IJCSMS International Journal of Computer Science & Management Studies, Vol. 12, Issue 03, September 2012
ISSN (Online): 2231 –5268
www.ijcsms.com

FUZZY NEURAL NETWORKS

Vikas Kumar[1], Dr. J.S Sohal[2]

[1]Research Scholar, NIMS, Shobha Nagar, Jaipur
beniwal_vikas23@rediffmail.com

[2]Ludhiana College of Technology, Ludhiana
lcet02ludhiana@rediffmail.com

Abstract

This paper comparator networks - a well-known model of parallel computation. This model is used extensively for keys arrangement tasks such as sorting and selection. This work investigates several aspects of comparator networks. It starts with presenting handy tools for analysis of comparator networks in the form of conclusive sets - non-binary vectors that verify a specific functionality. The 0-1 principle introduced by Knuth states that a comparator network is a sorting network if and only if it sorts all binary inputs. Hence, it points out a certain binary conclusive set. We compare these two models by considering several 0-1 -like principles and show that the min-max model is the 'strongest' model of computation which obeys our principles. That is, if a function is computable in a model of computation in which any of these principles holds, a min-max network can compute this function.

Keywords: *Computer Networks, Fuzzy Neural Networks, Fuzzy Reasoning.*

1. Introduction

Computer networks are becoming more abundant in today's business environments as they play a central role in maintaining and transmitting information. Many organizations have realized that ease of access to information is a critical need that can also build a competitive advantage if it is easily accessible. Networks play a central role in this concept for many reasons, with the most important being that they can help geographically dispersed organizations overcome the geographic obstacle.

The growing usage of computer networks is requiring improvements in network technologies and management techniques so that users will still be provided with high quality service. A major aspect of computer networks that is vital to quality of service is data routing.

As more individuals transmit data through a computer network, the quality of service received by the users begins to degrade. This indicates that more effective and adaptive measures must be developed for routing data through computer networks. Here we are developing an improved method for data routing. The primary tool applied in the routing method of this research was fuzzy reasoning. This was argued to be an appropriate technique for routing due to the imprecise measures currently used in present routing algorithms. Many of today's algorithms use various network measures, known as metrics, to establish the best path through a computer network. Few people have yet to recognize the nontrivial inaccuracies present in the measures. Increasing complexities and growth of computer networks is accelerating the significance of this notion.

To combat these inaccurate metrics, fuzzy reasoning was applied as the basis of the new algorithm. A secondary technique utilized was a neural network. The neural network was deemed suitable because it has the ability to learn. Once the neural network is designed, any alterations in the computer routing environment can easily be learned by this adaptive artificial intelligence method. The capability to learn and adapt is essential in today's rapidly growing and changing computer networks. These techniques, fuzzy reasoning and neural networks, when combined together provided a more effective routing algorithm for computer networks.

The principal objective is to demonstrate the advantages of applying fuzzy reasoning to routing data through a wide area network. Developing the new fuzzy routing algorithm involved many small processes, which were integrated to facilitate the modeling and testing required in the study. Simulation methods, neural network procedures, and fuzzy reasoning were all essential in achieving the research objective.

2. Importance of the research work

The uses of fuzzy systems will be discussed. Fuzzy logic criteria will be mentioned which can increase the network size. Fuzzification, fuzzy inferences, defuzzificiation through multi layers feed – forward.

3. Research Methodology

Fuzzy reasoning provides the foundation upon which the research is established. A second tool, neural networks, will be used to enhance the capabilities of the fuzzy reasoning process; therefore, it is important to provide thorough explanations of fuzzy reasoning and neural networks before presenting the routing methodology. The first section provides a brief introduction to fuzzy reasoning and how it differs from more common reasoning approaches. The second section discusses neural networks and how they are designed. This background information leads into the third section that provides an overview of the methodology to be implemented. Finally, the fourth section provides a brief conclusion and summary.

