Volume 4, Issue 3, March 2014 ISSN: 2277 128X
International Journal of Advanced Research in
Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Mathematical Modelling Based on Neural Network with Fuzzy Logic

M. Premalatha *
Research scholar, Department of Mathematics,
Sathyabama University, Chennai, Tamil Nadu, India

C.VijayaLakshmi
School of Advance Science, Depart of Mathematics
VIT University, Chennai, Tamil Nadu, India

Abstract*— *A Mathematical modelling in intelligent systems based on various combinations of neural nets and fuzzy logic, called Neural Fuzzy Systems (NFS). The rationale to combine fuzzy logic with neural nets is emphasized to the limitations of each of these technologies while adding their advantages. In this paper presented elegant algorithms to combine neural nets with fuzzy logic, resulting in feed forward systems. A system whose principal function is not computational but is controlled by a computational system embedded within it referred to as a computational embedded system. An objective of Fuzzy logic has been to make computers think like people. A fuzzy logic is particularly useful in the control of highly nonlinear processes difficult to control, by conventional methods. The objective of this paper idea is to use the knowledge of an expert to create the model. The conventional analysis techniques of control systems are based on mathematical models and hence cannot be used for the analysis of a fuzzy logic control in embedded system.

Keywords*— *Neural Network and Fuzzy Logic, Courier Service-Application, Fuzzy Logic and Embedded System, Fuzzy Logic Controller, Applications

I. INTRODUCTION

Networks and Fuzzy Logic are the two key technologies that have recently received growing attention in solving real world, nonlinear, time variant problems. Intelligent combinations of these two technologies can exploit their advantages while eliminating their limitations. Such combinations of neural networks and fuzzy logic are called Neural Fuzzy Systems (NFS). Intelligent Systems (IS) based on neural fuzzy techniques have shown good potential to solve many complex real word problems. The behaviour of the trained ANN depends on the weights, which are also referred to as strengths of the connections between the PEs. Artificial neural networks are being applied to a wide variety of automation problems including adaptive control, optimization, medical diagnosis, decision making, as well as information and signal processing, including speech processing. An Embedded system is a system that has software embedded into computer-hardware, which makes a system dedicated for an application (s) or specific part of an application or product or part of a larger system. Embedded systems are the electronic systems that contain a microprocessor or a microcontroller, but we do not think of them as computer the computer is hidden or embedded into the system [1] [8]

II. NEURAL NETWORK AND FUZZY LOGIC

In neural networks, as in Machine Learning, the excitement of technological progress is supplemented by the challenge of reproducing intelligence itself. Neural network approaches combine the complexity of some of the statistical techniques with the machine learning objective of imitating human intelligence: however, this is done at a more "unconscious" level and hence there is no accompanying ability to make learned concepts transparent to the user.

Fuzzy logic uses graded statements rather than ones that are strictly true or false. Fuzzy logic techniques have been successfully applied in a number of applications: computer vision, decision making, and system design including ANN training. The most extensive use of fuzzy logic is in the area of control, where examples include controllers for cement kilns, braking systems, elevators, washing machines, hot water heaters, air-conditioners, video cameras [9].

A. *Fusion*

Some of these techniques are fused as:

- Neural networks for designing fuzzy systems
- Fuzzy systems for designing neural networks
- Evolutionary computing for the design of fuzzy systems
- Evolutionary computing in automatically training and generating neural network architectures [6].

B. *Value of a Fuzzy Formula*

The value of a fuzzy formula is uniquely defined using the following rules:

1. ¼(A) = 0, if A = 0
2. ¼(A) = 1, if A = 1
3. ¼(A) = ¼(x), if A = 1
4. ¼(A) = 1 - ¼(B), if $A = \bar{B}$

5. ¼(A) = *min* (¼(B), ¼(C)), if $A = B - C$
6. ¼(A) = *max* (¼(B), ¼(C)), if $A = B + C$

III. Mathematical model of Fuzzy Logic Controller (FLC)

The main building units of an FLC are a fuzzification unit, a fuzzy logic reasoning unit, a knowledge base, Defuzzification unit.

