# Neuro-Fuzzy Systems : An Overview

Kushagra Gupta
Senior Undergraduate
Mathematics Department
Indian Institute of Technology - Delhi

Saransh Mahajan
Senior Undergraduate
Mathematics Department
Indian Institute of Technology - Delhi

***Abstract*—In this paper we will try to study the existing literature on Neuro-fuzzy Models, their usage and applications in depth. The neuro-fuzzy models are important systems which can learn the rules themselves, or refine the rules formed by experts in a fuzzy logic based inference system. We would try to highlight the advantages of using such hybrid systems and how they are different from regular system. We would learn the structures and working of such models. Further we would look upon the applications of such systems in classification and time series analysis. Finally we will try an small example to understand the working.**

***Keywords*—Neuro-Fuzzy, ANFIS, Fuzzy inference System, Sugueno Model, Mandami Model**

## I. Introduction

With the advancement in the fields of AI and Machine Learning and the availablity of huge amounts of data we have reached to a point where we can build intelligent systems to do our day-to-day tasks which were earlier assumed to be very difficult for computers. But the available data for some tasks need not always be precise. We often deal with things that are uncertain and imprecise, for example when we say that the weather today is very hot.

The human cognitive process essentially deals with uncertain and imprecise information. It is to deal with such kinds of uncertainties that Zadeh introduced Fuzzy Sets. Fuzzy sets are a way to represent and process data that is not precise, but is rather fuzzy. These sets manipulate data that possesses non statistical uncertainties. The use of fuzzy logic in humanlike inference systems has been proved to be very effective and has enable the computers to do tasks which were earlier considered impossible for them. Consider the example of classifying students in a class as good/average/poor by a teacher, so that he/she can alter teaching styles for each type. Now almost all around the world, this task is done based on intutions. But use of fuzzy inference systems have enabled the computers to perform this task wih greater accuracy than humans. [1]. If you look around yourself, you'll find many appliances that uses fuzzy logic, for example Washing Machine, Rice Cooker, Air Conditioner etc. [2]

An important part of Fuzzy Logic based infence systems is the rule base. But sometimes it is difficult to form the rules. The experts of the field may not be reliable or there may be a shift in context of data over time. So to overcome these disadvantages neuro-fuzzy models are used which could finetune or even find the rules based on the given training data.

In the following sections we would look at the basics of Neural Networks, Fuzzy Logic (Knowledge of basic fuzzy sets is assumed), Fuzzy Decision Systems and then the introduction of Neuro-Fuzzy System. We would then look the ANFIS (Adaptive Neuro Fuzzy inference System) in depth. Further we would try to look at how these networks are trained and some applications of such systems.

## II. Basics

### A. Fuzzy Logic

Just like classical logic is decision making with crisp sets, Fuzzy logic involves decision making with fuzzy sets.

In classical logic, a simple proposition $P$ is a linguistic statement contained within a universe of elements, $X$, that is a collection of elements in X that are either strictly true or strictly false, i.e. have a truth value of either $1$ or $0$. If $U$ is the universe of all propositions, then $T$ is a mapping of the elements, $u$, in these propositions (sets) to the binary quantities $\{0, 1\}$, or

$$T : u \in U \to \{0, 1\} \tag{1}$$

A fuzzy logic proposition, P, is a statement involving some concept without clearly defined boundaries. For example, if temperature is medium and humidity is high turn the fan on. This is a fuzzy proposition because the word medium and high are ambiguous. The language we speak is usually vague, since it involves imprecise terms. Examples of fuzzy propositions include describing a persons height as tall, or a person as fat or skinny. In fuzzy logic, the truth value assigned to proposition $P$ can be any value on the interval $[0, 1]$, in contrast to classical logic where it can be either $1$ or $0$. The assignment of the truth value to a proposition is actually a mapping from the interval $[0, 1]$ to the universe $U$ of truth values, $T$

$$T : u \in U \to [0, 1] \tag{2}$$

The goal in fuzzy logic is to form the theoretical foundation for reasoning about imprecise propositions, ie approximate reasoning. Approximate reasoning is basically classical logic, but instead of precise propositions it deals with partial truths. Fuzzy logic provides a way to apply human reasoning capabilities to knowledge based systems. It provides us with the mathematical strength to deal with uncertainties in the human cognitive process. The conventional approaches to knowledge representation do not have the means to represent the meaning of fuzzy concepts, because of which the approaches based on first order logic do not give us the framework for dealing with common sense knowledge, which is lexically imprecise.

Fuzzy logic has the following characteristics:

- In fuzzy logic, exact reasoning is viewed as a limiting case of approximate reasoning.
- In fuzzy logic, everything is a matter of degree.
- In fuzzy logic, knowledge is interpreted a collection of elastic or, equivalently, fuzzy constraint on a collection of variables.
- Inference is viewed as a process of propagation of elastic constraints.
- Any logical system can be fuzzified.

Fuzzy logic is better than classical logic because systems based on fuzzy logic are suitable for approximate reasoning, and because it allows us to take decisions under uncertain information.

### B. Fuzzy inference Systems (FIS)

A fuzzy inference system (FIS) is a system that uses fuzzy set theory to map inputs (features in the case of fuzzy classification) to outputs (classes in the case of fuzzy classification).

A Fuzzy Inference System has the following components:

- Fuzzifier : Crisp Input is fuzzified by using membership functions. For example if temprature input is coming as 20. It would be assigned its membership in all of the fuzzy sets of the variable temprature.
- Inference Engine : inference engine consists a set of Rules (called Fuzzy Rule Base) based on expert knowledge which are used to calculate the output value corresponding to each rule. Then the output value of each rule is aggregated into one using the firing strengths of each rule.
- Defuzzifier : The aggregated output is then defuzzified and a crisp value is given as output.

