



Building an Adaptive Neural Fuzzy Inference System for Revised General Test/Gross Point Average Problem

Tarik A. Rashid

Software Engineering, College of Engineering
Salahaddin University
Hawler, Kurdistan

Haval A. Ahmed

Software Engineering, College of Engineering
Salahaddin University
Hawler, Kurdistan

Abstract: This paper studies both fuzzy inference systems and artificial neural networks, or so-called neuro-fuzzy techniques to solve the problem of admitting students at the registration department at a high level college. In this paper several different models were used to solve the above problems, the models were structured with different learning algorithms and activation functions. All models were subject to the same input conditions. The input to the system was three variables: GPA, GRE, Rank, and the only output was admission or non-admission. The evaluation was carried out to determine the best system; a model with a trapezoidal function and hybrid learning algorithm engendered a better result than the others.

Keywords: Fuzzy logic, Neural Networks, Neuro-Fuzzy Systems, GPA/GRE System.

I. Introduction

Fuzzy logic techniques were first suggested by Ahmed Lotfi Zadeh in 1956 [1, [2] [3]]. These techniques aim at designing a system in which users are allowed to form sets of rules via linguistic variables and membership functions, where later, the system converts these rules into their mathematical counterparts. It is worth noting that these rules must be created based on expert knowledge. The linguistic variables and rules form an input/output mapping as a description of the problem at hand.

As a matter of fact, some application problems that need fuzzy interference systems require more than just expert knowledge of human behaviours, because they often involve more exactness in identifying and selecting the location of fuzzy sets and the number of rules than have been applied up to this time.

Thus, the processes of selecting the location of fuzzy sets and the number of rules are exasperating and they take long time to be tuned via trial and error. Bearing mind that fuzzy inference systems will only play little to no role in solving many application problems, these application problems can acquire a solution by using a fuzzy-logic rule base having spontaneous learning techniques that take patterns from data sets [4].

The idea of applying learning techniques to fuzzy logic is conceivable as long as we manage to parameterise all the components of the fuzzy inference system in order to produce a system in which its parameters can be tuned through the training process of the learning algorithm [4].

Both fuzzy logic and artificial neural networks [5] [6] are corresponding tools for designing systems that deal with prediction and classification tasks. The amalgamation of these two tools produces a cohesive and unified system that appears to be a favourable track for constructing systems with the ability to incorporate qualities exemplifying the intelligence capacity of the human brain. It is worth mentioning that both neural networks and fuzzy logic are prevailing design techniques that both have their ups and downs characteristics [4].

The new combined system will inherit some good traits from neural networks such learning competence, generalisation abilities, optimization capabilities modern mathematical models and adaptation of connectionist structures; while fuzzy systems will extend and add its features such as human like IF-THEN rules thinking and ease of incorporating expert knowledge [7] and the simplicity of linguistic variables with no mathematical expressions. By so doing, the new system will convey the low-level learning and computational power of neural

networks into fuzzy systems, and will also transfer the high-level humanlike IF-THEN thinking and reasoning of fuzzy systems into neural networks.

The amalgamation of these two techniques often will not mean that a neural network and a fuzzy system are composed in a certain way. The neural networks on their own with their ability to behave as black box are not able to create linguistic variables to behave in black box. Therefore, the neuro-fuzzy comes in hand in hand to create a fuzzy system using the data samples via some of sort of learning algorithms that are inspired by the learning procedures used in artificial neural networks. By doing so, this new system will significantly diminish both time and costs for producing a result that is more accurate.

As a matter of fact, combining neural network with a learning algorithm in which a fuzzy principle with language variables disguised in its rules that are formed and adjusted using numerical information is even better than a neural network only.

Different terminologies for the idea of neuro-fuzzy systems instituted in the literature were neural-fuzzy or sometimes neuro-fuzzy systems [8]. The term neuro-fuzzy system is normally an abbreviation of adaptive fuzzy systems industrialised by manipulating the resemblances amid fuzzy systems and some methods of neural networks, which can be categorised under the class of generalized local methods. Having a set of human like understandable rules or an amalgamation of localised basis functions associated with generalised or local models can represent the behaviour of a neuro-fuzzy system. This property has made a neuro-fuzzy system xxx a perfect framework to handle the nonlinear predictive.

We can say to a point that a neural network progresses its lucidities, pushing it nearer the fuzzy systems, and by the same token, the fuzzy-inference system acts to adapt itself pushing itself nearer neural networks [9].

We can look at a fuzzy-inference system as a distinctive instance of a local-model technique, in which the input variable spaces or each input variable space can be divided into different hedges of fuzzy regions characterised by multivariate membership functions. In every single region a particular rule can be well-defined in which the output can be specified in the same region. The multivariate fuzzy membership function performs nonlinear mapping to determine accurately the class function y rustling model.

