Comparison of Functional Performance of Radial Basis Function Network, Fuzzy Logic System and Adaptive Neuro-Fuzzy Systems

Madhukar S.Chavan, M.Tech.[1], Dr. Rajan H.Chile, Ph.D.[2] and Dr. Shashikant R.Sawant Ph.D.[3]

1. Department of Electronics and Telecommunication Engineering, P. V. P. I. T. Budhgaon, Sangli. Phone no. 9881062882,
2. S.G.G.S. Institute of Engineering and Technology, Vishnupuri, Nanded.
3. Department of Science and Technology Shivaji University Kolhapur.

ABSTRACT

Neuro-fuzzy hybridization results in a hybrid intelligent system that synergizes these two techniques by combining the human-like reasoning style of fuzzy systems with the learning and connectionist structure of neural networks. Neuro-fuzzy hybridization is widely termed as Fuzzy Neural Network (FNN) or Neuro-Fuzzy System (NFS) in the literature. Neuro-fuzzy system (the more popular term is used henceforth) incorporates the human-like reasoning style of fuzzy systems through the use of fuzzy sets and a linguistic model consisting of a set of IF-THEN fuzzy rules. The main strength of neuro-fuzzy systems is that they are universal approximators with the ability to solicit interpretable IF-THEN rules. This paper compares performance of three models: Radial Basis function network (RBFN), Fuzzy logic systems and Adaptive neuro-fuzzy inference system (ANFIS). These models are from different origins but they share some common characteristics. We have shown that under some minor restrictions, they show equality in their performance. The comparison results show that the ANFIS model have better performances.

Key words: Radial Basis function network, Fuzzy logic system, Adaptive neuro-fuzzy inference system, Fuzzification.

Corresponding Author: Madhukar S. Chavan

1. INTRODUCTION

The RBF network consists of a feed forward architecture with an input layer, a hidden layer of RBF "pattern" units and an output layer of linear units. The input layer simply transfers the input vector to the hidden units, which form a localized response to the input pattern. The activation levels of the output units provide an indication of the nearness of the input vector to the classes. Learning is normally undertaken as a two-stage process. The fuzzy logic provides an inference structure that enables appropriate human reasoning capabilities [1], [10]. It is a mathematical tool for dealing with uncertainty. Fuzzy logic also provides an inference structure that enables the human reasoning capabilities to be applied to artificial knowledge-based systems. It provides a means for converting linguistic variables into control actions and thus offers a high-level of computation. Compared to neural networks, the neuro-fuzzy methods provide models which can be interpreted by human beings. The models are in the form of the familiar if-then rules, implying easy integration with expert rules. By using a hybrid learning procedure, the ANFIS can construct an input-output mapping based on both human knowledge (in the form of fuzzy if-then rules) and stipulated input-output data pairs. Radial basis function (RBF) networks and a simplified class of fuzzy systems are functionally equivalent under some mild conditions. This functional equivalence has made it possible to combine the features of these two systems, which has been developed into a powerful type of neuro-fuzzy systems [5]. In this paper, performance of Radial basis function network, Fuzzy logic system and Adaptive neuro-fuzzy systems is compared. The comparison results show that the ANFIS model have better performances.

2. RADIAL BASIS FUNCTION NETWORKS

The RBF networks have very attractive properties such as localization, functional approximation, interpolation, cluster modeling and quasi-orthogonality. These properties made them attractive in many applications. Very different fields such as: Telecommunications, Signal processing, image processing, control engineering and computer vision used them successfully for various tasks. The radial basis function (RBF) network is a feed-forward neural network with a single layer of hidden units. Radial basis function networks are different from back propagation networks because they have only one layer of hidden units, and do not use the sigmoid activation function in the hidden layer unit. Instead, the radial basis function network has fixed-feature detectors in the hidden layer which use a specified basis function to detect and respond to localized portions of the input vector space. One advantage of radial basis networks over back propagation is that, if the input signal is non-stationary, the localized nature of the hidden layer response makes the networks less susceptible to “memory loss,” or, as some would say, “Weight loss.” Radial Basis Function Networks became very popular due to several important advantages over traditional multilayer perceptrons. Locality of radial basis functions and feature extraction in hidden neurons that allows usage of clustering algorithms and independent tuning of RBFN parameters.The radial basis network (RBFN) is shown in the Fig 1. has three layers. Each hidden unit implements a radial activated function. The output units implement a weighted sum of hidden unit outputs. The input to RBF network is nonlinear while the output is linear.

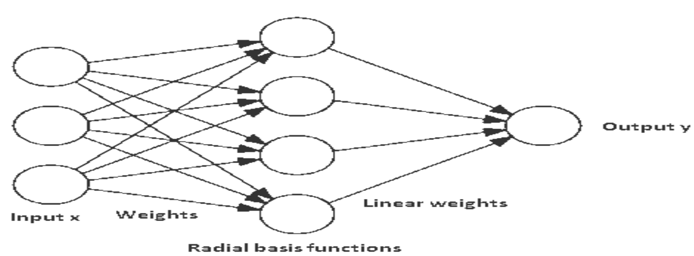

Fig 1: Architecture of RBFN

Let $\vec{x}$ is an N-dimensional input vector to all radial basis functions, each with different parameters. The output of the network is a linear combination of the outputs from radial basis functions [10], [11]. The output of i^th receptive field unit (or hidden unit) is

$$w_i = R_i(\vec{x}) = R_i(\|\vec{x} - \vec{c_i}\|/\sigma_i) \text{ , i =1, 2 , …, H} \quad \text{..... (i)}$$

Where $\vec{c_i}$ is a vector with the same dimension as $\vec{x}$., H is the number of receptive field units, and R_i (·) is the i^{th} receptive field response with a single maximum at the origin.
Typically, R_i (·) is chosen as a Gaussian function.