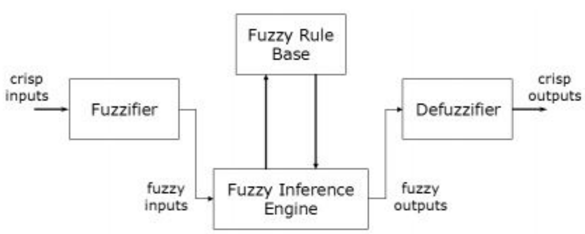


Fig. 1: Fuzzy Inference System

Based on the types of aggregation function and the types of output, the inference systems could be classified into 2 broad categories : Mamdani and the Sugeno.

The main disadvantage is the fuzzy inference systems is the process of calculating rule base and output memberships. It is highly dependent on the expert knowledge.

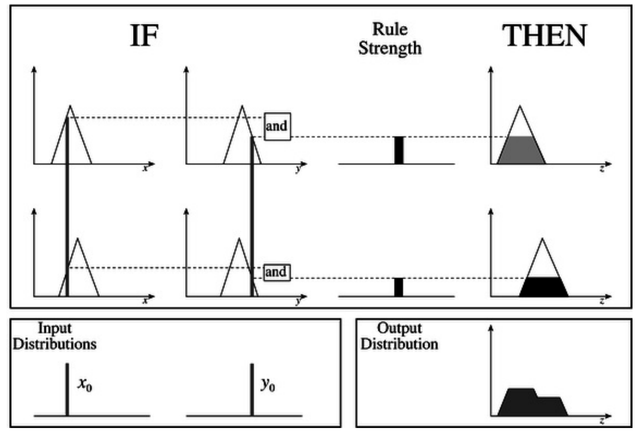


Fig. 2: Mamdani FIS

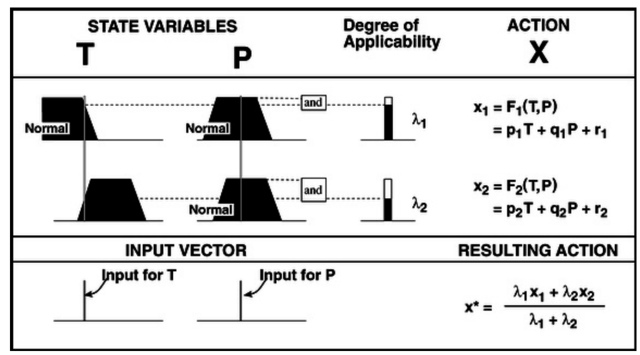


Fig. 3: Sugeno FIS

### C. Neural Networks

Artificial neural networks(ANNs) are simplified mathematical models of brain-like systems and they function as parallel distributed computing networks. Neural networks need to be trained with some training data, using which they can learn an approximation to a function, patterns in the data, clusters in the data and dependencies etc.

Neural networks can automatically adjust their weights (the connections between the neurons) to act as classifiers or regressors etc. Adaptivity allows the neural network to perform well even when the environment or the system being controlled varies over time.

Neural networks have an input layer, certain number of hidden layers and an output layer. The input layer takes the input, passes it on to the next layer, where they combine according to the synaptic weights and then typically a continuous function, for example a sigmoid is applied to this combination. This then is passed on to the next layer, where the same thing is repeated till we reach the output layer.

Neural networks usually are part of supervised learning, and the output produced by ANNs is compared with the expected output to find the error. The error function can be defined in many ways, and one common way is the mean squared error.

The ANN learns through the back propagation algorithm, where one forward pass through the ANN is followed by an error computation at the output side, and then the weights are adjusted so as to decrease the subsequent error. This adjustment is what is called the backpropagation algorithm, where the weights are modified according to the error produced. This is continued till the error falls in the acceptable range.

Neural Networks have found a wide range of applications in the field of machine learning. They have enabled us to make self driving cars, auto pilot etc. But the main disadvanage is that they act like a black box. It is very difficult to analyze that what is happening or what features are important for a certain output. This opaque behaviour sometimes restricts the interpretablity of the model.

| Skills | | Fuzzy Systems | Neural Nets |
|---|---|---|---|
| *Knowledge acquisition* | Inputs | Human experts | Sample sets |
| | Tools | Interaction | Algorithms |
| *Uncertainty* | Information | Quantitive and Qualitive | Quantitive |
| | Cognition | Decision making | Perception |
| *Reasoning* | Mechanism | Heuristic search | Parallel computations |
| | Speed | Low | High |
| *Adaption* | Fault-tolerance | Low | Very high |
| | Learning | Induction | Adjusting weights |
| *Natural language* | Implementation | Explicit | Implicit |
| | Flexibility | High | Low |

Fig. 4: Fuzzy System and Neural Net properties

## III. Neuro-Fuzzy Models : Introduction

Neuro-Fuzzy models combine both the neural nets and the Fuzzy Inference System to incorporate the advantages of both the model and avoid the disadvantages. The neural nets are easy to train given some sample data, are parallely processed and try to model the brain. Fuzzy Rule based inference systems are easy to interpret but suceptible to expert's bias. It is not at all data driven. But combining these two we can achieve the advantages of both and the disadvantages of none. Such systems are called hybrid intelligent system. A hybrid intelligent system is one that combines at least two intelligent technologies.

In ANNs, the learning as mentioned before is done using the backpropagation algorithm, which is how the ANN acquires knowledge. But this process is quite slow, and the ANN after training behaves like a black box, in that it is difficult to understand what goes on inside the ANN. It is usually very difficult to obtain rules from the ANN, and since it behaves like a black box we cannot use any special knowledge, insight or expertise we have about a particular problem to make the learning faster and easier.