3.1 Fuzzy Reasoning

3.1.1 Fuzzy Introduction

Fuzzy reasoning refers to the superset of classical reasoning that has been augmented to recognize and manage imprecise information. This is accomplished through the use of fuzzy sets or fuzzy rules. A non-fuzzy set is a set that categorizes objects or information as either completely belonging to the set, or not belonging at all. A set of this nature has a precise boundary that defines what belongs to the set and what does not. A fuzzy set is one whose members either belong completely, partially, or not at all. The boundary that defines membership and non-membership in a fuzzy set is imprecise, thus allowing an object, or piece of information, to partially belong to a set. A fuzzy set was methodology for solving problems having uncertain, imprecise or vague descriptions. Because few people are comfortable defining exact set definitions for descriptive classifications, fuzzy reasoning has been a popular concept in the literature.

Expert systems have been a popular vehicle for applying fuzzy reasoning. Traditional expert systems use Boolean logic to reason through a decision-making process. The rules in atypical expert system are of the form: If *x* is low and *y* is high then *z* is medium. Using Boolean logic, *x* is a variable whose value is defined as either completely low or absolutely not low, and *y* is a variable that is defined as either completely high or not high at all. Based on the precise values of *x* and *y*, Boolean logic will determine whether or not *z* is a member of the set "medium" or the set "not medium".

Fuzzy expert systems use fuzzy reasoning as the underlying logic to analyze a particular situation. A fuzzy system once again applies rules of the form: If *x* is low and *y* is high then *z* is medium. However, using fuzzy reasoning, *x* is a variable defined as being completely low, not low at all, or somewhere in between those two extremes. Similarly, *y* is a variable defined as being completely high, not high at all, or somewhere in between the two extremes.

Depending on the strength of membership of x and y in their respective sets, z will be inferred to belong, to some extent, to the set "medium." Although fuzzy reasoning uses fuzzy sets to represent imprecise concepts, it does so in a very precise and well-defined manner.

3.1.2 Manipulating Fuzzy Sets

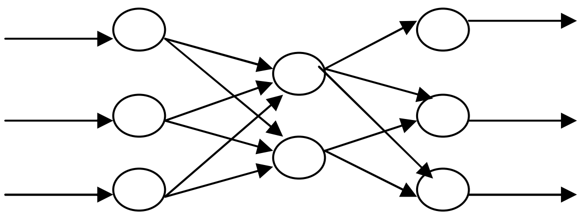
Once the fuzzy sets have been defined, several approaches are available for manipulating those sets. The first and most common is a rule based approach. This approach was employed in the previous attempt of fuzzy reasoning applied to network. The second approach involves using neural networks to process the information obtained from fuzzy sets. Neural networks are utilized in this study; therefore, an explanation for processing fuzzy sets with neural networks is provided in the next section. Appendix A contains a discussion of the rule based approach for the interested reader.

4. Neural Networks

A neural network is an artificial intelligence technique originally designed to mimic the functionality of the human brain. It is a non-algorithmic procedure that has a strong capability of learning and adapting to changes in its operating environment. The ability to successfully modify itself indicates neural networks could be a beneficial tool in managing volatile networks. Hence, this study chose this method to process the information obtained by the fuzzy sets.

A neural network is composed of many simple and highly interconnected processors called neurodes (Figure-1). These are analogous to the biological neurons in the human brain. The artificial neurodes are connected by links that carry signals between one another, similar to biological neurons. The neurodes receive input stimuli that are translated into an output stimulus.

A neural network consists of many neurodes connected together. This illustrates a three layer neural network, which is the type that will be designed for this study. However, it is possible to have more than (or less than) three layers and more than (or less than) nine neurodes in a neural network. Processing begins when information (input response) enters the input layer of neurodes in a neural network. Inputs entering any neurode in the neural network will follow the same basic process. This is a two-step process that uses two different mathematical expressions for evaluation. The first step utilizes a summation function to combine all input values to a neurode into a weighted input value. The second step utilizes a different mathematical expression, known as a transfer function that describes the translation of the weighted input pattern to an output response. This two-step process operates identically for all neurodes in the neural network. The summation function controls how the neurode will compute the net weighted input from the single inputs it has received. Although the summation function operates identically for all layers, summation outcomes for the input layer will be more direct than at the other layers. This is because each neurode in the input layer receives a single input value. Since the sum of a single value is equal to that original value, the net weighted inputs for these neurodes will be the original input values. Neurodes in the other two layers require some computation to obtain their net weighted inputs. This is accomplished using the following summation formula.