$$R_i(\vec{x}) = \exp\left[-\frac{\|\vec{x}-\vec{c_i}\|^2}{\sigma_i{}^2}\right] \quad \text{.... (ii)}$$

The radial basis function w_i computed by the i^{th} hidden units is maximum when the input vector $\vec{x}$ is near the centre $\vec{c_i}$ of that unit. The output of an RBFN can be computed in two ways. The output is the weighted sum of the function value associated with each receptive field.

$$f(\vec{x}) = \sum_{i=1}^{H} f_i w_i = \sum_{i=1}^{H} f_i R_i(\vec{x}) \quad \text{..... (iii)}$$

Where, f_i is the function value or strength of i^{th} receptive field. With the addition of lateral connections (not shown in Fig 1) between the receptive field units, the network can produce the normalized response function as the weighted average of the strengths.

$$f(\vec{x}) - \frac{\sum_{i=1}^{H} f_i w_i}{\sum_{i=1}^{H} w_i} = \frac{\sum_{i=1}^{H} f_i R_i(\vec{x})}{\sum_{i=1}^{H} R_i(\vec{x})} \qquad \text{..... (iv)}$$

Several learning algorithms have been proposed to identify the parameters $\vec{c_i}$, σ_i and f_i. The basic aim of every algorithm is to reduce the square errors between desired output and model output.

3. FUZZY LOGIC SYSTEM

The Fuzzy Logic is a mathematical tool for dealing with uncertainty. It offers to a soft computing partnership the important concept of computing with words'. It provides a technique to deal with imprecision and information granularity. The fuzzy theory provides a mechanism for representing linguistic constructs such as “many,” “low,” “medium,” “often,” “few.” In general, the fuzzy logic provides an inference structure that enables appropriate human reasoning capabilities. On the contrary, the traditional binary set theory describes crisp events, events that either do or do not occur. It uses probability theory to explain if an event will occur, measuring the chance with which a given event is expected to occur. The utility of fuzzy sets lies in their ability to model uncertain or ambiguous data, so often encountered in real life. The general observations about fuzzy logic are:

- Fuzzy logic is conceptually easy to understand: The mathematical concepts behind fuzzy reasoning are very simple. What makes fuzzy nice is the “naturalness” of its approach and not its far-reaching complexity.
- Fuzzy logic is flexible: With any given system, it’s easy to massage it or layer more functionality on top of it without starting again from scratch.
- Fuzzy logic is tolerant of imprecise data: Everything is imprecise if you look closely enough, but more than that, most things are imprecise even on careful inspection. Fuzzy reasoning builds this understanding into the process rather than tacking it onto the end.
- Fuzzy logic can model nonlinear functions of arbitrary complexity: You can create a fuzzy system to match any set of input-output data. This process is made particularly easy by adaptive techniques like ANFIS (Adaptive Neuro-Fuzzy Inference Systems), which are available in the Fuzzy Logic Toolbox.
- Fuzzy logic can be built on top of the experience of experts: In direct contrast to neural networks, which take training data and generate opaque, impenetrable models, fuzzy logic lets you rely on the experience of people who already understand your system.
- Fuzzy logic can be blended with conventional control techniques: Fuzzy systems don’t necessarily replace conventional control methods. In many cases fuzzy systems augment them and simplify their implementation.
- Fuzzy logic is based on natural language: The basis for fuzzy logic is the basis for human communication. This observation underpins many of the other statements about fuzzy logic.

4. FUZZY INFERENCE SYSTEM

Fuzzy logic provides an inference structure that enables the human reasoning capabilities to be applied to artificial knowledge-based systems. Fuzzy logic provides a means for converting linguistic strategy into control actions and thus offers a high-level computation. Fuzzy logic provides mathematical strength to the emulation of certain perceptual and linguistic attributes associated with human cognition. Fuzzy logic is aimed at a formalization of modes of reasoning which are approximate rather than exact.

Fuzzification: Fuzzification is an important concept in the fuzzy logic theory. Fuzzification is the process where the crisp quantities are converted to fuzzy (crisp to fuzzy).By identifying some of the uncertainties present in the crisp values, we form the fuzzy values. The conversion of fuzzy values is represented by the membership functions .In any practical applications, in industries, etc., measurement of voltage, current, temperature, etc., there might be a negligible error. This causes imprecision in the data. This imprecision can be represented by the membership functions. Hence fuzzification is performed. Thus fuzzification process may involve assigning membership values for the given crisp quantities. For the fuzzification process it is necessary to assign membership values for the given crisp quantities there are following methods for membership value assignments: Intuition, Inference, Rank ordering, Angular fuzzy sets, neural networks, Genetic algorithms, and Inductive reasoning.

Defuzzification: Defuzzification means the fuzzy to crisp conversions. The fuzzy results generated cannot be used as such to the applications, hence it is necessary to convert the fuzzy quantities into crisp quantities for further processing. This can be achieved by using defuzzification process. The defuzzification has the capability to reduce a fuzzy to a crisp single-valued quantity or as a set, or converting to the form in which fuzzy quantity is present. Defuzzification can also be called as "rounding off" method. Defuzzification reduces the collection of membership function values in to a single sealer quantity. Different defuzzification methods are:

1) Lambda cuts for fuzzy sets, 2) Lambda cuts for fuzzy relations, 3) Max membership principle, 4) Weighted average method, 5) Mean max membership, 6) Centre of largest area, 7) First of maxima or Last of maxima.