In expert systems such as fuzzy based decision systems, the knowledge is in the form of IF-THEN rules and seeing the individual rules does provide the user with some information about the system. But in contrast, in neural nets, the connection weights cannot be seen as discrete knowledge and all the synapse connections together encode all the knowledge, which is why its a black box.

Fuzzy systems on the other hand, are such that their behaviour depends on fuzzy rules, and their performance changes based on adjusting the rules. Since knowledge is difficult to acquire,applications of fuzzy systems are restricted to the fields where expert knowledge is available and the number of input variables is small. Expert knowledge may sometimes even be incorrect. Not at all data driven.

Every intelligent technique has particular computational properties (e.g. ability to learn, explanation of decisions) that make them suited for particular problems and not for others. So neural networks can recognize patterns in the data, they are unable to explain these patterns. Fuzzy logic systems on the other hand, can reason with imprecise information, but require expert knowledge to acquire rules on which to base the decisions on.

Knowledge in a rule-based expert system is represented by IF-THEN production rules. Knowledge in neural networks is stored as synaptic weights between neurons

To overcome both the problems mentioned above, hybrid neural networks are used. Fuzzy neural networks gives us the best of both worlds, in that the solve the problem of ANNs behaving as a black box by using fuzzy rules, and the neural networks can be used to tune the membership problem of fuzzy rule based systems, thus solving the problem of knowledge acquisition. Neural nets are used to obtain fuzzy rules from numerical data in this hybrid system.

While fuzzy logic provides an inference mechanism under cognitive uncertainty, computational neural networks offer exciting advantages, such as learning, adaptation, fault tolerance, parallelism and generalization.

Fuzzy neural networks are essentially systems which deal with cognitive uncertainties in a manner more like humans.

The above neural nets and FIS can be combined in various ways, but mainly they are classified in 2 categories:

- Heterogeneous Systems : A heterogeneous neuro A heterogeneous neuro-fuzzy system is fuzzy system is hybrid system that consists of a neural network and a fuzzy system working as independent components.
- Homogeneous System : A homogeneous neuro-fuzzy system is a neural network that is functionally equivalent to a fuzzy inference model.

For the further discussion we'll constrain ourselves to the homogeneous systems. Some of the example models are ANFIS, FuNe, Fuzzy RuleNet, GARIC, NEFCLASS and NEFCON.[3]

## IV. Neuro-Fuzzy Models : Structure

In fuzzy neural networks, neural networks are used to tune membership functions of fuzzy systems that are used for rule generation in any particular application. In theory, neural networks, and fuzzy systems are equivalent in that they are convertible.

In order to develop fuzzy neural systems, we proceed as follows : First we make a fuzzy neuron, which is done based on the understanding of biological neuronal morphologies. This is followed by learning mechanisms. Basically, we have the following steps:

- We build fuzzy neurons taking inspiration from biological neurons.
- Then we model synaptic connections between the neurons( again an effort at mimicking the actual human biology) to incorporate fuzziness into the neural network. These connections act as the weight for the inputs into the next layer.
- We conclude by building learning algorithms, ie learning how to adjust the weights of the connections between the fuzzy neurons, because this is essentially what makes the neural networks so special : the ability to adjust to the data so as to minimise the error.

A fuzzy neural net is equivalent to a fuzzy inference model. It can be trained to produce IF-THEN fuzzy rules. Expert knowledge can be incorporated into its structure. We also avoid the computational burden due to the connections between the different layers. In general, the fuzzy neural net has the same structure as an ANN, with input and output layers, and three hidden layers which correspond to the membership functions and fuzzy rules.

In a usual neural net The signal $x_i$ interacts with the weight $w_i$ to produce the product

$$p_i = w_i x_i, \;\; i = 1, \ldots, n \tag{3}$$

The input information $p_i$ is aggregated, by addition, to produce the input

$$net = p_1 + ... + p_n = w_1 x_1 ... w_n x_n \tag{4}$$

The neuron uses its transfer function f, which could be a sigmoidal function,

$$f(t) = \frac{1}{(1 + e^{-t})} \tag{5}$$

to compute the output

$$y = f(net) = f(w_1 x_1 + ... + w_n x_n) \tag{6}$$

A hybrid neural network is a neural network with crisp signals and weights and crisp transfer function. However,

- We can combine $x_i$ and $w_i$ using some other continuous operation.
- We can aggregate the $p_i$s with some other other continuous function.
- $f$ can be any continuous function from input to output.

All inputs ,outputs and weights are real numbers taken from the unit interval $[0, 1]$.

In a fuzzy neuron, we have different kinds of aggregate functions, some of which are:

1) **AND Fuzzy Neuron :** The signal $x_i$ and $w_i$ are combined by the maximum operator to produce

$$p_i = max\{w_i, x_i\}, i = 1, 2. \tag{7}$$

The input information pi is aggregated by the minimum operator to produce the output

$$y = min\{p_1, p_2\} = min\{w_1 \vee x_1, w_2 \vee x_2\} \tag{8}$$

of the neuron.

2) **OR Fuzzy Neuron :** The signal $x_i$ and $w_i$ are combined by the maximum operator to produce

$$p_i = min\{w_i, x_i\}, i = 1, 2. \tag{9}$$

The input information pi is aggregated by the minimum operator to produce the output

$$y = max\{w_1 \wedge x_1, w_2 \wedge x_2\} \tag{10}$$

of the neuron.