Input Layer Hidden Layer Output Layer

Figure 1 NN Architecture

$$X_j = \sum_{x=1}^{n} W_{xj} \ N_x$$

Xj symbolizes the net weighted input received by neurode *j* from *n* different neurodes that feed into neurode *j*. The input signal received from the x*th* neurode is symbolized by *xi* and *wij* designates the weight on the branch connecting node *i* to node *j* (Figure-2). The second step of the process is to convert the net input to an activation level, which will be the output of that neurode (neurode *j*). The activation level is obtained by applying the net input to a predefined S-shaped curve, the transfer function. The most common curve, and the one utilized in this study, is the sigmoid function:

$$f(I) = (1 + e^{-1})^{-1}$$

Two step procedures are executed for each individual layer of the neural network sequentially starting at the input layer. Once the output values for all neurodes on the input layer have been computed, these become the input values to the second layer. The two step process then continues with the second layer. The second layer neurodes all have an output value, then these outputs become input values for the third layer. This process continues until the output layer is reached, whereupon the output values computed at that layer become the final output response of the entire neural network.

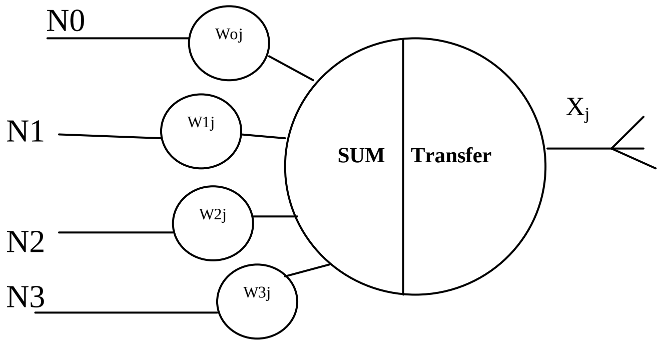


Figure 2 Structure of Neurode j

The neural network's learning procedure can be described as training by example. The most common procedure used for learning, is called supervised learning". This learning style begins with an example set of input and corresponding output patterns.

Summation:	$X_j = \sum_{x=1}^{n} W_{xj} \ N_x$
Transfer:	$Y_j = f(X_j)$

These input/output pairs are exposed to the neural network, which eventually learns the distinct type of output to expect upon receiving certain inputs. This training causes learning to occur by reducing the error produced when the neural network predicts an output from a given set of input values. The error reduction is accomplished by modifying the weights that connect the neurodes to one another. This is analogous to biological learning where the brain's synapses strengthen their connections between neurons upon learning. The weights of the artificial network are adapted according to a specified learning rule, which in this study is known as the delta learning rule.

4.1 Fuzzy Neural Networks

A neural network is considered a fuzzy neural network if the signals and/or the weights in the system are based around fuzzy sets. This paper will employ fuzzy neural networks whose signals are membership grades generated from fuzzy sets.

This alteration of the neural network inputs will not affect the neural network operations previously described.

In fuzzy neural network for routing a message needs to be sent from node A (source) to node G (destination). The first decision faced by the routing algorithm at node A will be to determine if the message should be transmitted through node B (link 1), node C (link 2) or node D (link 3). Determining a value for each of those three possible outgoing links will make this decision. These three values, computed by the proposed routing strategy, will represent the expected time to destination via node B (link 1), node C (link 2) and node D (link 3). These three time values will be compared and the link that gives the shortest expected time will be chosen as the first link in routing the message to the destination (node G).