Fuzzy rule base: Rules form the basis for the fuzzy logic to obtain the fuzzy output. The rule based system is different from the expert system in the manner that the rules comprising the rule-based system originate from sources other than that of human experts and hence are different from expert systems. The rule-based form uses linguistic variables as its antecedents and consequents. The antecedents express an inference or the inequality, which should be satisfied. The consequents are those, which we can infer, and is the output if the antecedent in equality is satisfied. The fuzzy rule-based system uses IF–THEN rule-based system, given by, IF antecedent, THEN consequent. The formation of rules is in general the canonical rule formation. For any linguistic variable, there are three general forms in which the canonical rules can be formed. They are: Assignment statements, Conditional statements, and Unconditional statements. The fuzzy rule-based system may involve more than one rule. The process of obtaining the overall conclusion from the individually mentioned consequents contributed by each rule in the fuzzy rule this is known as aggregation of rule. There are two methods for determining the aggregation of rules: 1) Conjunctive system of rules, 2) Disjunctive system of rules. Fuzzy inference is the actual process of mapping from a given input to an output using fuzzy logic [1]. The process involves membership functions, fuzzy logic operators, and if-then rules. No standard methods exist for transforming human knowledge or experience into the rule base and database of a fuzzy inference system. There is a need for effective methods for tuning the membership functions (MF's) so as to minimize the output error measure or maximize performance index. Fuzzy inference systems are also known as fuzzy-rule-based systems or fuzzy models. Fuzzy inference system can be classified as:

Type 1: The overall output is the weighted average of each rule's crisp output induced by the rule's firing strength and output membership functions. The output membership functions used in this scheme must be monotonic functions.

Type2: The overall fuzzy output is derived by applying "max" operation to the qualified fuzzy outputs (each of which is equal to the minimum of firing strength and the output membership function of each rule). Various schemes have been proposed to choose the final crisp output based on the overall fuzzy output; some of them are centroid of area, bisector of area, mean of maxima, maximum criterion etc.

Type3: Takagi and Sugeno's fuzzy if-then rules are used. The output of each rule is a linear combination of input variables plus a constant term, and the final output is the weighted average of each rule's output.

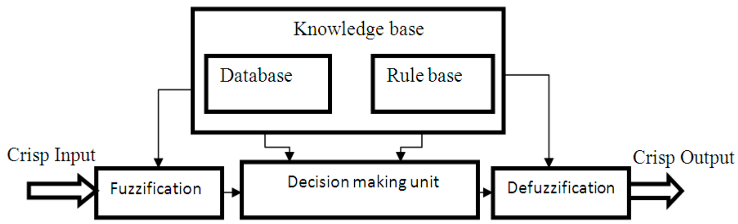

Fig 2: Fuzzy inference system

Fuzzy inference system is composed of five functional blocks: A rule base contains a number of fuzzy if-then rules. A database defines the membership functions of the fuzzy sets used in the fuzzy rules. A decision-making unit performs the inference operation on rules. A fuzzification interface transforms the crisp inputs into degree of match with linguistic values. A defuzzification interface transforms the fuzzy results of the inference into a crisp output. Usually the rule base and the database are jointly referred to as the knowledge base. The steps of *fuzzy reasoning* performed by fuzzy inference systems are:

- Compare the input variables with the membership functions on the premise part to obtain the membership values of each linguistic label. (This stage is often called as fuzzification).
- Combine (through a specific T-norm operator, usually multiplication or min.) the membership values on the premise part to get firing strength (weight) of each rule.
- Generate the qualified consequent (either fuzzy or crisp) of each rule depending on the firing strength.
- Aggregate the qualified consequents to produce a crisp output. The overall output can be chosen either as the weighted sum of each rule's output given as: $f(\vec{x}) = \sum_{i=1}^{R} w_i f_i$ or the weighted average of each rule's output given as:

$$f(\vec{x}) = \frac{\sum_{i=1}^{R} w_i f_i}{\sum_{i=1}^{R} w_i} \quad \text{.... (v)}$$

Where, R is number of fuzzy if-then rules.

5. ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS)

In the field of artificial intelligence, neuro-fuzzy refers to combinations of artificial neural networks and fuzzy logic. The neuro-adaptive learning techniques provide a method for the fuzzy modeling procedure to learn information about a data set, in order to compute the membership function parameters that best allow the associated fuzzy inference system to track the given input/output data [10]. Using a given input/output data set, the toolbox function ANFIS constructs a fuzzy inference system (FIS) whose membership function parameters are tuned (adjusted) using either a back propagation algorithm alone, or in combination with a least squares method. Assume that the fuzzy inference system has two inputs (x and y) and one output f. For a first-order Sugeno fuzzy model, a common rule set with two fuzzy if-then rules is as follows:

Rule 1: If x_1 is A_1 and x_2 is B_1.then $f_1 = a_1 x_1 + b_1 x_2 + c_1$.
Rule 2: If x_1 is A_2 and x_2 is B_2.then $f_2 = a_2 x_1 + b_2 x_2 + c_2$.

A fuzzy system can be considered to be a parameterized nonlinear map, called *f*. The expression of *f* is:

$$f(x) = \frac{\sum_{l=1}^{m} y^l \left(\prod_{i=1}^{n} \mu_{A_i^l}(x_i)\right)}{\sum_{l=1}^{m}\left(\prod_{i=1}^{n} \mu_{A_i^l}(x_i)\right)} \quad \text{.... (vi)}$$

Where, y^l is a place of output singleton if Mamdani reasoning is applied or a constant if Sugeno reasoning is applied.

The membership function $\mu_{A_i^l}(x_i)$ corresponds to the input x = $[x_1, x_2, \dots . x_n]$ of the rule l. The *and* connective in the premise is carried out by a product and defuzzification by the centre-of-gravity method.

This can be further written as $= \sum_{x=i=1}^{m} w_i b_i(x)$ .... (vii)

Where, $w_i = y^i$ and $b_j(x) = \frac{\prod_{i=1}^{n} \mu_{A_i^l}(x_i)}{\sum_{l=1}^{m}(\prod_{i=1}^{n} \mu_{A_i^l}(x_i))}$.