3) **OR (max-product) Fuzzy Neuron :** The signal $x_i$ and $w_i$ are combined by the maximum operator to produce

$$p_i = w_i x_i, i = 1, 2. \tag{11}$$

The input information pi is aggregated by the minimum operator to produce the output

$$y = max\{w_1 x_1, w_2 x_2\} \tag{12}$$

of the neuron.

The AND and OR fuzzy neurons realize pure logic operations on the membership values. The different connection weights are to signify the different levels of impact that individual inputs might have on the result of aggregation.

Neuro-fuzzy system has a structure similar to multi layer ANN. It has three hidden layers that represent membership functions and fuzzy rules, along with the input and output layers.

For the purpose of illustration, let us have just two inputs, $x_1$ and $x_2$ and one output $y$.Input $x_1$ is represented by fuzzy sets $A_1$, $A_2$ and $A_3$, input $x_2$ by fuzzy sets $B_1$, $B_2$ and $B_3$ and output $y$ by fuzzy sets $C_1$ and $C_2$.

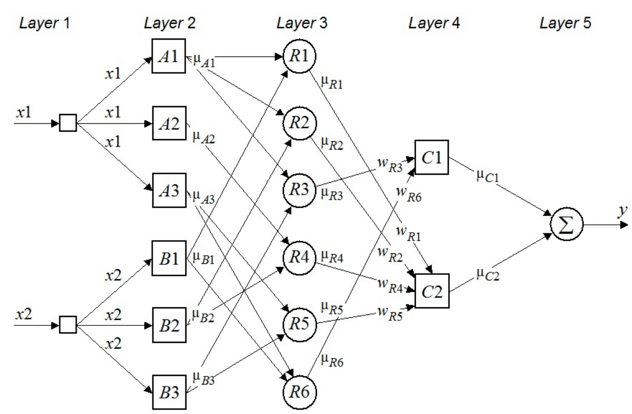


Fig. 5: Fuzzy Neural Net

Each layer in the neuro-fuzzy system is associated with a particular step in the fuzzy inference process. We explain below each of these layers:

1) **Layer 1** is the input layer. Neurons here simply transmit the crisp input signal to the next layer.

$$y_i^{(1)} = x_i^{(1)} \tag{13}$$

where $x_i^{(1)}$ is the input and $y_i^{(1)}$ is the output of input neuron $i$ in Layer 1.

2) **Layer 2** is the fuzzification layer. The antecedents in the fuzzy rules are fuzzy sets and this layers neurons correspond to these fuzzy sets. A fuzzification neuron receives a crisp input and determines the degree to which this input belongs to the neurons fuzzy set, by setting the activation function of these neurons to be the membership functions of the corresponding fuzzy sets. Fuzzification neurons may have any of the membership functions normally used in fuzzy systems.
Let us for demonstration take the membership functions used in the fuzzy neurons to be Triangular Fuzzy Membership. This function has parameters which control both its centre and width. The neuron input may remain constant, but the output will vary with the change of parameters, so the parameters of these neurons play the same role in a neuro-fuzzy system as synaptic weights in a neural network.

3) **Layer 3** is the fuzzy rule layer. Each neuron in this layer corresponds to a single fuzzy rule. A fuzzy rule neuron receives inputs from the fuzzification neurons that represent fuzzy sets in the rule antecedents. As an example, from the figure above we can see that the neuron R1, which corresponds to Rule 1, receives inputs from neurons A1 and B1.
In fuzzy systems, if a given rule has multiple antecedents, a fuzzy operator is used to obtain a single number that represents the result of the antecedent evaluation. The conjunction of the rule antecedents is evaluated by the fuzzy operation intersection. The same fuzzy operation can be used to combine multiple inputs to a fuzzy rule neuron. In a neuro-fuzzy system, intersection can be implemented by the product operator(we could also use the min operator). Thus, the output of neuron i in Layer 3 is obtained as:

$$y_i^{(3)} = x_{1i}^{(3)} \times x_{2i}^{(3)} \times ... \times x_{ni}^{(3)} \tag{14}$$

where $x_{1i}^{(3)}, x_{2i}^{(3)}, ..., x_{ni}^{(3)}$ are inputs and $y_i^{(3)}$ is the output of fuzzy rule neuron $i$ in Layer 3. For example

$$y_i^{(3)} = \mu_{A_1} \times \mu_{B_1} = \mu_{R_1} \tag{15}$$

The value of $\mu_{R_1}$ represents the firing strength of fuzzy rule neuron $R_1$.The weights between Layer 3 and Layer 4 represens the normalised degrees of confidence (known as certainty factors) of the corresponding fuzzy rules. These weights are adjusted during training of a neuro-fuzzy system, similar to how it is done for neural nets. Normalised degrees of confidence basically means that we normalize the degrees of confidence corresponding to each rule in the neuro fuzzy system. This is since an expert in the beginning may attach different importance to each rule and accordingly decide the weight of the synaptic connections to be within $[0, 1]$ but these weights get modified during training of the network.

4) **Layer 4** is the output membership layer. Neurons in this layer represent fuzzy sets used in the consequent of fuzzy rules. These neurons combine the inputs they get by using fuzzy union.This operation can be implemented by the probabilistic OR (also known as the algebraic sum). That is,

$$y_i^{(4)} = x_{1i}^{(4)} \oplus x_{2i}^{(4)} \oplus ... \oplus x_{ni}^{(4)} \tag{16}$$

where $x_{1i}^{(4)}, x_{2i}^{(4)}, ..., x_{ni}^{(4)}$ are inputs and $y_i^{(4)}$ is the output of fuzzy rule neuron $i$ in Layer 4. For example, in our case we have

$$y_i^{(4)} = \mu_{R_3} \times \mu_{R_6} = \mu_{C_1} \tag{17}$$

$\mu_{C_1}$ represents the integrated firing strength of fuzzy rule neurons $R_3$ and $R_6$. In fact, firing strengths of neurons in the output membership layer are combined in the same way as truth values of the fuzzy rules in the Mamdami system. In the Mamdani fuzzy system, output fuzzy sets are clipped by the truth values of the corresponding fuzzy rules. In the neuro-fuzzy system, we clip activation functions of the output membership neurons. For example, the membership function of neuron $C_1$ is clipped by the integrated firing strength $C_1$.