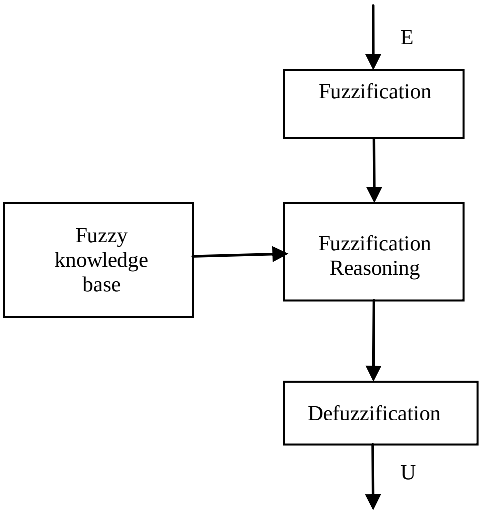

Fig.1 Basic Structure in Fuzzy Logic

The fuzzy knowledge base has a rule that maps fuzzy input variables, E in to a fuzzy output U.
E→U (condition E implies condition U)
Fuzzy reasoning is used to the output contributed from each rule [8] [9].

Input layer: No computation is done in this layer. Each node in this layer, which corresponds to one input variable, only transmits input value to the next layer directly. The link weight in layer 1 is unit. **Fuzzification Layer**: Each node in this layer corresponds to one label to one output links represent the membership values, which specifies the degree to this an input value belong fuzzy set, is calculated in layer 2.[11] [12]

A. Mathematical Modelling of Neural Network

In most cases we assume that each unit provides an additive contribution to the input of the unit with which it is connected. The total input to unit k is simply the weighted sum of the separate outputs from each of the connected units plus a bias or off set term θ_k

$$S_k(t) = \sum_j w_{jk}(t) y_j(t) + \theta_k \tag{1}$$

The contribution for positive w_{jk} is considered as an excitation and for negative w_{jk} as inhibition. In some cases more complex rules for combining inputs are used, in which a distinction is made between excitatory and inhibitory inputs. We call units with propagation rule (1) sigma units. A different propagation rule is known as the propagation rule for the sigma-pi unit.

$$S_k(t) = \sum_j w_{jk}(t) \prod_m y_{jm}(t) + \theta_k(t) \tag{2}$$

Often, the y_{jm} are weighted before multiplication.
The activation function is a non-decreasing function of the total input of the unit:

$$y_k(t+1) = A_k(S_k(t)) = A_k\left(\sum_j w_{jk}(t) y_j(t) + \theta_k(t)\right) \tag{3}$$

For this smoothly function often a sigmoid (S shaped) function like

$$y_k(t) = A(s_k) = \frac{1}{1 + e^{-s_k}} \tag{4}$$

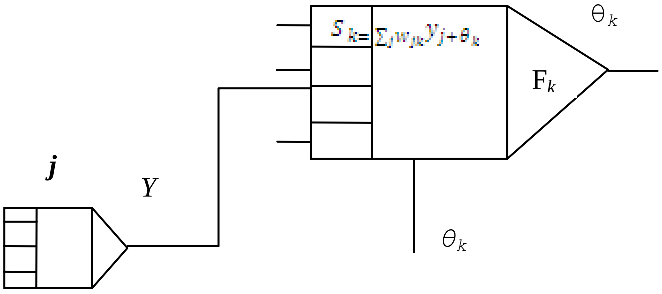

Fig.2 Mathematical Model of NN with Weighted vector

The activation is not deterministically determined by the neuron input, but the neuron input determines the probability p that a neuron gets a high activation value:

$$p(y_k \leftarrow 1) = \frac{1}{1+e^{-s_k/T}} \tag{5}$$

B. *Analysis*

A fuzzy logic is particularly useful in the control of highly nonlinear processes difficult to control, by conventional methods. The central idea is to use the knowledge of an expert to create the model. The conventional analysis techniques of control systems are based on mathematical models and hence cannot be used for the analysis of a fuzzy logic control system. [11] [14]

Energetistic Stability Method:

The energetistic stability method involves time-consuming calculations, it provides an intuitive method to analyse the stability of a fuzzy dynamic system [3]. It is based on the general physical law that any dynamic system is stable if its total energy decreases with time until it reaches a steady state.