The above trait helps us make association between fuzzy systems and the more conventional techniques referred to as generalised local methods; e.g., when bell-shaped Gaussian membership functions are used, then a Takagi-Sugeno fuzzy system is comparable to a special kind of Radial Basis Function network [10,11,12, 13].

Some learning algorithms based on local modelling of neural networks have been extended to tune fuzzy rules using data sets. A layered feed forward architecture is proposed in 1991, by Lin and Lee [14, 15]. It was the first Mamadan fuzzy models implementation [16]. However, in 1993, Jang developed the common example of neuron fuzzy network called the Adaptive Network-based Fuzzy Inference System (ANFIS) [10, 11, 12, 13]. The ANFIS applies a Takagi-Sugeno fuzzy system in network architecture, bearing in mind that the learning algorithm was a combination of both back-propagation and least mean squares procedure to train the system [17, 18].

II. Fuzzy Operators

Generally speaking, there are main connective operations in classical set theory: namely intersection, union and complement. We present only the standard operations in this paper.

We can describe the operations as follows:

- 1) The fuzzy intersection operator $\cap$ / And applied to two fuzzy sets A and B with the membership functions $\mu_A(x)$ and $\mu_B(x)$ is $\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}, x \in X$
- 2) The fuzzy union operator $\cup$ / union applied to two fuzzy sets A and B with the membership functions $\mu_A(x)$ and $\mu_B(x)$ is $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}, x \in X$

The fuzzy complement (fuzzy NOT operation) applied to the fuzzy set A with the membership function $\mu_A(x)$ is $\mu_{\bar{A}}(x) = 1 - \mu_A(x), x \in X$

Whilst the operations according to above equations are based on min/max operations, the complement is an algebraic one. Union and intersection can also be defined in an algebraic manner but giving different results as:

- 1) The *fuzzy intersection* operator $\cap$ (fuzzy AND connective) can be represented as the *algebraic product* of two fuzzy sets A and B , which is defined as the multiplication of their membership functions:

$$\mu A \cap B(x) = \mu A(x) \mu B(x), x \in X$$

- 2) The *fuzzy union* operator $\cup$ (fuzzy OR connective) can be represented as the *algebraic sum* of two fuzzy sets A and B , which is defined as:

$$\mu A \cup B(x) = \mu A(x) + \mu B(x) - \mu A(x) \mu B(x), x \in X$$

Traditional Bivalent logic uses the Boolean operators AND, OR, and NOT to perform the intersection, union and complement operations. These operators work well for bivalent sets and can be essentially defined using the following truth table (See Table I):

Table I shows the Boolean operators and, or, and not.

X	Y	$X \cap Y$	$X \cup Y$	NOT X
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

The truth table above works fine for bivalent logic but fuzzy logic does not have a finite set of possibilities for each input; this makes for an infinitely large truth table. The operators need to be defined as functions for all possible fuzzy values, that is, all real numbers from 0 to 1 inclusive. Fuzzy logic is actually a superset of bivalent logic since it includes the bivalent options (0, 1) as well as all reals in between, so a generalized form of these operators will be useful. The generalized forms for these three operators are (See Table II):

Table II shows a generalized form of the above operations.

X	Y	$X \cap Y$	$X \cup Y$	NOT X
x	y	$\min(x,y)$	$\max(x,y)$	$1 - x$

Using these definitions they can be applied to all of the bivalent combinations above as well as some fuzzy number combinations. These generalised definitions of the operators work well for fuzzy numbers as well as bivalent sets. The behaviour of the new generalized definitions of AND, OR and NOT can be visualised below for all possible inputs of x and y (see Table III):

Table III shows a generalized form of the above operations with examples.

X	Y	MIN(X,Y)	MAX(X,Y)	1 - X
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0
0.4	0.7	0.4	0.7	0.6
0.8	0.4	0.4	0.7	0.2
0.5	0.5	0.5	0.5	0.5