5) **Layer 5** is the defuzzification layer. Each neuron in this layer represents a single output of the neuro-fuzzy system. It takes the output fuzzy sets clipped by the respective integrated firing strengths and combines them into a single fuzzy set. The output of the neuro-fuzzy system is crisp, and thus a combined output fuzzy set must be defuzzified. Neuro-fuzzy systems can apply standard defuzzification methods, including the centroid technique, maximum , minimum, mean of maximum technique etc.

### A. *LEARNING*

Since a fuzzy neural net is also a multi-layer neural network, so it can use standard techniques such as the backpropagation algorithm. Given a training input-output pair, the algorithm computes the output and compares it with the expected output to find the error.This error is propagated backwards through the network from the output layer to the input layer. The neuron activation functions are modified as the error is propagated. To determine the necessary modifications, the backpropagation algorithm differentiates the activation functions of the neurons.

One of the other method often used in ANFIS is hybrid learning algorithm. The hybrid learning algorithm, consisting of Least Squares Method for the forward passing and the Gradient Decent method for the back propogation, is used to train the ANFIS. There are several ways of combining gradient descent method and the least squares method.

## V. ANFIS

ANFIS is neural net which behaves like sugueno fuzzy model based inference system.The Sugeno fuzzy model was proposed for a systematic approach to generating fuzzy rules from a given input-output data set. A typical Sugeno fuzzy rule can be expressed in the following form:

$$\begin{aligned}
&IF\ x_1\ is\ A_1\\
&AND\ x_2\ is\ A_2\\
&.......\\
&AND\ x_m\ is\ A_m\\
&THEN\ y = f(x_1, x_2, ..., x_m)
\end{aligned}$$

where $x_1$, $x2$, ... $x_m$ are input variables. $A_1, A_2, ...., A_m$ are fuzzy sets and $y$ is either a constant or a linear function of the input variables. When y is a constant, we obtain a zero-order Sugeno fuzzy model in which the consequent of a rule is specified by a singleton. When y is a first-order polynomial, i.e.

$$y = k_0 + k_1x_1 + k_2x_2 + ... + k_mx_m \tag{18}$$

we obtain a first-order Sugeno fuzzy model.

When the number of rules is not restricted, a zero-order Sugeno model has unlimited approximation power for matching well any nonlinear function arbitrarily on a compact set. This can be proved using the Stone-Weierstrass theorem.

The structure of ANFIS is almost similar to the standard neural network that we have discussed. The layers of the neural nets are described below.

- Layer 1 is the usual input layer.
- Layer 2 is the fuzzification layer.
- Layer 3 is the rule layer. Each neuron in this layer corresponds to a single Sugeno-type fuzzy rule. In an ANFIS, the conjunction of the rul antecedents is evaluated by the operator product.
- Layer 4 is the normalisation layer. Each neuron in this layer receives inputs from all neurons in the rule layer, and calculates the normalised firing strength of a given rule. The normalised firing strength is the ratio of the firing strength of a given rule to the sum of firing strengths of all rules. It represents the contribution of a given rule to the final result.
- Layer 5 is the defuzzification layer. (It is different from usual Fuzzy Neural Network). Each neuron in this layer is connected to the respective normalisation neuron, and also receives initial inputs, $x_1$ and $x_2$. A defuzzification neuron calculates the weighted consequent value of a given rule as,
$$y_i^{(5)} = x_i^{(5)}[k_{i0} + k_{i1}x_1 + k_{i2}x_2] \tag{19}$$
- Layer 6 is represented by a single summation neuron. This neuron calculates the sum of outputs of all defuzzification neurons and produces the overall ANFIS output, $y$.

Thus, the ANFIS shown in figure is indeed functionally equivalent to a firstorder Sugeno fuzzy model. However, it is often difficult or even impossible to specify a rule consequent in a polynomial form. Conveniently, it is not necessary to have any prior knowledge of rule consequent parameters for an ANFIS to deal with a problem. An ANFIS learns these parameters and tunes membership functions.

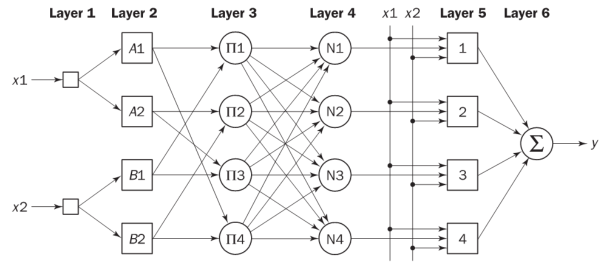


Fig. 6: ANFIS

## A. Learning in an ANFIS

An ANFIS uses a hybrid learning algorithm that combines the least-squares estimator and the gradient descent method (Jang, 1993). First, initial activation functions are assigned to each membership neuron. The function centres of the neurons connected to input $x_i$ are set so that the domain of $x_i$ is divided equally (discretization of the fuzzy sets), and the widths and slopes are set to allow sufficient overlapping of the respective functions. In the ANFIS training algorithm, each epoch is composed from a forward pass and a backward pass. In the forward pass, a training set of input patterns (an input vector) is presented to the ANFIS, neuron outputs are calculated on the layer-by-layer basis, and rule consequent parameters are identified by the least squares estimator. In the Sugeno-style fuzzy inference, an output, y, is a linear function. Thus, given the values of the membership parameters and a training set of $P$ input-output patterns, we can form $P$ linear equations in terms of the consequent parameters.