If F is a continuous, nonlinear map on a compact set, then f can approximate F to any desired accuracy, i.e. F ≈ f_{FS}.

5.1 ANFIS Architecture

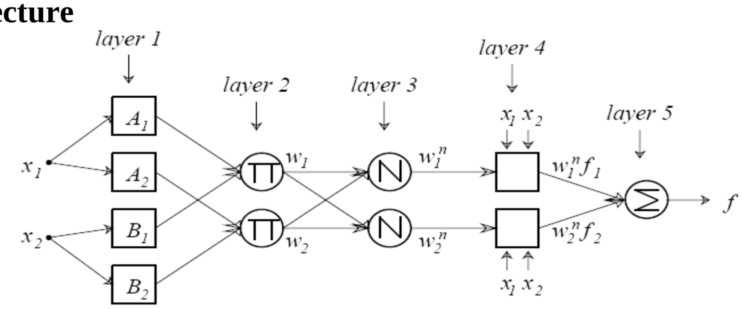

Fig 3: ANFIS Architecture

Consider a Sugeno type of fuzzy system having the rule base

If x is A_1 and y is B_1 then $f_1 = p_1 x + q_1 y + r_1$

If x is A_2 and y is B_2 then $f_2 = p_2 x + q_2 y + r_2$

Let the membership functions of fuzzy setsA_i,B_i, i=1,2 be μ_{A_i}, μ_{B_i}. In evaluating the rules, choose product for T-norm (logical *and*).

Evaluating the rule premises results in

$$w_i = \mu_{A_i}(x), \mu_{B_i}(y) \text{ , i = 1, 2.} \quad \text{..... (viii)}$$

Evaluating the implication and the rule consequences gives

$$F(x, y) = \frac{w_1(x,y) f_1(x,y) + w_2(x,y) f_2(x,y)}{w_1(x,y) + w_2(x,y)}$$

or leaving the arguments out we can write

$$f = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2}. \quad \text{..... (ix)}$$

This can be separated to phases by first defining $\overline{w_i} = \frac{w_i}{w_1 + w_2}$. Then f can be written as:

$$f = \overline{w_1} f_1 + \overline{w_2} f_2 \text{ .} \quad \text{..... (x)}$$

All computations can be presented in a diagram form in fig 3. In the ANFIS architecture shown in Fig 3, there is no weight associated with each link. The nodes in different layers perform different functions as per steps of fuzzy reasoning mechanism. Layer 1 calculates membership

Issue 2 volume 4, August 2012
ISSN 2249-9954

values, Layer 2 perform multiplication ,layer 3 computes normalized weights, layer 4 derives the product of each rule's output and corresponding normalized weight, layer 5 sums its inputs and provides overall output.

6. EXPERIMENTAL RESULTS

6.1 RBFN

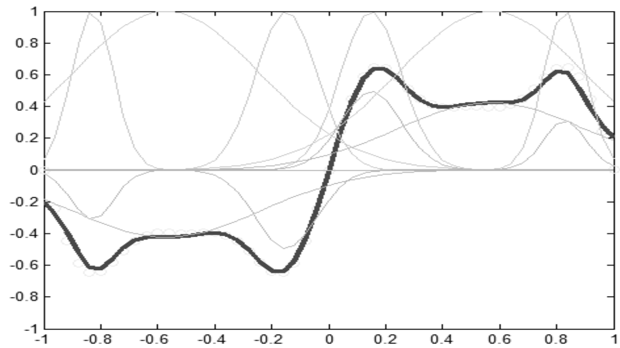

Fig 4: Basis functions, Training data, Output of RBFN

Scale used: 1) For Gaussian basis function and bell basis function
X-axis: Distance from centre, Y-axis: Amplitude
2) For training data
X- axis: Iterations, Y- axis: Function of iterations [f (iterations)]
3) Output: (weighted average of firing strength)
X-axis: Distance, Y-axis: Amplitude

The basis functions used, training of RBFN and weighted average of the firing strength is shown in the fig 4.

6.2 Fuzzy logic system

6.2.1 Membership functions

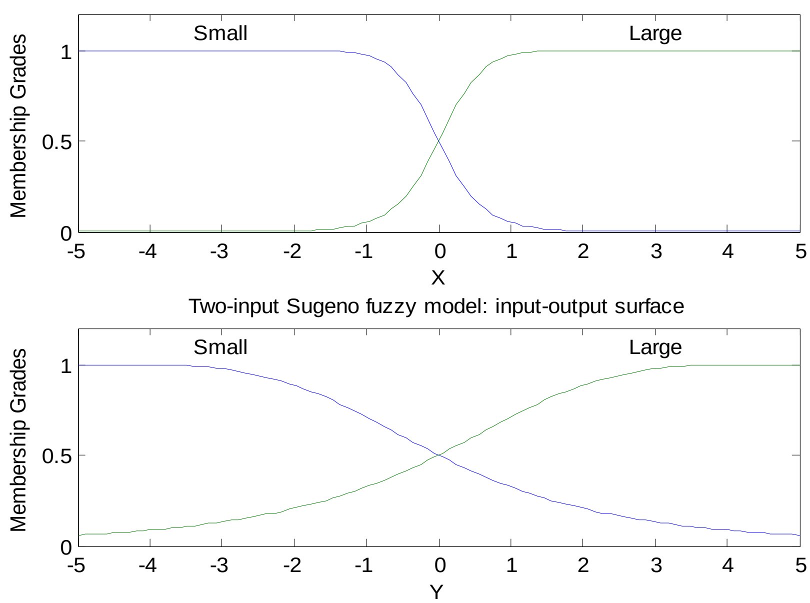

Fig 5: Membership Functions

Fig 5 shows two input variables X and Y. The degree of membership function for these two input variables is taken on the Y- axis.