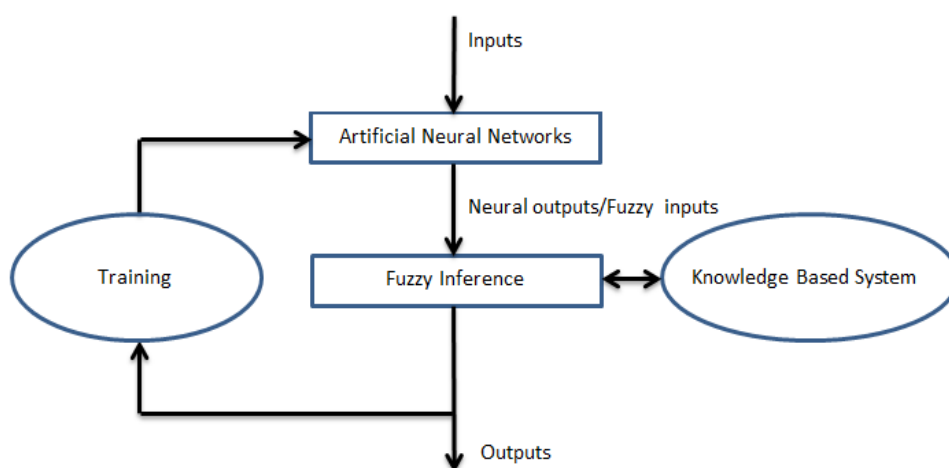
III. Types of Neuro-Fuzzy Systems

Researchers have tried to combine the two techniques of fuzzy logic and neural networks in several different ways. Generally speaking, there are three combinations techniques and these are neural-fuzzy systems, fuzzy-neural networks and fuzzy-neural hybrid systems.

A. Neural Fuzzy Systems

In this scheme, the neural network is used to determine function and mapping between fuzzy sets that are utilized as fuzzy rules as shown in Fig. 1. The neural network changes its weight in the training in the hope of minimising the mean square error between the actual output of the networks and the targets. The parameters such as the fuzzification function, fuzzy word membership function, and fuzzy rule confidences are being represented in the weights of the neural network.

Figure 1: shows a neural fuzzy system.

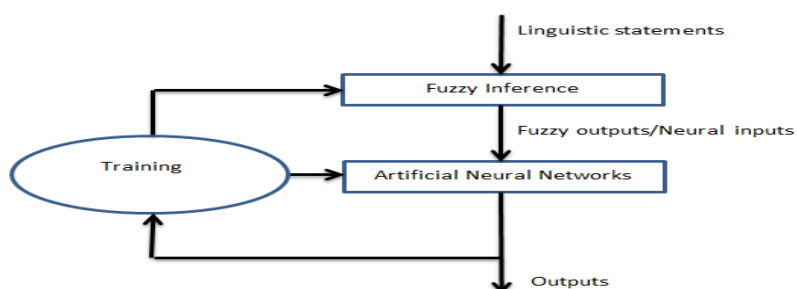


Automatic adjusted parameters and their optimal values of a fuzzy system are produced from the neural network's training. Controller systems are a good example of this kind of amalgamation. Examples of this approach can be found in [19, 20, 21].

B. Fuzzy Neural System

As you can see from its name, the technique tries to fuzzificate some of the features of neural networks via fuzzy logic (Fig. 2). In other words, a crisp neuron can become fuzzy. These types of fuzzy neural systems are used in pattern recognition applications. In 1996, for instance, Lin and Lee presented a neural network composed of fuzzy neurons [9]. The inputs are non-fuzzy but the weighting operations are replaced by membership functions. The outcome of each weighting operation is the membership value of the corresponding input in the fuzzy set. Also, the aggregation operation may use any aggregation operators such as min and max and any other t-norms and t-conorms.

Figure 2: shows a fuzzy neural system.



C. Hybrid Neuro-Fuzzy Systems

A hybrid system is made independently from both fuzzy-logic and neural-network techniques to carry out applications such as pattern recognition and control systems. The main objective of the hybrid system can be accomplished by having each technique do its task by integrating and complimenting one another. This kind of merging is application-oriented and suitable for both control and pattern recognition applications.

The idea is to interpret the fuzzy rules in the context of neural network. In other words, the fuzzy sets can be interpreted as weights. The rules input and output variables can be represented as neurons. In a neural network, the result of the learning algorithm reflects an alteration in the internal structure of the interconnection (adaptation of weights, or new or deleting connections). The black box aspect of neural networks can be averted by the above alteration, in other words the alteration can be represented or interpreted in terms of both neural networks and fuzz controller. It is worth mentioning that the above alteration will make the knowledge increase clearly as a result of the learning process. The approaches such as ARIC, GARIC, ANFIS the NNDFR model [22, 23, 10, 11, 17] are good examples of hybrid neuro-fuzzy controllers. These techniques comprise more or less elements of neural networks and they have the ability to learn fuzzy sets.

IV. Adaptive Neuro-Fuzzy Inference System

The adaptive neuro-fuzzy inference system is a common approach in which the two techniques such as a neural network and a fuzzy logic get combined to create a complete shell [10] Basically the system of ANFIS applies the technique of the artificial neural network learning rules to determine and tune the fuzzy inference systems' structure and parameters. A number of important features of ANFIS can help the system accomplish a task brilliantly; these features are characterised as easy to implement, fast and accurate learning, strong generalization abilities, excellent explanation facilities through fuzzy rules, and easy to incorporate both linguistic and numeric knowledge for problem solving [10,11,12,13, 14,15].

