Journal of Sustainable Development in Africa (Volume 9, No.1, 2007)

ISSN: 1520-5509

Fayetteville State University, Fayetteville, North Carolina

Explicitly Representing Knowledge Embedded in Pattern Recognition Weighted Neural Network

Nontokozo Mpofu

Abstract

In this research work, we propose explicitly representing the knowledge embedded in a Pattern Recognition Weighted Neural Network (PRWNN) system in a "humanly comphrensible" form, using the concepts of Fuzzy Logic. The targeted audiences are the Zimbabwean system designers and/or system analyzers of (PRWNN) systems that are used as forecasting tools in the Meteorological Department, in Zimbabwe. We view the knowledge represented inside a PRWNN system as tacit, "hidden" knowledge, that is well understood and known by the system and it's weights. We then extract, refine and revise this trapped knowledge using the concepts of fuzzy logic, that is: fuzzy set, membership function, use of the connectives AND, OR and implication, the determination of action to be taken based on human determined fuzzy IF – THEN rules, combined with non-fuzzy rules and the fact that correctly trained networks can act as a Content Addressable Memory. The proposed method is tested on the weight matrix of already trained networks using past data obtained from the Meteorological Department in Zimbabwe.

Keywords: Fuzzy Logic, tacit knowledge, explicit representation

Introduction

Pattern Recognition Weighted Neural Network (PRWNN) systems have diverse applications, one of which is their use as forecasting tools, be it in the Meteorological Department for forecasting weather patterns or in the stock market industry for forecasting stock prices in the market. Such systems are loosely based on the model of the brain as a network of simple interconnected processing elements and they derive their power from the collective processing of artificial neurons, the chief advantage being that such systems can learn and adapt to a changing environment. However, a real problem faced by the designers and analyzers of such systems is that a lot of mathematics is used and humans find it difficult to interpret the numeric representation of the network, thus making the knowledge acquired by such systems not transferable to other knowledge representation schemes such as expert or rule-based systems. It also prevents users to gain better understanding of a classification task learned by the network. The key assumption of knowledge-based neurocomputing is that knowledge is obtainable from, or can be represented by, a neurocomputing system in a form that humans can understand [1] The knowledge embedded in a (PRWNN) system is captured in the weights between the neurons, thus making it difficult for local system designers and/or analyzers to extract, understand and share with others this knowledge. How the system reasoned to its conclusions? This leaves our weather forecasters with no choice but to resort to the traditional forecasting tools for forecasting weather patterns. Quite a lot of research work has been done as an attempt to extract and explicitly represent the knowledge embedded in a weighted neural network system in a way that humans can understand. This has been necessitated by the fact that the major criticism of all weighted neural networks is their opaque structure where the designer cannot easily interpret the information stored. Researchers have used various computer science techniques in an attempt to come out with a solution to the described problem. Robert L. Fry, in his paper "A theory of neural computation", incorporates the concepts of Boolean Algebra as he proposes a theory that is complementary to information theory and which provides a means of understanding biological neurons. [2] . Alina Lazar and Ishwar K. Sethi, in their paper, "Decision Rule Extraction from Trained Neural Networks Using Rough Sets" propose a method can help to make the knowledge embedded in a trained neural network comprehensible using rough sets. [3] In an attempt to understand the operations of a weighted neural network, Eyal Kolman and Michael Margaliot, in their paper "Knowledge Extraction from Neural Networks using the All-Permutations Fuzzy Rule Base", they use the mathematical equivalence between artificial neural networks and a specific fuzzy rule base to extract the knowledge embedded in the network. They demonstrate this using a benchmark problem: "The recognition of digits produced by a LED device". [4] As the complexity of the system increases, it becomes more difficult and eventually impossible to make a precise statement about it's behaviour, eventually arriving at a point of complexity where the fuzzy logic method born in humans is the only way to get at the problem. [5]. In this paper, we propose explicitly representing the knowledge trapped in a PRWNN that is used as a forecasting tool, using the concepts of fuzzy logic. The input to a PRWNN is an incomplete or distorted pattern that the system, through it's reasoning capacity that is stored in it's weights, has to identify. The output is the identified pattern. The process of identifying the incomplete or distorted pattern mimics human reasoning. In two-valued logic, each node output is a Boolean value, (0 or 1) or (-1 or 1) or (True or false) and the overall system output is considered as identified (if the pattern was identified) or not identified (if the pattern was not identified). However, in real world, objects are not conceived as having either one state or the other. Intermediate values are also incorporated such as the output pattern "closely resembles" the desired pattern (with a possibility of 0.8 occurrence). [6] Fuzzy Logic uses the whole interval between 0(False) and 1(True) to describe human reasoning, thus overcoming the limitations imposed by conventional Boolean Algebra.