$X_{n+1} = X_n \, . \, U_n \, . \, R$, n = 1, 2,3,… (6)

Where X_n and X_{n+1} are fuzzy sets at the nth and (n+1)th times instants, R is a fuzzy relation. If there is no input, the system is said to be free. This occurs when U_n = Zero for all values of n, where zero is a fuzzy singleton.

Let $S = U_n \, . \, R$ then $X_{n+1} = X_n \, . \, S$, k = 1, 2, 3,…. (7)

Equilibrium state:

$$\left.\begin{array}{l} X_{n+1} = X_n = X_l \text{ for all } k \\ X_l = X_l \, . S \end{array}\right\} \tag{8}$$

Energy of fuzzy relation

$$E(S) = \frac{1}{r.c} \sum_{i=1}^{r} \sum_{j=1}^{c} w(x_i, y_j) . f(\mu_s(x_i, y_j))$$

$w(x_i, y_j)$ = Cartesian product ith and jth element in the support of the fuzzy set

$f(\mu_s(x_i, y_j))$ =

Corresponding values of the membership in the normalized fuzzy relatio matrix

Ms.

Energy expression for free fuzzy dynamic system

$E(X_{n+1}) = E(X_n \, . \, S)$

$X_1 = X_0 \, . \, S$

$X_2 = X_1 \, . \, S = X_0 \, . \, S \, . \, S = X_0 \, . \, S^2$

$X_3 = X_2 \, . \, S = X_1 . \, S \, . \, S = X_0 \, .S \, .S \, .S = X_0 \, . \, S^3$

$X_n = X_0 \, . \, S^n$

X_0 = Initial fuzzy system

$E(X_{n+1}) = E(X_n \, . \, S) = E\,(X_0 \, . \, S^{n+1})$

The change in energy ΛE between two consecutive states

$\Lambda E = E\,(X_0 \, . \, S^n) - E\,(X_0 \, . \, S^{n-1})$. Since $E(X_0)$ is a constant. ΛE does not depend on initial conditions. The Rational Characteristic Energy Function

$\Lambda E_{\text{Characteristic Energy Function}}(S, n) = E(S^n) - E(S^{n-1})$

To determine the stability used this expression as follows: The system is stable if $\Lambda E_{\text{Characteristic Energy Function}}(S, n) \leq 0$ for n→∞

The system us unstable if $\Lambda E_{\text{Characteristic Energy Function}}(S, n) \geq 0$ for n→∞

The system is oscillatory with a period T if

$\left| \Lambda E_{\text{Characteristic Energy Function}}(S, n) \right| = \left| \Lambda E_{\text{Characteristic Energy Function}}(S, n+T) \right|$ for n→∞

Determine the Stability of the system if the fuzzy relation S = $\begin{bmatrix} 0.1 & 0.8 & 0.9 \\ 0.7 & 0.6 & 0.5 \\ 0.3 & 0.1 & 0.2 \end{bmatrix}$

$$\begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{bmatrix} \cdot \begin{bmatrix} \beta_{11} & \beta_{12} & \alpha_{13} \\ \beta_{21} & \beta_{22} & \alpha_{23} \\ \beta_{31} & \alpha_{32} & \alpha_{33} \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \end{bmatrix}$$

With γ_{11} = Max [min (α_{11}, β_{11}), min(α_{12}, β_{21}), min(α_{31}, β_{31})]