That is, “short distance” is one fuzzy set. It happens to overlap with “medium distance” which is another fuzzy set. For a particular outgoing link and destination, we might have membership grades of 0.0 for “short distance”, 0.4 for “medium distance”, and 0.8 for “long distance”, meaning that the distance tends to be slightly more long than medium for this route. The source node will maintain a fuzzy neural network that will assess the time required for the data to reach the destination via that particular link. Therefore, this membership grade information needs to be conveyed to the neural network for each of our four metrics. Thus, three fuzzy sets for each of four metrics results in twelve fuzzy sets for each link considered see the Table.

Short Distance	**Medium Distance**	**Long Distance**
Low Throughput	Medium Throughput	High Throughput
Low Congestion	Medium Congestion	High Congestion
Mild Failure	Severe Failure	Very Severe Failure

Table 1: Twelve Fuzzy Sets

Data for a particular link (distance, throughput, congestion, failure) will be transformed into twelve fuzzy membership grades, one for each of the fuzzy sets, thus resulting in twelve inputs to the neural network. In addition to the twelve fuzzy membership grades, there will be two additional inputs to the neural network, namely the packet size and destination of the message

4.2 Neural Network Training And Design

Training the neural network will be the most time consuming phase of the study and will require that a simulation model first be designed. The input layer consisted of fifteen nodes corresponding to twelve fuzzy inputs, two crisp inputs and one bias input node. Although the membership grade inputs were all between zero and one, the crisp inputs ranged from 0 to 1500, requiring them to be scaled to fall between zero and one before training.

The output layer was comprised of a single node representing the expected time to reach the destination. Scaling, based on the transfer function, is suggested for the output node as well. The transfer function used was the sigmoid function (Figure), which is a continuous monotonic mapping of the input into a value between zero and one. The majority of values fall between 0.2 and 0.8. For this reason, the output values were scaled between 0.2 and 0.8.

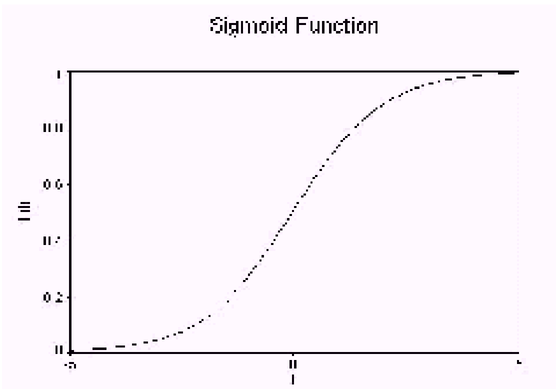


Figure: Sigmoid Function $y = (1 + e^{-1})^{-1}$

Determining the number of nodes in the hidden layer was accomplished using a popular rule of thumb:

Hidden Nodes = 0.5 * (Output Nodes + Input Nodes)

5. Conclusion

This paper comes out that this study is not focused on a specific problem. Instead, it is more of a walk through the field of comparator networks. The conclusion starts with the more general - computability and complexity of functions under the comparator networks model and the min-max model. It then looks into more specific functionalities studied here.

In this work, we study the min-max model by considering several 0-1 -like principles. It comes out that the min-max model is the ‘strongest’ model of computation which obeys our principles. That is, if a function is computable in a model of computation in which any of these principles holds, this function can be computed by a min-max network. Therefore, we find the min-max model to be more natural then the comparator model for solving key arrangement problems. This work shows that in some cases sorting Bitonic sequences can also be performed faster by a min-max model then by a comparator network. However, the depth difference is at most one. Some functions are computable only in the min-max model and not in the comparator networks model. This fact was also demonstrated by Knuth. We strengthen this result by providing an isomeric mapping that is computable by a min-max network and not by a comparator network.