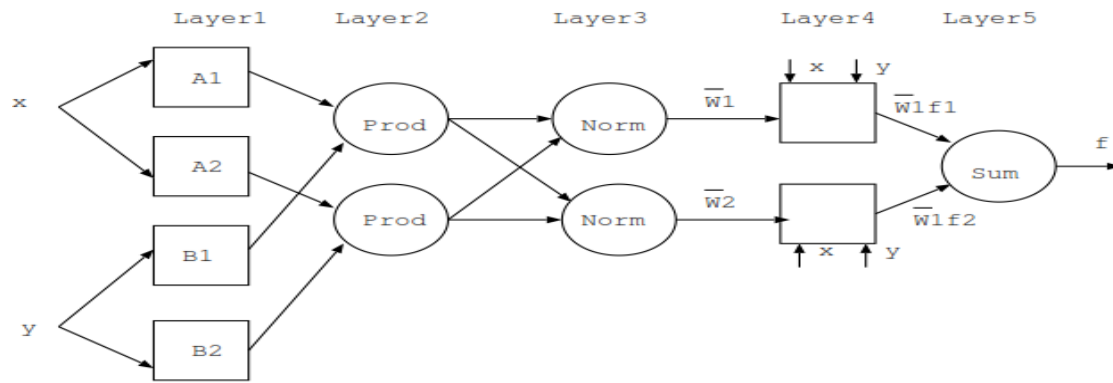
In Neuro-Fuzzy technique a neural network is introduced to devise the fuzzy system so that the structure and parameters which identify the fuzzy rules are accomplished by adopting and optimising the topology and the parameters of corresponding the neuron fuzzy network based on data sets. The system is considered to be an adaptive fuzzy inference system with the capability of learning fuzzy rules from data and as a connectionist architecture provided with linguistic meaning. Jang had developed one type of hybrid neuro-fuzzy inference expert system that works in Takagi-Sugeno type fuzzy inference system [24, 25, 26, 27, 28]. This is called ANFIS. Both the Feed forward neural network and ANFIS have a similar structure but the links in an ANFIS only indicate the flow direction of signals between nodes and no weights are associated with the links. The architecture of the ANFIS consists of five layers. Among those layers both the first and the fourth layers consist of adaptive neurons despite the fact that the second, third and fifth layers consist of fixed neurons. The adaptive neurons are related with their respective parameters, and get duly updated with each in subsequent iteration, while the fixed nodes are devoid of any parameters [24, 25, 26, 27, 28].

Rule 1: If (x is A1) and (y is B1) then ($f_1 = p_1x + q_1y + r_1$)

Rule 2: If (x is A2) and (y is B2) then ($f_2 = p_2x + q_2y + r_2$)

Where x and y are the inputs, A_i and B_i are the fuzzy sets, f_i are the outputs within the fuzzy region specified by the fuzzy rule, p_i, q_i, r_i is the parameter set of this node. These are referred to as consequent parameters. The design parameters those are determined during the training process. The architecture of ANFIS is shown in Fig. 1, in which a circle indicates a fixed neuron, whereas a square indicates an adaptive [24, 25, 26, 27, 28].

Figure 3: shows ANFIS structure.



The layers of ANFIS are explained as follows:-

Layer 1: this is called the fuzzification layer, in this layer every ith neuron is an adaptive neuron. The outputs of this layer are the fuzzy membership grade of the inputs, they can be expressed as

$$O_{1,i} = \mu A_i(x) \dots \dots \dots (1) \quad \text{for } i=1, 2$$

$$O_{1,i} = \mu B_{i-2}(y) \dots \dots \dots (2) \quad \text{for } i=3, 4$$

As mentioned above x or y is the inputs to neuron i and A_i or B_{i-2} is linguistic label associated with this neuron.

$O_{1,i}$ is the membership grade of a fuzzy set (A_1, A_2, B_1, B_2)

Usually we choose $\mu A_i(x)$ to be bell-shaped with maximum equal to 1 and minimum equal to 0, such as:

$$\mu A_i(x) = \frac{1}{1 + \left\| \frac{x - c_i}{a_i} \right\|^{2b_i}} \dots \dots \dots (3)$$

A_i , b_i and c_i are the parameters set. Parameters are referred to as premise parameters.

Layer 2: this is called a rule layer, a fixed neuron labelled Prod whose output is the product of all the incoming signals, the outputs of this layer can be represented as:

$$O_{1,i} = w_i = \mu A_i(x) \cdot \mu B_i(y) \dots \dots \dots (4)$$

As can be seen from the Fig. 3, that every neuron in this layer is fixed and labelled prod .The output is the product of all the incoming signals. Each neuron represents the fire strength of the rule. Any other T-norm operator that performs the AND operator can be used.

Layer 3: this is a normalization layer, the neurons in this layer are also fixed and indicated as a circle, and labelled norm. The i th neuron computes the ratio of the i th rule's firing strength to the sum of all rule's firing strengths. Outputs are called normalized firing strengths. $\bar{w}_i$ is the normalized firing strength from layer 3.