$S^2 = \begin{bmatrix} 0.7 & 0.6 & 0.5 \\ 0.6 & 0.7 & 0.7 \\ 0.2 & 0.3 & 0.3 \end{bmatrix}$ $S^3 = \begin{bmatrix} 0.6 & 0.7 & 0.7 \\ 0.7 & 0.6 & 0.6 \\ 0.3 & 0.3 & 0.3 \end{bmatrix}$ $S^4 = \begin{bmatrix} 0.7 & 0.6 & 0.6 \\ 0.6 & 0.7 & 0.7 \\ 0.3 & 0.3 & 0.3 \end{bmatrix}$

$S^5 = \begin{bmatrix} 0.6 & 0.7 & 0.7 \\ 0.7 & 0.6 & 0.6 \\ 0.3 & 0.3 & 0.3 \end{bmatrix}$ $S^6 = \begin{bmatrix} 0.7 & 0.6 & 0.6 \\ 0.6 & 0.7 & 0.7 \\ 0.3 & 0.3 & 0.3 \end{bmatrix}$

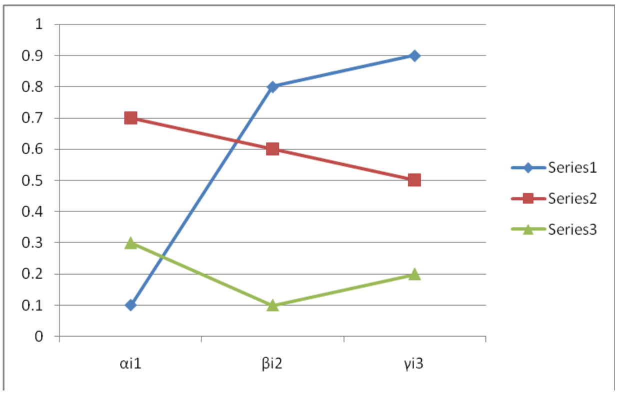

Fig.3 Values of Energy Stability of the system

By the Equation of the energy
E(S), E(S^2), E(S^3),E(S^4), ….
E(S) = $\frac{1}{9}\{1 \times 1 \times 0.1 + 1 \times 2 \times 0.8 + 1 \times 3 \times 0.9 + 2 \times 1 \times 0.7 + 2 \times 2 \times 0.6 + 2 \times 3 \times 0.5 + 3 \times 1 \times 0.3 + 3 \times 2 \times 0.1 + 3 \times 3 \times 0.2\}$.

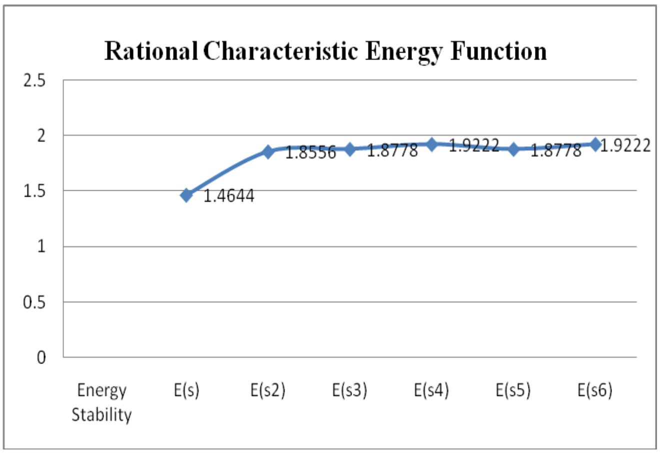

Fig.4 Energy Stability oscillation

E(S) = $\frac{1}{9}(13.18) = 1.4644$
E (S^2) = $\frac{1}{9}(16.7) = 1.8556$
E (S^3) = $\frac{1}{9}(16.9) = 1.8778$
E (S^4) = $\frac{1}{9}(17.3) = 1.9222$
E (S^5) = $\frac{1}{9}(16.9) = 1.8778$
E (S^6) = $\frac{1}{9}(17.3) = 1.9222$

We conclude that the system is oscillatory.

