DATA CLASSIFICATION SYSTEM WITH FUZZY NEURAL BASED APPROACH

by

LUONG TRUNG TUAN

Dissertation submitted in partial fulfillment of
The requirements for the
Bachelor of Technology (Hons)
Information Technology

July 2005

Universiti teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

DATA CLASSIFICATION SYSTEM WITH FUZZY NEURAL BASED APPROACH

By

Luong Trung Tuan

Dissertation submitted in partial fulfillment of
The requirements for the
Bachelor of Technology (Hons)
Information Technology

Approved by,

____

Ms Vivian Yong Suet Peng
Supervisor

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

JULY 2005

CERTIFICATION OF ORGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledges, and that the original work contained herein have not been undertaken or done unspecified sources or persons.

Tuan

Luong Trung Tuan

ABSTRACT

Knowledge Discovery in Database and Data Mining use techniques derived from machine learning, visualization and statistics to investigate real world data. Their aim is to discover patterns within the data which are new, statistically valid, interesting and understandable.

In recent years, there has been an explosion in computation and information technology. With it have come vast amounts of data. Lying hidden in all this data is potentially useful information that is rarely made explicit or taken advantage. New tools based both on clever applications of established algorithms and on new methodologies, empower us to do entirely new things. In this context, data mining has arisen as an important research area that helps to reveal the hidden interesting information from the raw data collected.

The project demonstrates how data mining can address the need of business intelligence in the process of decision making. An analysis on the field of data mining is done to show how data mining can help in business such as marketing, credit card approval. The project's objective is identifying the available data mining algorithms in data classification and applying new data mining algorithm to perform classification tasks. The proposed algorithm is a hybrid system which applied fuzzy logic and artificial neural network, which applies fuzzy logic inference to generate a set of fuzzy weighted production rules and applies artificial neural network to train the weights of fuzzy weighted rules for better classification results.

The result of this system using the iris dataset and credit card approval dataset to evaluate the proposed algorithm's accuracy, interpretability. The project has achieved the target objectives; it can gain high accuracy for data classification task, generate rules which can help to interpret the output results, reduce the training processing. But the proposed algorithm still require high computation, the processing time will be long if the dataset is huge. However the proposed algorithm offers a promising approach to building intelligent systems.

ACKNOWLEDGEMENT

First and foremost I would like to take this opportunity to thanks to god because give me opportunity to finish this project. I also like to express and acknowledge my greatest gratitude to Ms. Vivian Yong Suet Peng, who had been given many supportive and constructive advices. Her constructive comments, advices and suggestion have guided me throughout this project.

Next I would like to acknowledge University Teknologi PETRONAS (UTP) specifically the final year research project committee for proper organization and management of this course.

Lastly I would like to acknowledge my parents for their endless support and confidence in me throughout the semester especially during conducting this project. They have given me the strength, motivation and determination in all my undertakings

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLE

CHAPTER 1

INTRODUCTION

1.1 Background of Study

Data Mining refers to extracting or "mining" knowledge from large amounts of data. The term is actually a misnomer. We used to refer the mining of gold from rocks or sand as gold mining rather than rocks or sand mining. Thus, "data mining" should have been more appropriately named "knowledge mining from data" which is unfortunately somewhat long. "Knowledge mining", a shorter term, may not reflect the emphasis on mining from large amounts of data. Nevertheless, mining is a vivid term characterizing the process that finds a small set of precious nuggets from a great deal of raw material. Thus, such a misnomer which carries both "data" and "mining" became a popular choice. There are many other terms carrying a similar or slightly different meaning to data mining, such as knowledge mining from databases, knowledge extraction, data/pattern analysis, data archaeology, and data dredging [5]. Data mining refers to the discovery step in knowledge discovery in database

The term Data Mining and "Knowledge Discovery in Databases" (KDD) are often used interchangeably, though KDD is more popular in academic world and Data Mining is the term most used in business world. There has recently been a move toward using "KDD" to indicate the whole process, and "Data mining to refer the discovery step. Knowledge discovery consists of an iterative sequence of the following steps [5]:

- Data cleaning (to remove noise or irrelevant data),
- Data integration (where multiple data sources may be combined),
- Data selection (where data relevant to the analysis task are retrieved from the database),

- Data transformation (where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance),
- Data mining (an essential process where intelligent methods are applied in order to extract data patterns),
- Pattern evaluation (to identify the truly interesting patterns representing knowledge based on some interestingness measures), and
- Knowledge presentation (where visualization and knowledge representation techniques are used to present the mined knowledge to the user).

Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks. In general data mining tasks can be classified into 2 categories: descriptive and predictive. Descriptive mining tasks characterize the general properties of the data in the database. Predictive mining tasks perform inference on the current data in order to make prediction [1]

- Data mining
    - a. Predictive
        - i. Classification: maps data into predefined groups or classes
        - ii. Regression: maps a data item to a real valued prediction value
        - iii. Time series analysis: the value of an attribute is examined as it varies over time
        - iv. Prediction: classify to predict the future state.
    - b. Descriptive
        - i. Clustering: classify into un-predefined class, which is defined by the data alone
        - ii. Summarization: maps data into subsets with associated simple description
        - iii. Association rules: uncovering relationships among data
        - iv. Sequence discovery: determine sequential patterns in data.

Data mining is a burgeoning and promising research field in computer science. It is a technique used to deduce useful and relevant information to guide professional decisions and other scientific research. It is a cost-effective way of analyzing large amounts of data, especially when a human could not analyze such datasets. Data mining is useful for telecommunications, medical science, professional institutions, business ventures, educational institutions and agriculture.

The current evolution of data mining functions and products is the result of years of influence from many disciplines, including databases, information retrieval, statistics, algorithm and machine learning, that are applied in mining unstructured data such as text, voice, and video; mining data in disturbed and heterogeneous databases; and mining data from the World Wide Web to help electronic commerce sites.

1.2 Problem Statement

Organizations today are overrun with data. One challenge often encountered is how to turn this massive amount of content into usable information. The modern world is a data-driven one. We are surrounded by data, numerical and otherwise, which must be analyzed and processed to convert it into information that informs, instructs, answers, or otherwise aids understanding and decision-making. In recent years there has been an explosive growth of methods for discovering new knowledge from raw data. In response to this, a new discipline of data mining has been specially developed to extract valuable information from such huge data sets. Given the proliferation of low-cost computers (for software implementation), low-cost sensors, communications, database technology (to collect and store data), and computer-literature application experts who can pose "interesting" and "useful" application problems. Data mining technology has recently become a hot topic for decision-makers because it provides valuable, hidden business and scientific "intelligence" from a large amount of historical data.

Many algorithms have been proposed and developed in data mining field. There are several challenges that data mining algorithm must satisfy to perform either descriptive or predictive tasks. The following are the criteria for evaluating data mining algorithms:

- Accurate: This is the most important properties of data mining algorithms; the usefulness of algorithms depends on its accuracy in either predictive or descriptive tasks
- Scalable: the algorithms must be able work on large scale of data, because there are many ways to store data fast and more convenient, organizations work with massive amount of data
- Interpretable: whether for descriptive or predictive tasks, data mining algorithms should provide more information to comprehend and explain the output results
- Versatile: the algorithms should be able apply on many type of data
- Fast: the processing time is also very important features

1.3 Objectives and Scope of Study

The objectives of this project are to:

- Research on the methodologies that has been applied on developing data mining system.
- Propose a new algorithm for classifying data to build a predictive model with the accuracy of 90% and above
- The proposed algorithm is interpretable, versatile, fast and scalable.

The System will assist users to extract hidden knowledge from raw data. A sample iris dataset and credit card approval dataset is used to demonstrate the capability of the proposed algorithm.

CHAPTER2

LITERATURE REVIEW

2.1 Knowledge Discovery Framework

Knowledge discovery is a process that involves many different steps. The input to this process is the data and the output is the useful information desired by the users. In this chapter we are presenting the methodology of applying the information-theoretic approach to real world problems. Fayyad [1996] and Pyle [1999] model are used to illustrate the overall knowledge discovery process

- The discovery stages covered here include:
- Understanding the problem Domain
- Obtaining and understanding data
- Preparation of the data
- Construction of knowledge model from data
- Evaluation of Model
- Using Model