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2} \dots\dots\dots(5) \quad \text{For } i = 1, 2$$

Layer 4: is called the defuzzification layer, the neurons are adaptive. The output of each neuron in this layer is simply the product of the normalized firing strength and a first order polynomial. The neuron function can be expressed as:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_x + q_i y + r_i) \dots\dots\dots(6)$$

Layer 5: this is called a single, summation neuron and it is fixed and labelled sum which calculates the overall output as the summation of all incoming signals.

$$O_{5,i} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \dots\dots\dots(7)$$

A hybrid learning algorithm is used to train the ANFIS model in the hope to adjust all the different parameters above mentioned. The data sets with inputs and outputs are feed to the ANFIS. The back propagation and the least square algorithms are coupled to change and update the membership function parameters of the fuzzy inference system.

V. Simulation and Results

In this paper several ANFIS systems have been used via Math lab. Each system is being built via steps as follows:

A. Inputs/outputs

A relatively large data set is used in this paper. 400 samples are used. The data set is divided into two parts: the training set in which the number of samples was 300, and the testing set in which the number of samples was 100. The systems are fed with identical inputs and output variables. Three input features are fed in to the systems, namely: Revised General Test (GRE), Gross Point Average (GPA) and Rank. The system has one output only admit or non-admit.

B. Structure of the systems

ANFIS systems are used to implement the application of Revised General Test/Gross Point Average problem ith Sugeno technique. Different membership functions such as triangle and trapezoidal functions are used for the three inputs, linear and constant functions are used for the output. Different learning algorithms are used, namely back propagation and hybrid (combination of back propagation and least square error) learning algorithms [29, 30, 31, 32, 33, 34, 35, 36].

After performing different tests by changing functions and parameters, we acquired the following results as shown in Figure 4, a-d and Figure 5, e-h. The actual outputs as they are indicated in red colour dotted line and the targets as in blue colour dotted line.

Figure 4: shows ANFIS results (a, b, c, d).

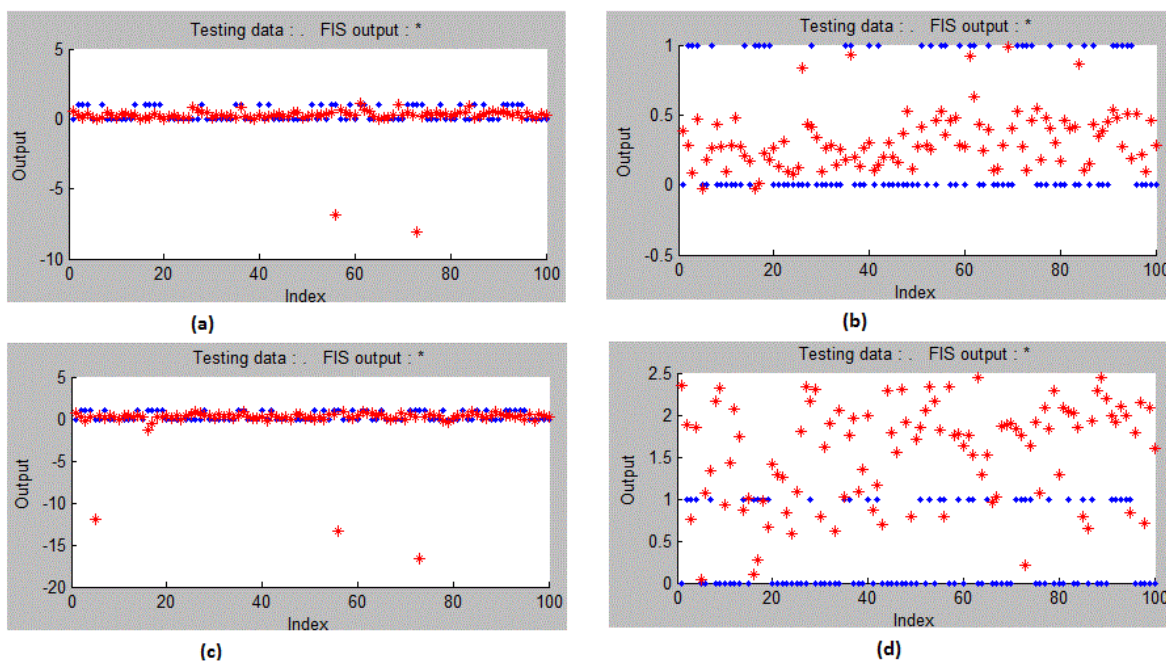


Figure 5: shows ANFIS results (e, f, g, h).