6.2.2 Output

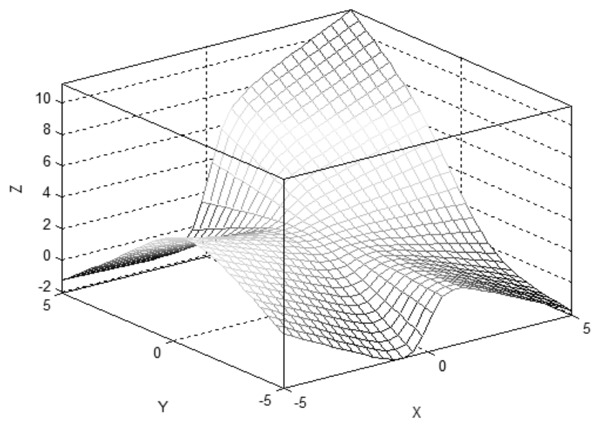

Fig 6:Fuzzy system output

Input-output surface for two-input Sugeno fuzzy model is shown in the fig 6. The equation for output is,
output = (w1.*f1+w2.*f2+w3.*f3+w4.*f4)/ (w1+w2+w3+w4).

6.3 Adaptive Neuro-Fuzzy Inference System

6.3.1 Membership functions

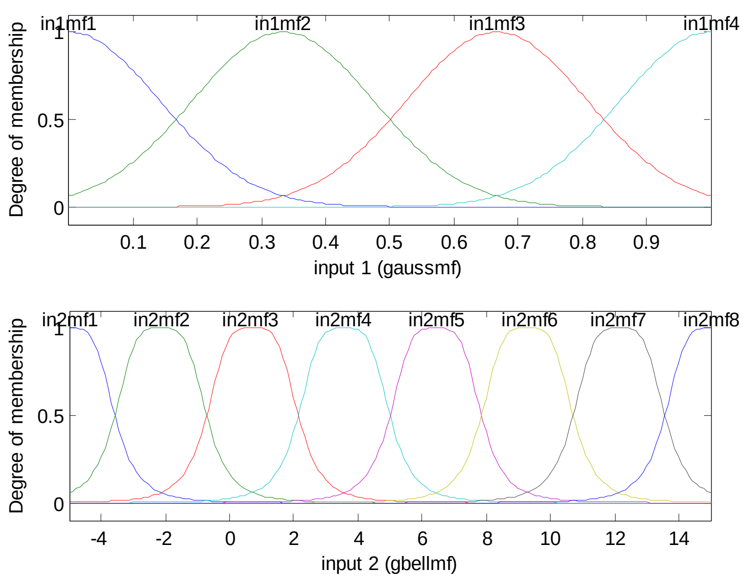

Fig 7: Membership Functions.

The number and type of membership functions are specified as, NumMf = [4 8];
MfType = str2mat ('gaussmf','gbellmf');

6.3.2 Training Data

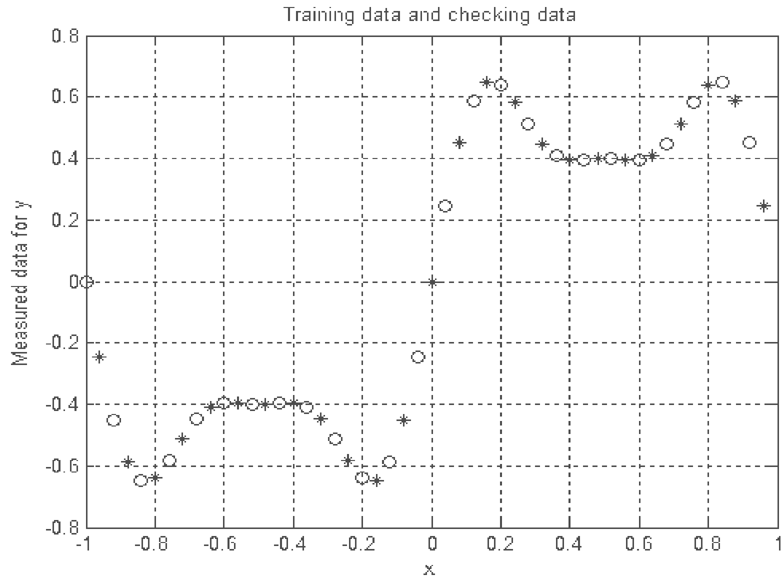

Fig 8: Training and checking data

The number of input data points = 50. ‘o’ represents training data points and ‘*’ represents checking data points

6.3.3 ANFIS output

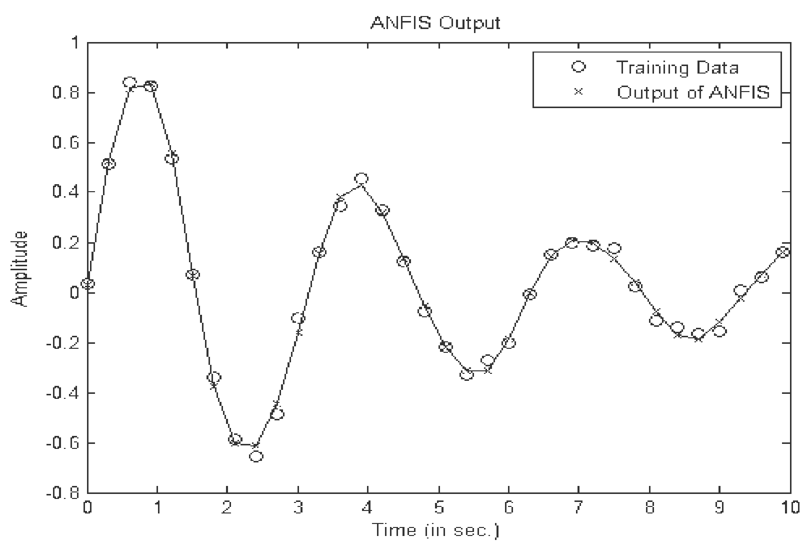

Fig 9: ANFIS Output

‘o’ represents training data points and ‘*” represents output of ANFIS.