Motivation

Apart from the fact that fuzzy logic uses the interval (0,1]) it also uses vague production rules such as:

IF (error is small) THEN (output is small)

The rules are generally linguistic representations and the information is transparent as the designer can easily interpret it. The power of the system lies in the way these humanly interpretable production rules are given a precise mathematical meaning and the manner in which it generalizes to produce appropriate outputs for previously unseen inputs. It is empirically based, relying on the operator's experience rather than their technical understanding of the system. [7]

Knowledge Extraction, refining and revising

In extracting the knowledge embedded inside the PRWNN, we follow a series of steps.

1: We define the following sets, the Classical Set X (which is the universal set of all the inputs to the system), the membership function μ and the fuzzy set on the classical set X. The definitions are given in continuation:

i) The classical set $X = \{x_1w_1, x_2w_2, \ldots\ldots\ldots x_nw_n\}$; where x_i is the ith input to that node and w_i is the weighting associated with that input

ii) The membership function: $\mu = [0,1]$

iii) The fuzzy set on the classical set X:

$$\tilde{A} = \{((x_i w_i), \mu_a(x)) / x \in X; i = 1,2,\ldots..n\} \qquad 3.1$$

2: We are then tasked with finding a way of representing the following in fuzzy logic, taking into account we do not loose the meaning and that the representation is in such a way that it can be understood by the system designer and /or the system analyzer.

i) Input representation of each node, that is:

$x_1w + x_2w_{2+} \ldots\ldots\ldots + x_nw_n = S$; where S is the total sum

ii) Comparison of the Sum of each node to it's threshold function T, that is:

Is $S > T$? Or $S < T$?

so as to determine whether the node will fire or not

iii) Way of comparing the network output with the actual output in the case where the expected pattern is known in terms of it's degree of membership.

3: Providing the answers

As a way of providing the answers to the above, we made the following analysis. The problem is targeted at providing a solution to the system designer and/or analyzer as to how the PRWNN actually reasons to come out with the conclusions that it arrives at. How it draws conclusions from only a distorted or incomplete pattern that is fed to it as input. Since a correctly trained network can act as Content Addressable Memory (CAM) and in this case, memory addressing is achieved solely by the content of memory and not by any label attached to a memory address (as in a serial computer), then a noisy pattern can be completed so that it ends up being identified to be a certain pattern. Just like the way a human would reason in trying to recall something with only the input as a hazy picture/view of that incident. The human mind will concentrate only on that section of the mind that contains the contents related to the event. Pieces of information that form or complete the picture are grouped together till the whole picture is clear/recalled. For situations i) and ii) above, we determine the action to be taken based on human determined IF – THEN rules combined with non-fuzzy rules. [8] That is we construct a rule-base composed of production rules of the form:

(R_{ij}): IF x_1 is A_1^i AND ……..AND x_n is A_n^i THEN y is B^j (c_{ij}) 3.2

where there are n inputs $x = (x_1,\ldots\ldots\ldots x_n)^T$ and a single output y. The terms A_k^i where k = 1,………n and B^j are linguistic variables that represent vague descriptions such as, in the case of two patterns, how closely the knowledge captured in the nodes contributes in identifying the required pattern (e.g *closely, resembles, exactly, almost, very different*) and associated with each rule is a variable $c_{ij} \in [0,1]$ that denotes the confidence in the rule being true.