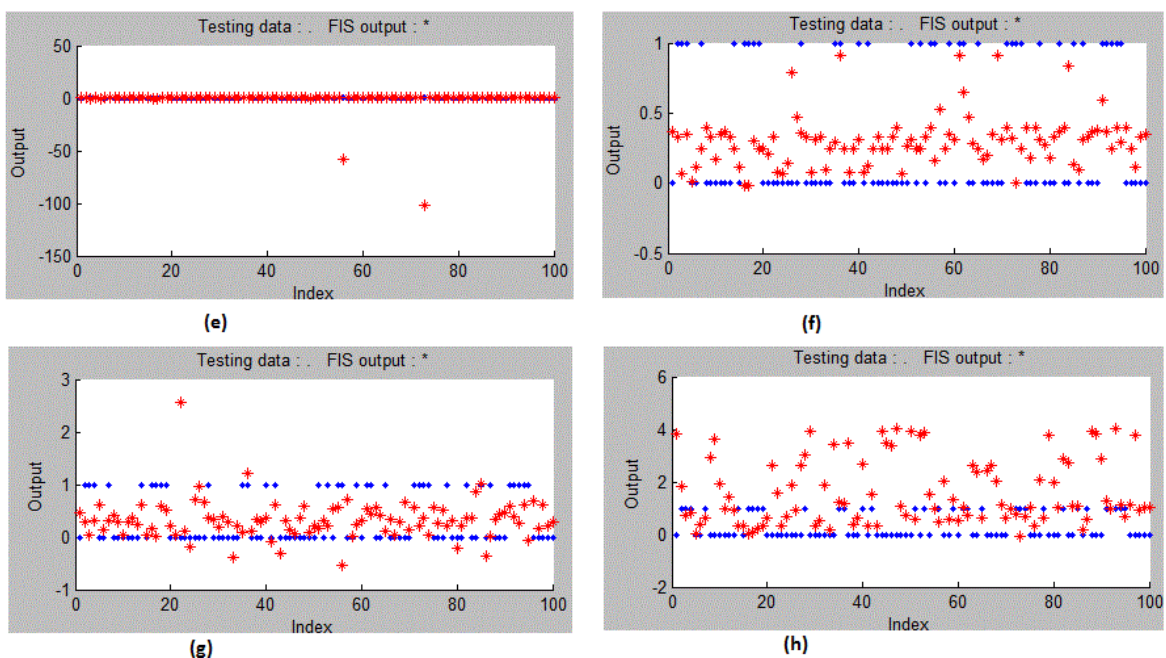


Table IV and Table V, show different tests for ANFIS with different parameters, Table IV, shows a test for each model using a graphical user interface in the math lab. A test result for model 3, produces the least error on the training data set, however the error on the testing data set is the highest among all.

Table IV [Different tests of ANFIS - GUI]

	Inputs					Outputs		
	No. of MFs	Input MF Type	Output MF Type	Train FIS	Epochs	Training Error	Avg. Testing Error	Plot
Test1	3 3 3	Triangle mf	Constant	hybrid	100	0.402211	1.3034	(a)
Test2	3 3 3	Triangle mf	Constant	backprop	100	0.41679	0.49881	(b)
Test3	3 3 3	Triangle mf	Linear	hybrid	100	0.36412	2.6444	(c)
Test4	3 3 3	Triangle mf	Linear	backprop	100	1.3635	1.4575	(d)
Test5	3 3 3	Trapezoidal mf	Constant	hybrid	100	0.40371	11.8847	(e)
Test6	3 3 3	Trapezoidal mf	Constant	backprop	100	0.41267	0.50257	(f)
Test7	3 3 3	Trapezoidal mf	Linear	hybrid	100	0.38006	0.59609	(g)
Test8	3 3 3	Trapezoidal mf	Linear	backprop	100	1.8154	1.8943	(h)

Table V, shows a test for each model using a command line. As can be seen from Table V, a test result for a model 5 produced a better classification result among all with only 2 training epochs. A hybrid learning algorithm was used, a trapezoidal membership function was used, and the output function was constant.

Table V [Different tests of ANFIS – Command Line]

	Inputs					Outputs
	No. of MFs	Input MF Type	Output MF Type	Train FIS	Epochs	Correctly Classified Total: 100 Samples
Test1	3 3 3	Triangle mf	Constant	hybrid	2	65
Test2	3 3 3	Triangle mf	Constant	backprop	2	65
Test3	3 3 3	Triangle mf	Linear	hybrid	2	58
Test4	3 3 3	Triangle mf	Linear	backprop	70	64
Test5	3 3 3	Trapezoidal mf	Constant	hybrid	2	66
Test6	3 3 3	Trapezoidal mf	Constant	backprop	2	65
Test7	3 3 3	Trapezoidal mf	Linear	hybrid	10	62
Test8	3 3 3	Trapezoidal mf	Linear	backprop	150	66

VI. Conclusions

Artificial neural networks or so-called neuro-fuzzy techniques can be used to solve the problem of admission of students at the registration department at high level college.