7. PERFORMANCE EVALUATION

7.1 Root mean square error (RMSE)

7.1.1 RBFN

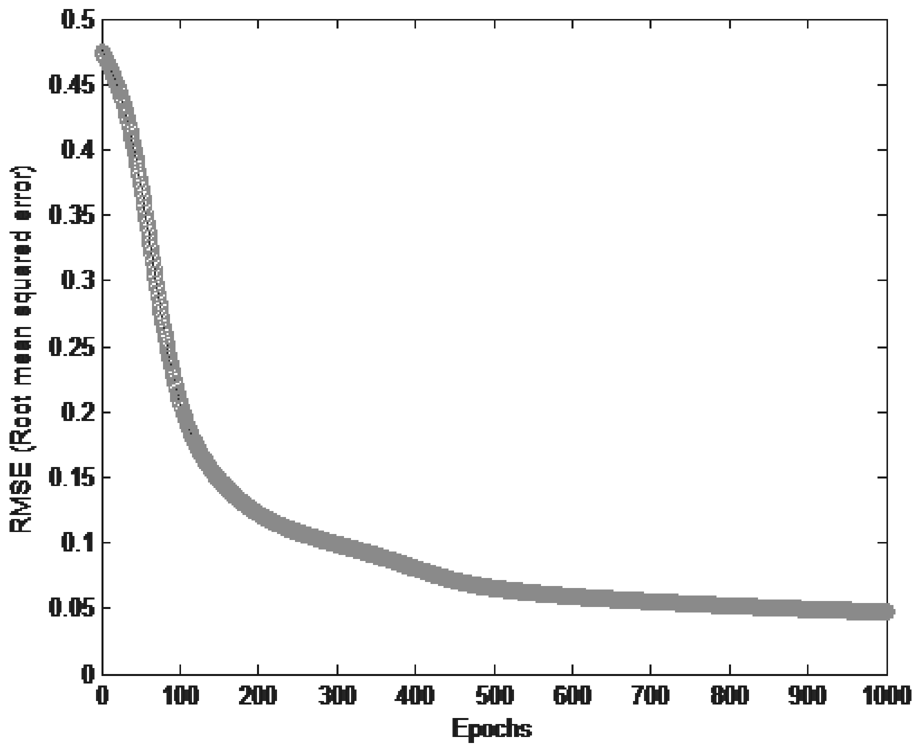

Fig 10: Plot of RMSE Vs Epochs

7.1.2 Fuzzy logic system

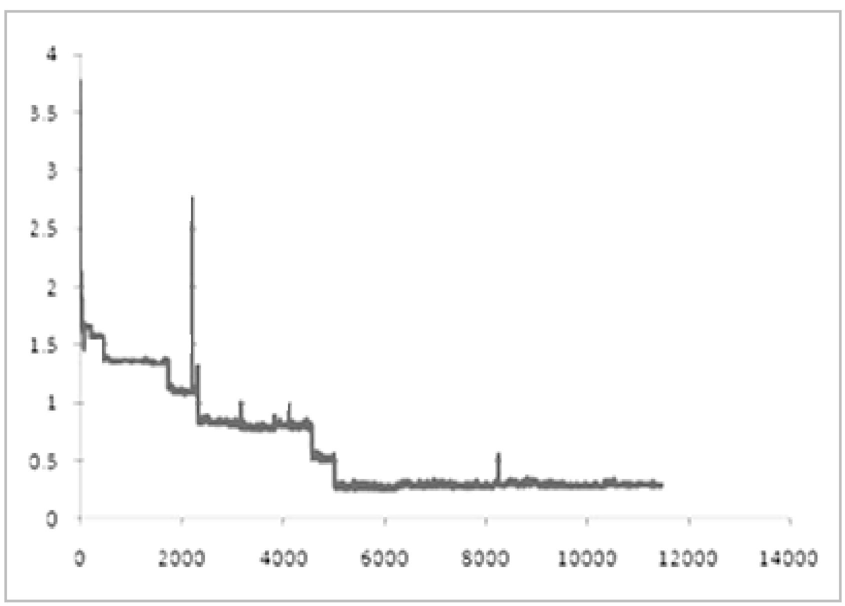

Fig 11: Plot of MSE Vs Epochs

Scale: X- axis: No. of epochs, Y- axis: mean square error.

7.1.3 ANFIS

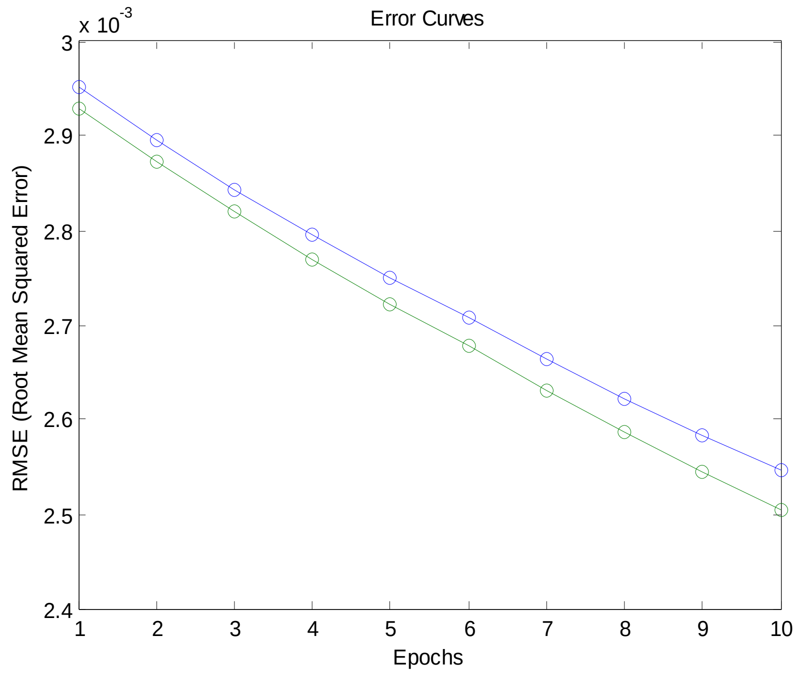

Fig 12: Plot of RMSE Vs Epochs

Scale: X- axis: No. of epochs, Y- axis: Root mean square error.