Let there be $m$ inputs and $n$ outputs. So there will be $n(1+m)$ total output parameters. Let the the $P$ equations formed be of the form

$$y_d = A \times k \tag{20}$$

where $y_d$ is a $P\times 1$ output vector containing the training output values, $A = P \times n(1+m)$ matrix containing the coeffecients of the linear equations, and k is the vector containing the unknown output parameters. The value of $P$is usually more than the number of output parameters, so the exact solution of the equations might not even exist. So we need least squares method to calculate the value of parameters. To minimize the least squares error i.e. $||Ak - y_d||^2$ we use the pseudoinverse technique:

$$k^* = (A^TA)^{-1}A^Ty_d \tag{21}$$

As soon as the rule consequent parameters are established, we can compute an actual network output vector, $y$, and determine the error vector, $e = y_d - y$. In the backward pass, the back-propagation algorithm is applied. The error signals are propagated back, and the antecedent parameters are updated.

In the ANFIS training algorithm suggested by Jang, both antecedent parameters and consequent parameters are optimised. In the forward pass, the consequent parameters are adjusted while the antecedent parameters remain fixed. In the backward pass, the antecedent parameters are tuned while the consequent parameters are kept fixed. However, in some cases, when the inputoutput data set is relatively small, membership functions can be described by a human expert. In such situations, these membership functions are kept fixed throughout the training process, and only consequent parameters are adjusted.

## VI. Applications

Neuro-fuzzy system have found a lot of applications in our daily life. The refrigerator you use, the AC you use might be using fuzzy logic. And neural networks have emerged as an exciting tool to learn the parameters of such system. These systems have found a large domain of applications beside the control systems.

Apart from the direct applications, neuro-fuzzy systems are also used as a classifier and to predict time series values. These applications further increase the domain where neuro-fuzzy systems are used.

### A. *Neuro-fuzzy classifiers*

Conventional approaches of pattern classification involve clustering training samples and associating clusters to given categories. The complexity and limitations of previous mechanisms are largely due to the lacking of an effective way of defining the boundaries among clusters. This problem becomes more intractable when the number of features used for classification increases. On the contrary, fuzzy classification assumes the boundary between two neighboring classes as a continuous, overlapping area within which an object has partial membership in each class. This viewpoint not only reflects the reality of many applications in which categories have fuzzy boundaries, but also provides a simple representation of the potentially complex partition of the feature space. In brief, we use fuzzy IF-THEN rules to describe a classifier.

The task of fuzzy classification is to generate an appropriate fuzzy partition of the feature space . In this context the word appropriate means that the number of misclassified patterns is very small or zero. Then the rule base should be optimized by deleting rules which are not used.

The neuro-fuzzy approach is better than neural network classifiers in the sense that prior knowledge about the training data set can be encoded into the parameters of the neuro-fuzzy classifier. This encoded knowledge, usually acquired from human experts or data visualization techniques, can almost always allow the learning process to begin from a good initial point not far away from the optimal one in the parameter space, thus speeding up the convergence to the optimal or a near-optimal point. Moreover, the parameters obtained after the learning process can be easily transformed into structure knowledge in the form of fuzzy if-then rules.

Fuzzy Classification have found its application in large range of problems like the standard IRIS problem.

### B. *Time Series*

Just like classification, Time Series Forecasting could also be done with the help of Neuro-Fuzzy Systems. Given a predicting sequence, the local context of the sequence is located in a series of the observed data. A distance measure taking the trend of the data into account is adopted. Proper lags of relevant variables for prediction are selected and training patterns are extracted. Based on the extracted training patterns, a set of TSK fuzzy rules are constructed automatically by an incremental clustering algorithm. The parameters involved in the fuzzy rules are then refined by the hybrid learning algorithm. The refined fuzzy rules constitute the predicting model and can then be used for prediction. Both direct and iterative forecasting models are developed. This approach has several advantages. It can produce adaptive forecasting models. It works for univariate and multivariate prediction. It also works for one-step as well as multi-step prediction. The neuro-fuzzy modeling scheme is adopted because it offers good properties such as non-linear learning capability, quick convergence, and high accuracy. Furthermore,the rules obtained are comprehensible to human beings.

According to a paper by A. Chaudari, a neuro-fuzzy time series forecasting model is developed to forecast the exchange rate of US Dollar to Indian Rupee. The model yields more accurate results with fewer observations and incomplete data sets for both point and interval forecasts. The empirical results indicate that performance of the model is comparatively better than other models which make it an ideal candidate for forecasting and decision making.

These were just 2 examples of indirect applications. The internet is flooded with the literature on applications of neuro-fuzzy system. The consumer market is one of the biggest field of applications.

The first applications of fuzzy neural networks to consumer products appeared on the (Japanese and Korean) market in 1991. Some examples include air conditioners, electric carpets, electric fans, electric thermo-pots, desktype electric heaters, microwave ovens, refrigerators, rice cookers, vacuum cleaner, washing machines, clothes driers, photocopying machines, and word processors. Neural networks are used to design membership functions of fuzzy systems that are employed as decision-making systems for controlling equipment. Although fuzzy logic can encode expert knowledge directly using rules with linguistic labels, it usually takes a lot of time to design and tune the membership functions which quantitatively define these linquistic labels. Neural network learning techniques can automate this process and substantially reduce development time and cost while improving performance.