The advantages of the fuzzy systems are: capacity to represent inherent uncertainties of human knowledge with linguistic variables; simple interaction of the expert of the domain with the engineer designer of the system; easy interpretation of the results because of the natural rules representation; easy extension of the base of knowledge through the addition of new rules; and robustness in relation of the possible disturbances in the system. Its disadvantages are: it is incapable of generalizing either/or, it only answers to what is written in its rule base; it is not robust in relation the topological changes of the system, such changes would demand alterations in the rule base; and it depends on the existence of an expert to determine the inference logical rules.

Neural networks have important features, these are learning capacity, generalization capacity, and robustness in relation to disturbances. And their drawbacks are the impossible interpretation of the functionality, and the difficulty in determining the number of layers and number of neurons.

Neural networks and fuzzy systems can be combined to join their advantages and to cure their individual drawbacks. Neural networks introduce their computational characteristics of learning into fuzzy systems and receive from them the interpretation and clarity of systems representation. Thus, the disadvantages of the fuzzy

systems are compensated for by the capacities of the neural networks. These techniques are complementary, which justifies their use together.

In this paper several different models were used to solve the above problems with different learning algorithms and activation functions. All models were subject to the same input conditions. A model that uses a hybrid learning algorithm, a trapezoidal membership function, and a constant output function produced more accurate results with a few training epochs.