7.2 SNR VS BER Plots

7.2.1 RBFN

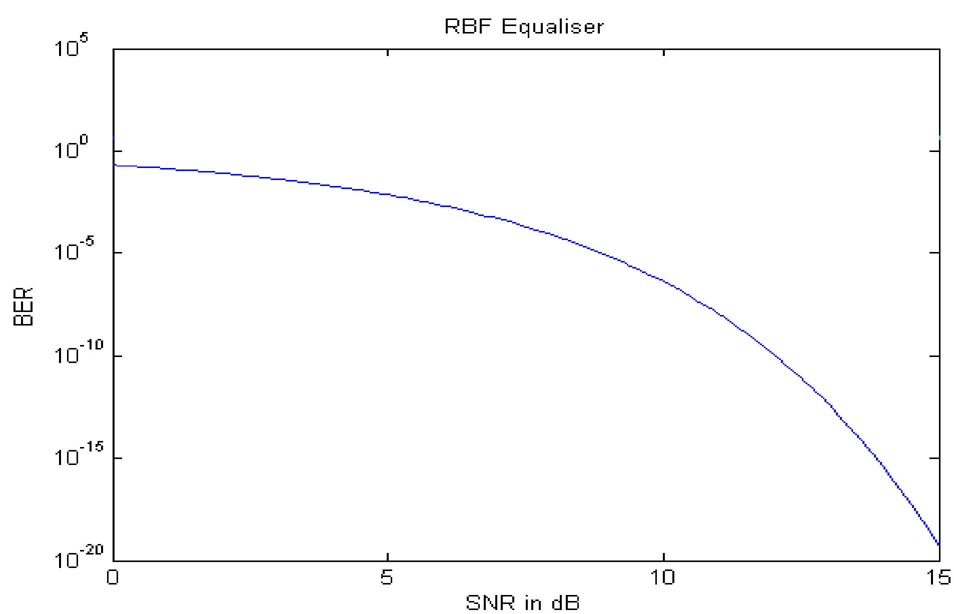

Fig 13: Plot of BER Vs SNR

Scale: X-axis: SNR value in dB, Y-axis: BER (bit-error rate).

7.2.2 Fuzzy logic system

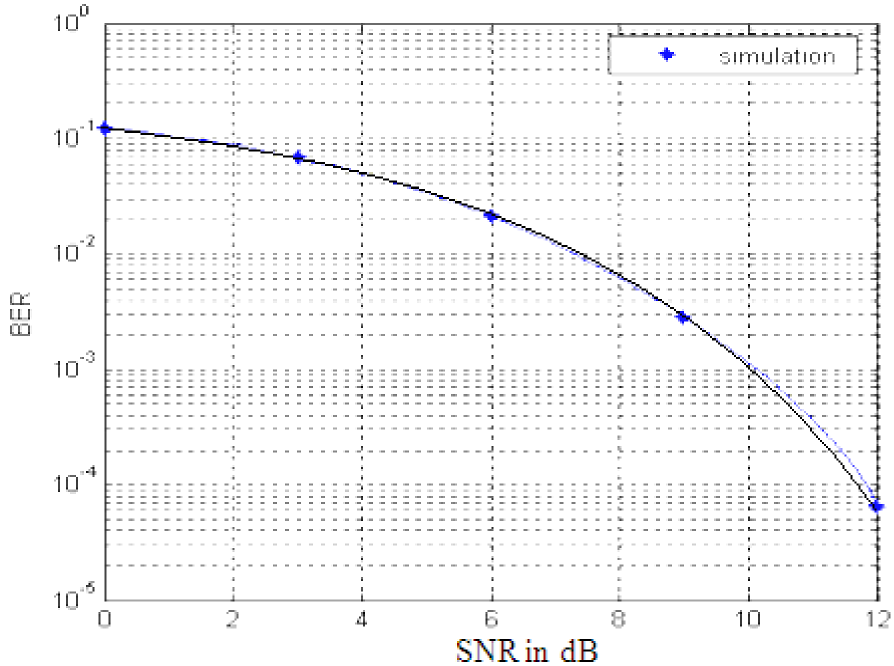

Fig 14: Plot of BER Vs SNR

Scale: X-axis: SNR value in dB, Y-axis: BER (bit-error rate).

7.2.3 ANFIS

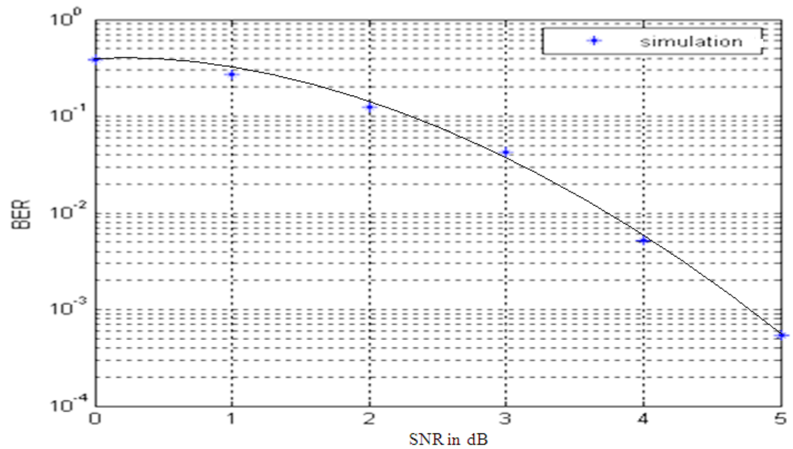

Fig 15: Plot of BER Vs SNR

Scale: X-axis: SNR value in dB, Y-axis: BER (bit-error rate).