IV. FUZZY LOGIC CONTROL APPLICATION

A. System Design Process: Automatic Chocolate Vending Machine (ACVM)

Let us consider an automatic chocolate vending machine.

Assume that ACVM has following components:

It has keypad on the top of the machine. That enables a child to interact with it when buying a chocolate. The owner can also command and interact with the machine.

It has an LCD display unit on the top of the machine. It displays menus, text entered into the ACVM and pictograms, welcome, thank you and other messages. It enables the child as well as the ACVM owner to graphically interact with the machine. It also displays time and date.

It has a coin insertion slot and a mechanical coin sorter so that child can insert coins to buy a chocolate.

It has a delivery slot so that child can collect the chocolate and coins, if refunded.

It has an internet connection port using a USB based wireless modem so that owner can know status of the ACVM sales from ma remote location.

ACVM Functions:

The ACVM displays the GUIs and if the child wishes to enter contact information, birthday information or get answer to FAQs, it displays the appropriate menu.

It displays a welcome message when in idle state. IT also continuously displays time and date at the right bottom corner of display screen. It can also intermittently display news, weather data or advertisement or important information of interest during idle state.

When first coin is inserted, a timer also starts. The child is expected to insert all required coins in 2 minutes.

After 2 minutes the ACVM will display a query to the child doest insert sufficient coins. If the query is not answered the coins are refunded.

Within 2 minutes if sufficient coins are collected, it displays the message, 'Thanks, wait for the few moments please!' delivers the chocolate through the delivery slot and displays message, 'Collect the chocolate and visit again, please!'

Hardware Units:

Microcontroller or ASIP (Application Specific Instruction Set Processor).

RAM for sorting temporary variables and stack.

Rom for application codes and RTOS codes for scheduling the tasks.

Flash memory for storing user preferences, contact data, user address, user date of birth, user identification code, answers of FAQs.

Timer and interrupt controller.

Power supply.

Software components:

Keypad input read task, Display task, Read coin task for finding coins sorted, Deliver chocolate task, TCP/IP stack processing task, TCP/IP stack communication task [16].

B. Neural Network Application for Assignment Problem in LPP for' n' values

The courier service application, the neural network model is used to dispatch new requests to appropriate vehicles. Using a set of examples composed of dispatching situations where the vehicle selected by the dispatcher is known, the neural network model is trained to assign high evaluations to these vehicles. Ideally, for each dispatching situation, the vehicle selected by the dispatcher should be ranked first by the neural network (i.e., it should get the highest evaluation among all available vehicles). **Using 120 requests collected from a courier service company operating a fleet of 9 vehicles in the Montreal area. After training the network with 70 requests, the model was tested on the 50 remaining requests. In 47 cases out of 50, the vehicle selected by the dispatcher was ranked first, second, third, or fourth by the neural network (out of nine possible ranks).** When the vehicle chosen by the expert did not get the highest evaluation, it was often quite close. Furthermore, the vehicle with the highest evaluation was typically a good alternative choice. **In this work, the network was trained in batch mode from previously collected data. However, it would be possible to "link" the neural network model with the dispatcher during one or more operations per day in order to adjust the connection weights in real time after each new decision.** At the end of the training period, the network would then be able to mimic the dispatcher's behaviour. Such a network could be used, for example, to facilitate the dispatcher's task by displaying the neural network evaluation of the best vehicles or to assist less experienced dispatchers when confronted with difficult situations. In each case, it is assumed that the final decision is left to the human expert. Clearly, vehicles with high evaluations are the best candidates for servicing a new request [10] [12] [17].

C. Real time Operating System

Harmony (National Research Council of Canada)

Harmony is a multitasking, multiprocessing operating system for real-time control. It is developed at the National Research Council to serve the need for a flexible system for real-time control of robotics experiments and for other applications of embedded systems where predictable temporal performance is a requirement. Harmony is scalable, configurable and portable, both across different target computers, and across different development hosts [11].