VI. Reference

- [1] Zadeh, L.. Fuzzy Sets. *Inf Cont*, Vol. 8, Pp. 338–353, 1965.
- [2] Royas, I.; Pomares, H.; Ortega, J.; And Prieto, A. (2000). Self-Organized Fuzzy System Generation From Training Examples, *Ieee Trans. On Fuzzy Systems*, Vol. 8, No. 1, Pp. 23-36, 2000.
- [3] Cox, E. , *The Fuzzy Systems Handbook*. Ap Professional - New York.1994.
- [4] Azar, Ahmad Taher Adaptive Neuro-Fuzzy Systems, Electrical Communication & Electronics Systems Engineering Department, Modern Science And Arts University (Msa), 6 Th Of October City, Egypt.
- [5] Haykin, S. , *Neural Networks, A Comprehensive Foudation*. Second Edition, Prentice Hall. 1998.
- [6] Mehrotra, K., Mohan, C. K., And Ranka, S. ,*Elements Of Artificial Neural Networks*. The Mit Press, 1997
- [7] Brown M. & Harris C.J. (1995). *Neurofuzzy Adaptive Modelling And Control*. Prentice Hall, 1 Edition, Hemel Hempstead, 1995.
- [8] Buckley, J.J. & Eslami, E., *Fuzzy Neural Networks: Capabilities*. In *Fuzzy Modeliparadigms And Practice* , Pedrycz W, Ed., Pp. 167-183, Kluwer, Boston, 1996.
- [9] Lin, C.T. And Lee, G. , *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism To Intelligent Systems*. Ed. Prentice Hall. 1996.
- [10] Jang, J.S.R. & Sun, C.T., *Functional Equivalence Between Radial Basis Function Networks And Fuzzy Inference Systems*, *Ieee Trans. On Neural Networks*, Vol. 4, No. 1, Pp. 156-159, 1993.
- [11] Jang, J.S.R. ,*Anfis: Adaptive-Network-Based Fuzzy Inference System*, *Ieee Transactions On Systems, Man And Cybernetics*, Vol. 23, No.3, Pp. 665–685. 1993.
- [12] Jang, J.S.R. & Sun, C.T. , *Neuro-Fuzzy Modeling And Control*, *Proceedings Of The Ieee*, Vol. 83, Pp. 378-406,1995.
- [13] Jang, J.S.R; Sun, C.T & Mizutani, E. , *Neuro-Fuzzy And Soft Computin*. Prentice-Hall: Englewood Cliffs, Nj, 1997.
- [14] Lin, C.T. & Lee, C.S., *Neural-Network-Based Fuzzy Logic Control And Decision Systems*. *Ieee Trans. On Computers*, Vol. 40, No. 12, Pp. 1320-1336, 1991
- [15] Lin, C.T. And Lee, G. , *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism To Intelligentsystems*. Ed. Prentice Hall, 1996.
- [16] Mamdani, E.H. & Assilian, S.,. An Experiment In Linguistic Synthesis With A Fuzzy Logic Controller, *International Journal Of Man-Machine Studies*, Vol. 7, No. 1, Pp. 1-13, 1975
- [17] Takagi, H. & Hayashi, I., *Nn-Driven Fuzzy Reasoning*. *International Journal Of Approximate Reasoning*, Vol. 5, Issue 3, 1991.
- [18] Takagi, T. And Sugeno, M., *Fuzzy Identification Of Systems And Its Applications To Modeling And Control*. *Ieee Trans Syst Man Cybern*, Vol. 15, No. 1, Pp. 116-132, 1985.
- [19] Wang, L. And Mendel, J., *Back-Propagation Fuzzy System As Nonlinear Dynamic System Identifiers*. *Proceedings Of Ieee International Conferenceon Fuzzy Systems*, Pages 1409–1416, 1992
- [20] Shi, Y. And Mizumoto, M. (2000a). A New Approach Of Neurofuzzy Learning Algorithm For Tuning Fuzzy Rules. *Fuzzy Sets And Systems*, 112(1):99–116, 2000a.
- [21] Shi, Y. And Mizumoto, M., *Some Considerations On Conventional Neuro-Fuzzy Learning Algorithms By Gradient Descent Method*. *Fuzzy Sets And Systems*, Vol. 112, No. 1, Pp. 51–63, 2000b.
- [22] Berenji, R.H. , *A Reinforcement Learning-Based Architecture For Fuzzy Logic Control*. *International Journal Of Approximate Reasoning*, Vol. 6, Issue 2, 1992.
- [23] Bersini H.; Nordvik, J.P & Bonarini, A. , *A Simple Direct Adaptive Fuzzy Controller Derived From Its Neutral Equivalent*, *Proceedings Of 2 Ieee International Conference On Fuzzy Systems*, Vol. 1, Pp. 345-350. Nd, 1993.
- [24] Abraham A., "Adaptation Of Fuzzy Inference System Using Neural Learning, *Fuzzy System Engineering: Theory And Practice*", Nadia Nedjah Et Al. (Eds.), *Studies In Fuzziness And Soft Computing*, Springer Verlag Germany, Isbn 3-540-25322-X, Chapter 3, Pp. 53–83, 2005.
- [25] Tharwat E. Alhanafy, Fareed Zaghloul And Abdou Saad El Din Moustafa, *Neuro Fuzzy Modeling Scheme For The Prediction Of Air Pollution*, *Journal Of American Science*, 6(12) 2010.
- [26] T. M. Nazmy, H. El-Messiry, B. Al-Bokhity, *Adaptive Neuro-Fuzzy Inference System For Classification Of Ecg Signals*, *Journal Of Theoretical And Applied Information Technology*, Pp-71-76, 2010.
- [27] Abdulkadir Sengur., "An Expert System Based On Linear Discriminant Analysis And Adaptive Neurofuzzy Inference System To Diagnosis Heart Valve Diseases, *Expert Systems With Applications*, 2008.
- [28] G. Zhao, C. Peng And Xiting Wang., "Intelligent Control For Amt Based On Driver's Intention And Anfis Decision-Making," *World Congress On Intelligent Control And Automation*, 2008.
- [29] Anupam Das, J. Maiti And R.N. Banerjee., "Process Control Strategies For A Steel Making Furnace Using Ann With Bayesian Regularization And Anfis," *Expert Systems With Applications*, 2009.
- [30] Tarik A. Rashid And Haval A. Ahmed" Revised General Test/Gross Point Average System Via Fuzzy Logic Techniques" *Journal Of Computer Science And Engineering*, © 2012 Jcse www.journalcse.co.uk , Volume 16, Issue 1, November 2012.
- [31] Hojjat Adeli, F.Asce And Xiaomo Jiang , *Neuro-Fuzzy Logic Model For Freeway Work Zone Capacity Estimation 2*, *Journal Of Transportation Engineering* © Asce / September/October 2003.
- [32] Azar, A.T.; Kandil, A.H.; Wahba, K. And Massoud, W., *Prediction Of Post-Dialysis Blood Urea Concentration Using Artificial Neural Network*. *Proceedings Of 2 International Conference On Advanced Control Circuits & Systems (Accs'08)*, March 30/april 2, 2008, Cairo, Egypt. www.intechopen.com Nd Adaptive Neuro-Fuzzy Systems, 20008.
- [33] Azar, A.T.; Kandil, A.H.; Wahba, K. And Massoud, W. , *Neuro-Fuzzy System For Post-Dialysis Urea Rebound Prediction*, *Proceedings Of Ieee 4 Cairo International Biomedical Engineering Conference (Cibec'08)*, Dec. 18-20, , Cairo, Egypt, 2008
- [34] Azar, A.T.; Kandil, A.H.; Wahba, K., *Artificial Intelligence For Prediction Of Equilibrated Dialysis Dose Without Intradialytic Sample*. Saudi J Kidney Dis Transplant, To Be Published, 2000a
- [35] Azar, A.T., *Adaptive Neuro-Fuzzy System For Hemodialysis Treatment Process*, Ph.D. Dissertation, Dept. Sys. & Biomed. Eng., Cairo Univ., Egypt.2009b
- [36]. Christian Schmid, , *Introduction Into System Control*<http://www.atp.ruhr-uni-bochum.de/Rt1/Syscontrol/Node1.htm>, 2005.