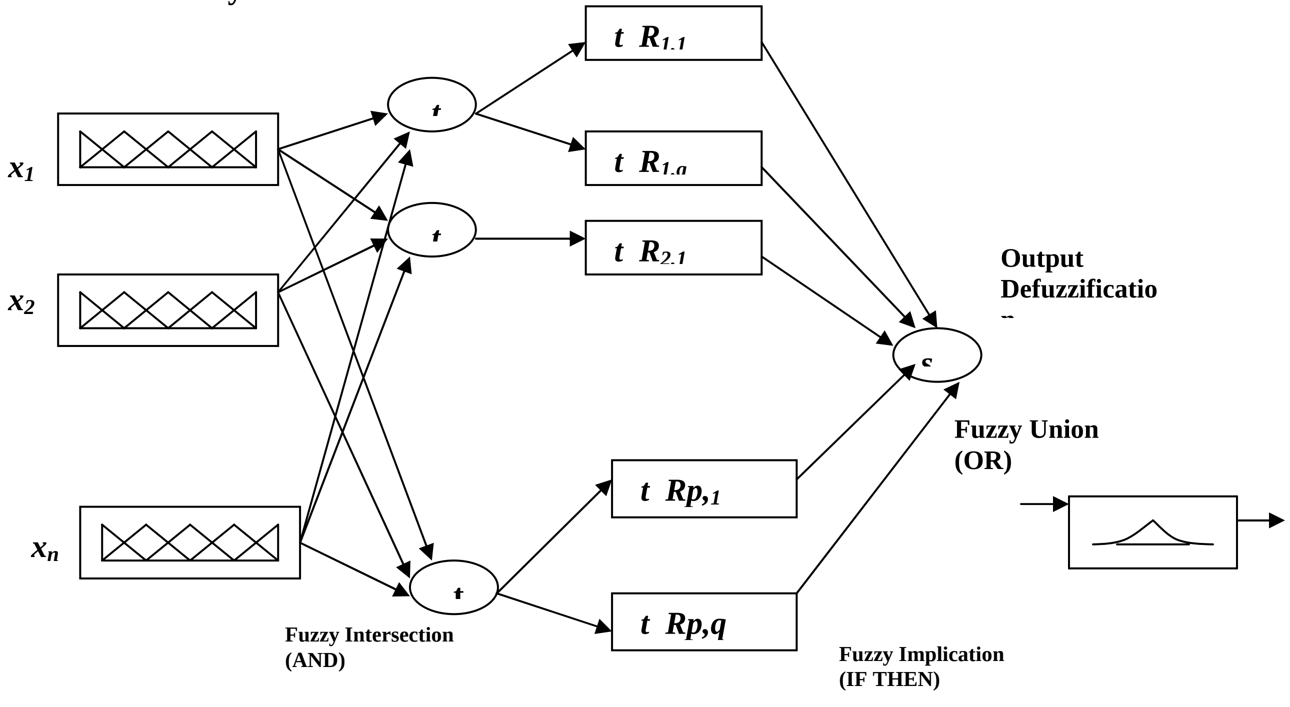
We incorporate the connectives intersection (AND), implication and union (OR) of fuzzy logic. The rule maps the antecedent, formed by the intersection (AND) of n linguistic statements x_k which is A_k^i , to the consequent formed by a single univariate linguistic element y which is B^j. A confidence of $c_{ij} = 0$ means that the rule will never fire, whereas if $c_{ij} > 0$ means that the rule will partially fire when the input is a partial member of the antecedent. The production rule describes the relationship between input and output and these are connected together using the fuzzy union (OR) operator to form the fuzzy algorithm.

Information flow through the system

- We first present an n-dimensional numerical input to the system
- We perform a process fuzzification using the singleton as our fuzzifier(since the information must first be represented as a fuzzy set)
- The system then performs as a intersection operation to decide the strength with which each rule antecedent will fire.
- The implication operator multivariate fuzzy input sets to univariate fuzzy output sets with a rule confidence that expresses the degree of belief in the rule
- The results is a set of active fuzzy output sets that are combined by the fuzzy union operator and if a numerical output is required, the resulting output membership function can be defuzzified.
-

We use the triangular norm t (which is the min or the product operators) to implement fuzzy intersection and implication and define fuzzy union by a triangular co-norm s (which we choose to be the max or the addition operators). This is depicted in the diagram below.

Information flow in the system



In cases where the network output constitutes a pattern that is known, the network output can be compared to the required pattern so as to tell the degree at which it resembles the desired pattern.

Conclusions

In this paper, we propose a method that we hope will help Zimbabwean system designers and/or system analyzers of PRWNN in understanding the knowledge captured in the system. The method eliminates the use of lots of mathematics which system designers and/or system analyzers find difficult to interpret and instead bases on human determined fuzzy IF – THEN rules to determine the action to be taken. We challenge the system designers and/or system analyzers of the PRWNN in the Meteorological Department to put into practice the proposed method, as it will aid them gain better understanding of the classification task learned by the network (the reasoning being undertaken by the network to reach it's conclusions), as a result, discarding the use of the traditional forecasting methods and resorting to the powerful knowledge acquisition tool (the PRWNN systems). With such understanding, we hope also that our system designers and/or analyzers will then be able to transfer the understood knowledge to other knowledge representation schemes such as expert or rule-based systems.

References:

Cloete I, Zurada M.J, *Knowledge-Based Neurocomputing,* February 2000

http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=3507

Fry R.L, *A theory of neural computation*, John Hopkins University, U.S.A, 2003

Lazar A and Sethi I, De*cision, Decision Rule Extraction from Trained Neural Networks Using Rough Sets,* Department of Computer Science, Wayne State University

Kolman E and Margalioty M, *Knowledge Extraction from Neural Networks using the All-Permutations Fuzzy Rule Base*, 2005

Sowell T, Fuzzy Logic for "*Just Plain Folks*", http://www.fuzzy-logic.com/

Jantzen J, Tutorial on fuzzy logic, http://www.iau.dtu.dk/~jj/pubs/logic.pdf

Taylor J.G, *Neural Networks*, Kings College London, 1995

Kaehler S, *Fuzzy Logic-An Introduction*

http://www.seattlerobotics.org/encoder/mar98/fuz/fl_part1.html#I

Abdullah M, Benest I, Evans A, Kimble C, *Knowledge Modelling Techniques for*

Developing Knowledge Management Systems, Department of Computer Science, University of York, Heslington, UK;

Brule J, Fuzzy Systems-A tutorial, 1985: http://www.austinlinks.com/Fuzzy/tutorial.html

Stergiou C and Siganos D, *Neural Networks*, :http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html

Smith L, An *Introduction to Neural Networks*, 1996: http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html