8. PERFORMANCE COMPARISON

8.1 RMSE

Table 1. RMSE based comparison table.

Epoch Value	RBFN	Fuzzy System	ANFIS
2	0.474	1.93	0.00290
5	0.470	1.48	0.00275
10	0.464	1.26	0.00250

8.2 SNR Vs BER

Table 2. BER based comparison table.

Epoch Value	RBFN (BER)	Fuzzy System (BER)	ANFIS (BER)
0	10^{-1}	10^{-1}	$10^{-0.5}$
5	10^{-3}	$10^{-1.5}$	$10^{-3.5}$
10	10^{-6}	10^{-3}	10^{-6}

Table 1 shows that root mean square error rate for a given SNR is least for adaptive neuro-fuzzy inference system. Table 2 shows adaptive neuro-fuzzy inference system gives better bit error rate performance.

9. CONCLUSION

In this Paper we have tested performance of RBFN, fuzzy inference system and neuro-fuzzy inference system. These models are from different origins but they share some common characteristics. Radial basis function (RBF) networks and a simplified class of fuzzy systems are functionally equivalent under some mild conditions. This functional equivalence has made it possible to combine the features of these two systems, which has been developed into a powerful type of neuro-fuzzy systems. The comparison results shows that adaptive neuro-fuzzy system is more efficient than fuzzy logic systems and radial basis function networks (RBFN).

10. REFERENCES

[1] J.S. Roger Jang, **“***ANFIS: Adaptive-network-based fuzzy inference systems”* IEEE Trans. Syst. Man, Cybern. 1991.

[2] J.S. Roger Jang, “*Fuzzy modeling using generalized neural networks and Kalman filter algorithm,”* in Proc. Ninth National Conf. Artificial Intelligence (AAAI-91), July 1991, pp. 762-767.

[3] J. S. Roger Jang, “*Rule extraction using generalized neural networks,”* in Proc. 4th IFSA World Congress, July 1991.

[4] J. Moody and C. Darken, “*Learning with localized receptive fields,”* in Proc. 1988 Connectionist Models Summer School, D. Touretzky, G. Hinton, and T. Sejnowski, Eds., Carnegie Mellon University, Morgan Kaufmann Publishers, 1988.

[5] J. Moody and C. Darken, “*Fast learning in networks of locally tuned processing units,”* Neural Computation, vol. 1, pp. 281-294, 1989.

[6] L. X. Wang, *"Fuzzy systems are universal approximates,"* in Proc. IEEE Int. Conf., Fuzzy Systems, San Diego, CA, Mar. 1992.

[7] S.K.Patra, B. Mulgrew, “ *Efficient architecture for Bayesian equalization using fuzzy filters”,* IEEE Transactionon Circuit and Systems II, vol. 45, 1998, pp.812-820.

[8] S.K.Patra, B. Mulgrew, “ *Fuzzy implementation of Bayesian equalizer in the presence of intersymbol and cochannel interference*, “ Proc. Inst. Elect. Eng. Commun., vol.145, pp.1998, pp. 323-330.

[9] S.Siu, Chia-Lu, Chien-Min Lee, *“ TSK-based decision feedback equalization using an evolutionary algorithm applied to QAM Communication Systems”,* IEEE Transactions on Circuits and Systems, vol.52, No.9, 2005.

[10] Rahib Abiyev, Fakhreddin Mamedov , Tayseer Al-shanableh, *"Neuro-fuzzy system for Channel Noise Equalization"* International Conferenceon Artificial Intelligence.IC-AI'04, Las Vegas, Nevada, USA, June 21-24, 2004.
[11] J.S.Lee, C.D.Beach and N.Tepedelenlioglu, *"Channel equalization using radial basis function neural network",* Proc.IEEE Int. Conf.Acoust., Speech, Signal Proc., 1996, vol.III, (Atlanta, GA), 1996, pp.1719-1722.
[12] Deniz Erdogmus, Deniz Rende, Jose C. Principe, Tan F. Wong, *" Nonlinear channel equalization using multiplayer perceptrons with information-theoretic criterion*", Proc. Of 2001 IEEE Signal Processing Society Workshop, 2001, pp.443-451.
[13] M.S.Chavan, R.H.Chile, S.R.Sawant, "*Multipath fading channel modeling and performance comparison of wireless channel models,*" International Journal of Electronics and Communication Engineering, Volume 4, Number 2 (2011), pp. 189-203
[14] Wang J F, "*Study on Adaptive Equalization Algorithms Based on Wavelets and Neural Networks,"* Xi'an: Xidian University, 2003.
[15] Rahib H.Abiyev,"*Neuro-Fuzzy System for Equalization Channel Distortion,"* International Journal of Computational Intelligence. 2005, Fall, pp.229-232.
[16] Rahib H.Abiyev, Tayseer Al-shanableh, "Neuro-Fuzzy Network for Adaptive Channel Equalization," *Procee-dings of the Fifth Mexican International Conference on Artificial Intelligence (MICAI'06*), 2006, IEEE.
[17] LIU Zhen-Xing, GUO Ye-cai, GAO Min, ZHAO Xue-qing, "Fuzzy Neural Network Blind Equalization Algorithm Based on Radial Basis Function," *International Conference on Intelligent Human-Machine Systems and Cybernetics,* 2009, IEEE.