Some of the less conventional but upcoming applications are

- Student Modelling
- Medical System
- Economics System
- Traffic Control
- Image Processing and Feature Extraction
- Forecasting and Predictions
- Manufacturing and System modelling
- Electrical and Electronic Systems
- NFS Enhancement

- Social Sciences
- Screening Systems (Jobs)

## VII. Simulation of ANFIS in MATLAB

MATLAB provides a toolbox to train ANFIS systems. We could select input-output, types of membership functions, edit rules etc. So to test the ANFIS approach we tried to evaluate the function

$$f(x, y) = \frac{sin(x + y)}{e^{xy}} \tag{22}$$

We created 4 systems. Each system takes 2 inputs and gives 1 output.

1) $FIS0$ has 2 fuzzy sets for each input. Input membership are of type bell functions.
2) $FIS1$ has 3 fuzzy sets for each input. Input membership are of type bell functions.
3) $FIS2$ has 5 fuzzy sets for each input. Input membership are of type bell functions.
4) $FIS3$ has 3 fuzzy sets for each input. Input membership are of type triangualar functions.

101 data points are generated for testing using random function. The training method used is hybrid training.

### A. Observations

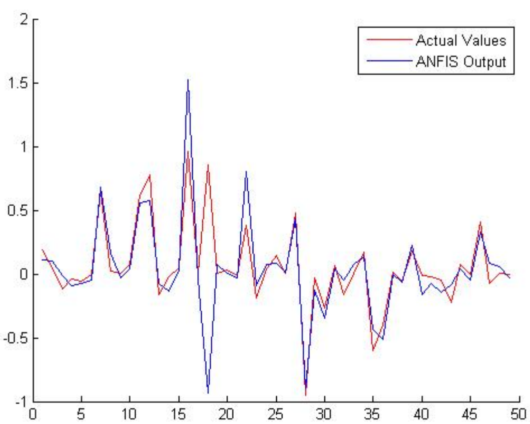


Fig. 7: FIS0 results

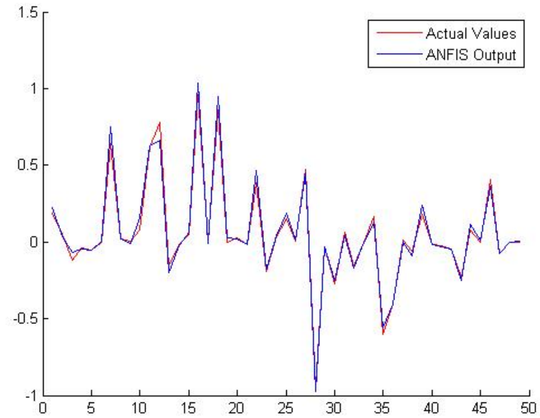


Fig. 8: FIS1 results

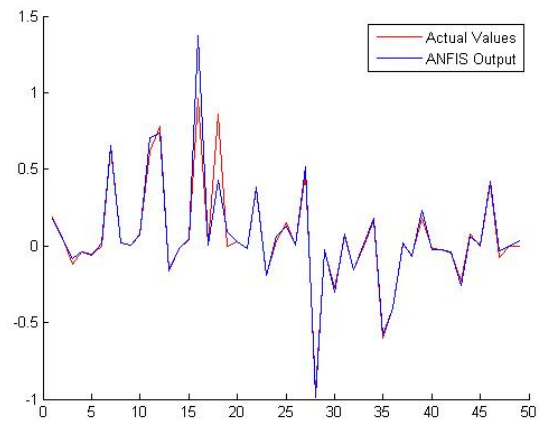


Fig. 9: FIS2 results

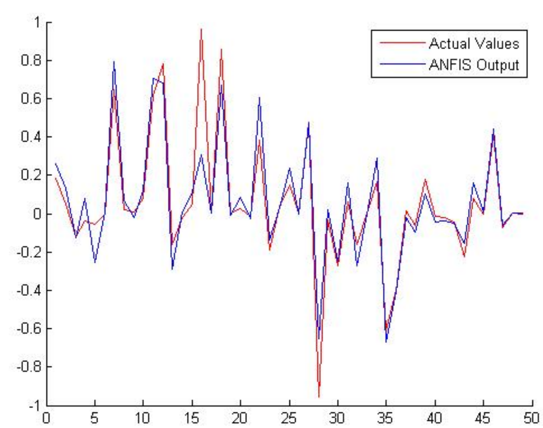


Fig. 10: FIS3 results

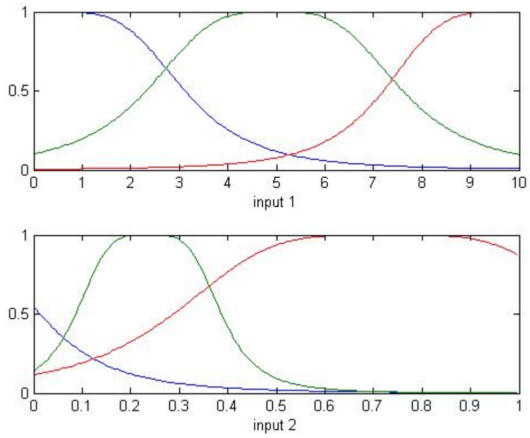


Fig. 11: FIS1 Membership Functions

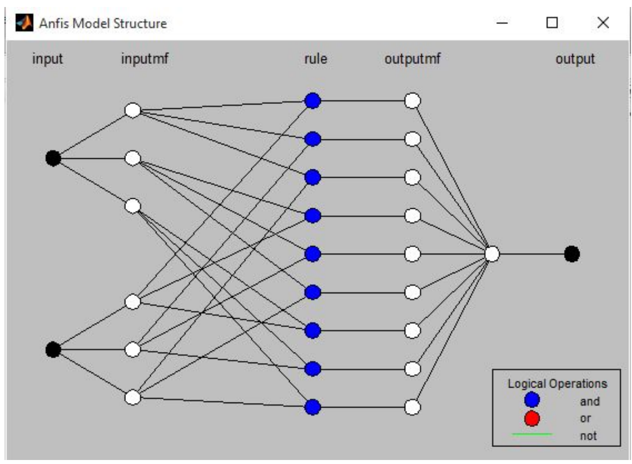


Fig. 12: FIS1 Network

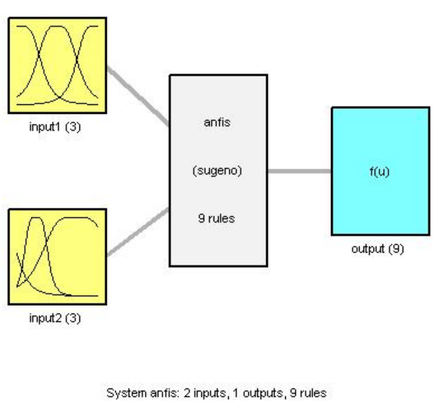


Fig. 13: FIS1 Structure

The error was found minimum for the FIS1. The results

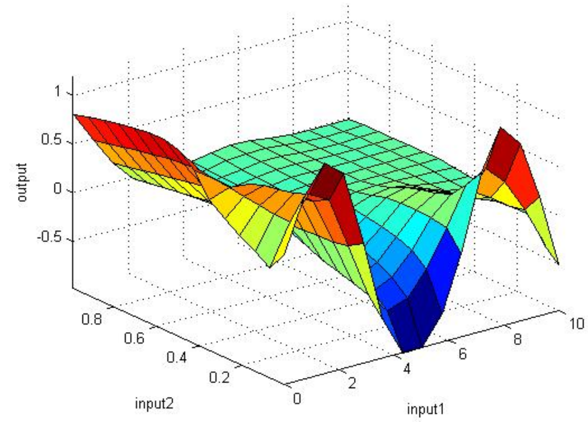


Fig. 14: FIS1 Surface

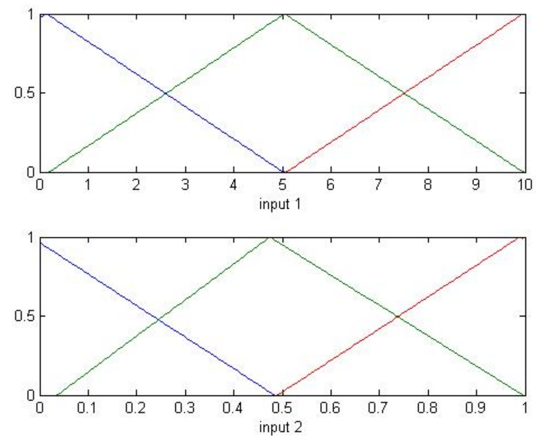


Fig. 15: FIS3 Membership Functions

were very accurate and were comparable to any other method used for fitting non linear functions. Moreover ANFIS systems provide added benefit of incorporating expert's analysis and understanding the rules which govern the output.