V. CONCLUSION

The application of fuzzy logic principles enable solving problems that would have been difficult or even impossible to solve it mathematical models were used. Fuzzy logic, would not create a solution is one did not exists in the first place through the knowledge of an expert, such knowledge is the basis of fuzzy algorithm. Next generation real-time operating systems would demand new operating systems and task designs to support predictability and high degree of adaptability. In neural networks, as in Machine Learning, the excitement of technological progress is supplemented by the challenge of reproducing intelligence itself. Expressing the weights of the neural net using fuzzy rules helps provide greater insights into the neural nets, thus leading to a design of better neural nets. Neural Fuzzy System's learning and generalization capabilities allow generated rules and membership functions to provide more reliable and accurate solutions than alternative methods. However, with the proper combination of fuzzy logic and neural networks it is possible to completely map (100%) the neural net knowledge to fuzzy logic. This enables users to generate fuzzy logic solutions that meet a pre-specified accuracy of outputs. This is possible because the neural net is able to learn to a pre-specified accuracy, especially for the training set and learned knowledge can be fully mapped to fuzzy logic.

REFERENCES:

[1] "Internet application with Fuzzy Logic", A SURVEY BY H.CHRIS TSENG. Vol I 2007.

[2] "Fuzzy Logic and the measures of certainty in E-commerce expert system", Louis sullvan, 2001.

[3] International Journal of Soft Computing—Special Issue in FLINT, 2002.

[4] W.Banks and G.Hayward, Fuzzy Logic in embedded Microcomputers and control Systems, Byle Craft Limited, Waterloo, Ontario, 2001.

[5] R.E. Filman "Embedded Internet Systems Corne Home, "IEEE Internet Computing, 52-53, 2001.

[6] F. Vahid and T.Givargis Embedded System Design, John Wiley & Sons Inc., 2002.

[7] T. Wilmshurst, An Introduction to the Design of Small-scale Embedded Systems, Palgrove, Newyork,2001

[8] Xu, L., 2002, BYY learning, regularized implementation, and model selection on modular networks with one hidden layer of binary units, Neurocomputing, in press.

[9] Abraham and M.R. Khan, Neuro-Fuzzy Paradigms for Intelligent Energy Management, Innovations in Intelligent Systems: Design, Management and Applications, Abraham A., Jain L. and Jan van der Zwaag B. (Eds.), Studies in Fuzziness and Soft Computing, Springer Verlag Germany, Chap. 12, pp. 285– 314, 2003.

[10] X.H. Li and C.L.P. Chen, The Equivalence between Fuzzy Logic Systems and Feed forward Neural Networks, IEEE Transactions on Neural Networks, Vol 11, No. 2, pp. 356–365, 2000.

[11] "Neural Network, Fuzzy Logic, and Genetic Algorithm – Synthesis and Applications", by S. Rajasekaran and G.A. Vijayalakshmi Pai, (2005), Prentice Hall Chapter 3, pp.34-86.

[12] "Soft Computing and Intelligent Systems Design – Theory, Tools and Applications", by Fakhreddine Karray and Clarence de Silva (2004), Addison Wesley, Chapter 5, pp. 249 – 293.

[13] "Introduction to Fuzzy Sets and Fuzzy Logic", by M. Ganesh, (2008), Prentice Hall, pp. 1 – 256.

[14] "Fuzzy Logic with Engineering Applications", by Timothy Ross, (2004), John Wiley & Sons Inc, pp. 1-623.

[15] W. Wolf, Computers as Components: Principles of Embedded Computing System Design, Morgan Kaufmann Publishers, San Francisco, 2001.

[16] L.A.Zadeh, "What is Fuzzy Logic and what is its application?" Scientific Computing Seminar, NERSC, 2002.

[17] M.Premalatha, C. Vijayalakshmi (2012), Analysis of Soft Computing in Neural Network, in Proc 2nd International Conference in Computer Applications'12, Associated with ASDF, ACM, SERSC, Pondicherry, Vol 2, PP.no.172-177