[24]. Samarjit, Sujit Das, and Pijush Kanti Ghosh. Kar, "Applications of neuro fuzzy systems: A brief review and future outline," Applied Soft Computing , vol. 15, pp. 243-259, 2014.

[25]. Regina, George D. Magoulas, Maria Grigoriadou, and Maria Samarakou. Stathacopoulou, "Neuro-fuzzy knowledge processing in intelligent learning environments for improved student diagnosis. ," Information Sciences., vol. 170, no. 2, pp. 273-307, 2005.

[26]. Luigi, Gianguido Rizzotto, Mario Lavorgna, Giuseppe Nunnari, M. Gabriella Xibilia, and Riccardo Caponetto. Fortuna, Soft computing: new trends and applications.: Springer Science & Business Media, 2012.

[27]. S.Sumathi and S.N.Deepa S.N.Sivananadam, Introduction to Fuzzy Logic using Matlab.: Springer-Verlag Berlin Heidelberg , 2007.

[28]. Ph.D., D.Sc.,Professor Jacek ~ski, Ph.D., D.Sc. Professor Ernest Czogalat, Fuzzy and N euro-Fuzzy Intelligent Systems, Prof. Janusz Kacprzyk, Ed. Poland: A Springer-Verlag Company.

[29]. C., B. Krause, and H-J. Zimmermann. Von Altrock, "Advanced fuzzy logic control technologies in automotive applications.," In Fuzzy Systems, 1992., IEEE International Conference , pp. 835-842, 1992.

[30]. Kazuo, and Hideyuki Takagi. Asakawa, "Neural networks in Japan.," Communications of the ACM , vol. 37, no. 3, pp. 106-112, 1994.

[31]. J. S. R. Jang, "Neuro-fuzzy modeling: architectures, analyses and applications [dissertation]. ," California: University of Berkeley), 1992.

[32]. J.S. Jang, "ANFIS: adaptive-network-based fuzzy inference system.," IEEE transactions on systems, man and cybernetics., vol. 23, no. 3, pp. 665-685, 1993.

[33]. S. R., P. J. Nikumbh, and S. P. Kulkarni. Nikam, "Fuzzy logic and neuro-fuzzy modeling. ," Journal of Artificial Intelligence., vol. 3, no. 2, p. 74, 2012.

[34]. Ebrahim H., and Sedrak Assilian. Mamdani, "An experiment in linguistic synthesis with a fuzzy logic controller.," International journal of man-machine studies, vol. 7, no. 1, pp. 1-13, 1975.

[35]. Ajith. Abraham, "Adaptation of fuzzy inference system using neural learning.," Fuzzy systems engineering (2005): 914-914., pp. 914-914, 2005.

[36]. Lotfi A.Zadeh L Zadeh, "Fuzzy sets," Information and Control, vol. 8, no. 3, pp. 338-353, 1995.