## VIII. Conclusion

In this paper we have studied existing literature on Fuzzy-Neuro Systems and their applications. We have found that they have a lot of applications in a wide domain of tasks. We also studied the mathematics behind the infrence models and how these systems are trained. Neuro-Fuzzy systems have become a core component of soft-computing hybrid systems and present a very good model which is not only accurate but also provide IF-THEN type of rules for interpretation. Neuro-Fuzzy Models would be very important if we want machines to be intelligent like humans.

## References

[1] Z. Sevarac, "Neuro fuzzy reasoner for student modeling," in *null*. IEEE, 2006, pp. 740–744.

[2] H. Takagi, "Application of neural networks and fuzzy logic to consumer products," in *Industrial Electronics, Control, Instrumentation, and Automation, 1992. Power Electronics and Motion Control., Proceedings of the 1992 International Conference on*. IEEE, 1992, pp. 1629–1633.

[3] "Neuro-fuzzy systems," http://fuzzy.cs.uni-magdeburg.de/nfdef.html.

[4] "Hybrid neuro-fuzzy systems," http://www.iaria.org/conferences2013/...Keynote_Negnevitsky.pdf.

[5] D. Nauck, F. Klawonn, and R. Kruse, *Foundations of neuro-fuzzy systems*. John Wiley & Sons, Inc., 1997.

[6] M. Nnegnevitsky, *Artificial Intelligence*.

[7] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, "Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence," 1997.

[8] R. Fullér, *Introduction to neuro-fuzzy systems*. Springer Science & Business Media, 2013, vol. 2.

[9] S. Kar, S. Das, and P. K. Ghosh, "Applications of neuro fuzzy systems: A brief review and future outline," *Applied Soft Computing*, vol. 15, pp. 243 – 259, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494613003487

[10] A. Chaudhuri, "Forecasting financial time series using multiple regression, multi layer perception, radial basis function and adaptive neuro fuzzy inference system models: A comparative analysis," *Computer and Information Science*, vol. 5, no. 6, p. 13, 2012.

[11] E. Czogala and J. Leski, *Fuzzy and neuro-fuzzy intelligent systems*. Physica, 2012, vol. 47.

[12] D. Nauck and R. Kruse, "A neuro-fuzzy method to learn fuzzy classification rules from data," *Fuzzy sets and Systems*, vol. 89, no. 3, pp. 277–288, 1997.

[13] J.-S. R. Jang, "Anfis: adaptive-network-based fuzzy inference system," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 23, no. 3, pp. 665–685, 1993.