A simulation model was designed following the development of the new algorithm that applied fuzzy reasoning enhanced by a neural network. The basis of the simulation was for comparing the new algorithm to a current routing algorithm based on the shortest route technique. Before the simulations could be employed, an experimental design having two factors was established. These two factors, congestion level and failure rate, were selected as primary factors in the experimental design because of their high correlation to routing level achieved. The level of congestion present in the computer network greatly affects the travel time for all types of data. Similarly, failure in the computer network can delay or completely stop the transmission of data. Each factor was divided into two levels, low and high; thus, leading to an experimental design having four sampling units. Each unit represented a different network situation under which a comparison test was performed between the two algorithms. The comparisons demonstrated that the new algorithm outperformed the shortest route algorithm in routing effectiveness under all network situations except an extremely stable one having low congestion and low failure rate. Nonparametric statistical tests were applied to establish significance at $\alpha = 0.10$ significance level. This was the expected result, and furthermore proves that the new algorithm has large potential benefits associated with it. The paucity of so-called stable networks being used today emphasizes the usefulness of this new algorithm.

	Low Congestion	**High Congestion**
Low Failure	Not Signification	$P < 0.10$
High Failure	$P < 0.10$	$P < 0.10$

Table 2: Significant P-values

References

[1] Pascal Auillans, Patrice Ossona de Mendez, Pierre Rosenstiehl and Bernard Vatant: A Formal Model for Topic Maps. Proceedings of the First International Semantic Web Conference, Sardinia, Italy, June 2002.

[2] Varol Akman and Müjdat Pakkan: Nonstandard Set Theories and Information Management. Journal of Intelligent Information Systems, Volume 6 Number 1, 1996, pp. 5-31.

[3] Serge Abiteboul, Dan Suciu, and Peter Buneman: Data on the Web: From Relations to Semistructured Data and XML. Morgan Kaufmann Publishers, October 1999.

[4] Scott Boag, Don Chamberlin, Mary Fernández, Daniela Florescu, Jonathan Robie and Jérôme Siméon, editors: XQuery 1.0: An XML Query Language. W3C Working Draft, August 2002. http://www.w3.org/TR/xquery/

[5] Tom Bellwood, Luc Clément, David Ehnebuske et al., editors: UDDI Version 3.0. Published Specification, July 2002. http://uddi.org/pubs/uddi_v3.htm

[6] Shawn Bauers and Lois Delcambre: Representing and Transforming Model-Based Information. Proceedings of the Semantic Web Workshop 2000, Lisbon, Portugal, September 2000.

http://www.ics.forth.gr/proj/isst/SemWeb/proceedings/session1-1/paper.pdf

[7] Dave Beckett, editor: RDF/XML Syntax Specification (Revised). W3C Working Draft, March 2002. http://www.w3.org/TR/rdf-syntax-grammar/

[8] Ciborra, Claudio (ed.) (2004). The Social Study of ICT;

[9] H. Huang, D. W. C. Ho, and J. Lam, "Stochastic stability analysis of fuzzy Hopfield neural networks with time-varying delays," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 52, no. 5, pp. 251–255, 2005.

[10] T. Hamel and R. Mahony, "Image based visual servo control for a class of aerial robotic systems," Automatica, vol. 43, no. 11, pp. 1975–1983, 2007.

[11] R. Sakthivel, K. Mathiyalagan, and S. M. Anthoni, "Design of a passification controller for uncertain fuzzy Hopfield neural networks with time-varying delays," Physica Scripta, vol. 84,

a. Article ID 045024, 2011.

[12] H. Li, B. Chen, Q. Zhou, and W. Qian, "Robust stability for uncertain delayed fuzzy Hopfield neural networks with Markovian jumping parameters," IEEE Transactions on Systems,Man, and Cybernetics, Part B, vol. 39, no. 1, pp. 94–102, 2009.

[13] C. K. Ahn, "Takagi-Sugeno fuzzy hopfield neural networks for $H\infty$ nonlinear system identification," Neural Processing Letters, vol. 34, no. 1, pp. 59–70, 2011.

[14] Ulrik Brandes, Markus Eiglsperger, Ivan Herman, Michael Himsolt, and M. Scott Marshall: GraphML Progress Report: Structural Layer Proposal. Proceedings of Ninth International Symposium on Graph Drawing, Vienna, September 2001. http://www.inf.uni-konstanz.de/~brandes/publications/behhm-gprsl-01.pdf