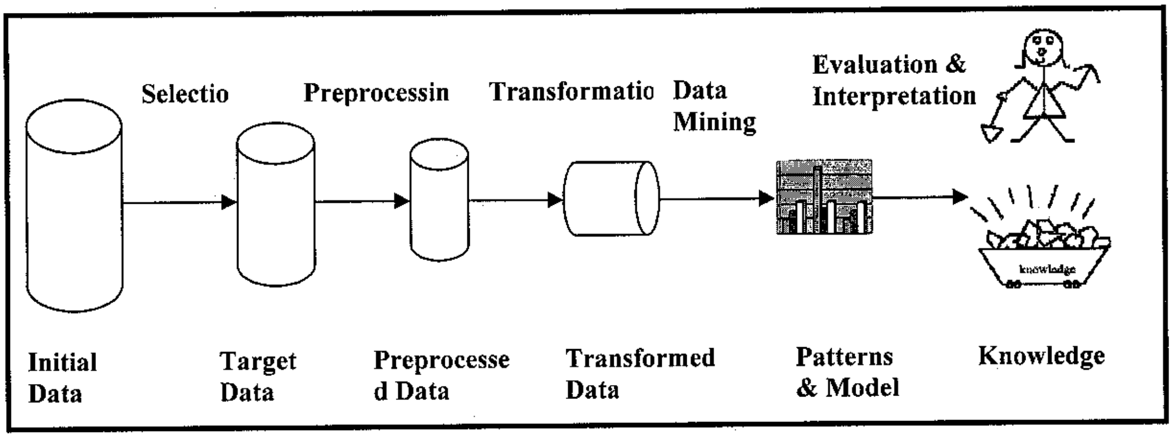


Figure 2.1: Knowledge Discovery Process

2.1.1 Understanding the Problem Domain

A minimum amount of background knowledge is essential for exploring data coming from any domain. The purpose of the Knowledge Discovery process is to enrich the existing knowledge, but starting from zero does not make practical sense. According to (Pyle, 1999), identification of the business problem is the most important part. Thus, universities are interested in finding the most successful admission criteria, manufacturers are concerned about the quality of their products, and doctors are looking for the most accurate diagnose.

2.1.2 Obtaining and Understanding the Data

The data needed for the data mining process may be obtained from many different and heterogeneous data sources. This first step obtains the data from various databases, files.

Once the data is obtained, the following issues will be addressed:

- **Identifying the candidate input and the target attributes:** The minimum set of input attributes will be determined in the process of constructing the model
- **Scale of numeric attributes**: knowing the scale and units of measurement can help in identifying potential outliers.
- **Interpretation of codes:** Values of nominal attributes are usually represented by numeric or alphanumeric codes. The meaning of each code can be provided manually or in the form of auxiliary tables
- **Obtaining auxiliary tables:** besides the main table, which contains the values of candidate input and target attributes, the database can include additional tables that are related to the main table through the values of certain attributes.
- **Data auditing:** we need data of sufficient quantity and quality

2.1.3 Preparation of the Data

Pyle (1999) provides a comprehensive coverage of existing data preparation techniques, applicable to different data models.

i. Treatment of missing values

Missing and empty values are denoted in relational databases by a special character ("null"). Some Meta data is needed to differentiate between meaningful missing values and meaningless missing values. In the first case the missing values can be assigned a special code or number, for example a missing test grade means that test has not been taken by the student. In the second case we can try to estimate the missing values, for example assigning the most frequent value of the same attribute. Attributes with large missing values can be removed from the analysis

ii. Data transformation

Original attribute values can be changed to increase the predicting power of the model. Several transformation methods are:

- Discrimination
- Re-scaling of continuous attributes
- Grouping of nominal attributes
- Decoding of complex attributes

2.1.4 Constructing the Knowledge Model from Data

The application of the data mining method to a preprocessed dataset results in the following outputs:

- **Network Structure:** the network structure is represented by the list of network nodes, including the layer associated with each node and the numbers of node leaves or the predicted values. Each leave is related to a value or an interval of the corresponding input attribute.
- **Association rules**: The rules extracted from the network connections are represented with their weights expressing the contribution of each rule to the overall mutual information

2.1.5 Evaluation the Model

The resulting model can be evaluated from the following aspects:

- Amount of dimensionality reduction: This can be measured by the proportion of selected input attributes out of the total number of candidate input attributes
- Network size: number of nodes
- Computational effort:
- Prediction accuracy
- Maximal prediction accuracy
- Cross-validation accuracy

2.2 Data mining Algorithm

Data mining is about predicting and describing. There are several algorithms for classifying and clustering data:

2.2.1 Classification by decision tree induction

A decision tree is a flow-chart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distributions. The top most node in a tree is the root node [5]

Table2.1: Sample data for generating decision tree

Name	Gender	Height
Kristinia	F	1.6
Jim	M	2
Maggie	F	1.9
Martha	F	1.87
Stephanie	F	1.7
Bob	M	1.85
Kathy	F	1.6
Dave	M	1.7
Steven	M	2.1
Todd	M	1.95
Amy	F	1.8
Wynette	F	1.65

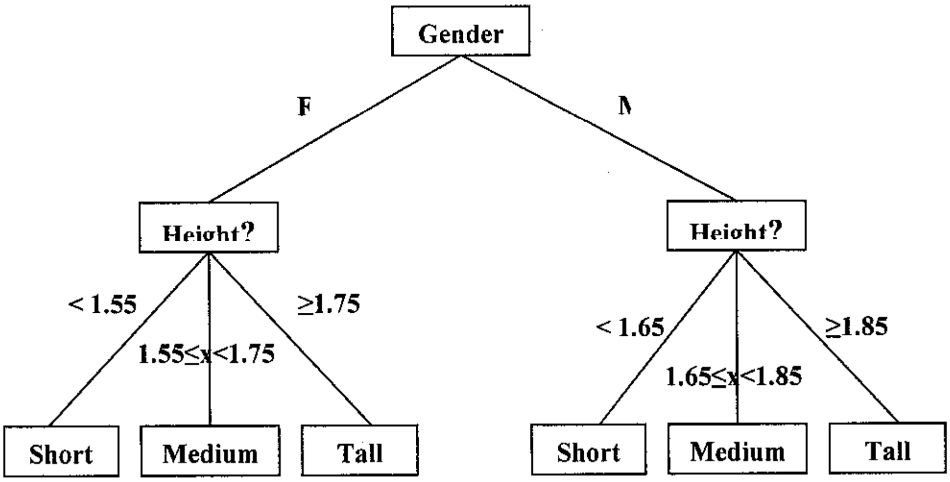


Figure 2.2: Sample decision tree

Decision tree algorithm

- ID3: The ID3 (Quinlan, 86) is to build a decision tree based on information theory and attempts to minimize the expected number of comparisons. The basic idea of the induction algorithm is to ask questions whose answers provide the most information.
- C4.5 and C5.0 (Quinlan, 93): successors of ID3, that improve the following ways: missing data, continuous data, pruning, rules, and splitting
- CART Classification and regression tree: generates a binary decision tree
- SPRINT Scalable Parallelizable induction of decision tree: address the scalability issue by ensuring that the CART technique can be applied regardless of the availability of main memory.

2.2.2 Association Rule Mining

Association rule mining finds interesting association or correlation relationships among a large set of data items. With massive amounts of data continuously being collected and stored, many industries are becoming interested in mining association rules from their databases. Atypical example of association rule mining is market basket analysis. This process analyzes customer buying habits by finding associations between the different items that customer place in their "shopping basket".

A famous algorithm for association rule mining is apriori algorithm. Apriori algorithm is an efficient association rule mining algorithm. It uses prior knowledge of frequent itemsets properties. It employs an iterative approach known as a level-wise search, where k-itemsets are used to explore (k+1)-item sets.

2.2.3 Bayesian classification

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities. It assumes that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computation and in this sense considered to be "Naïve".

2.2.4 Fuzzy Logics

Experts usually rely on common sense when they solve problems. They also use vague or ambiguous terms. For example, an expert might say, "Though the power transformer is slightly overloaded, I can keep this load for a while". Other experts have no difficulties with understanding and interpreting this statement because they have the background knowledge to hearing the problem described like this. However e knowledge engineer would have difficulties providing a computer with the same level of understanding. How can we present expert knowledge that uses vague or ambiguous terms in a computer? [29]

Fuzzy logic is not logic that is fuzzy, but logic that is used to describe fuzziness. It is the theory of fuzzy sets, sets that calibrate vagueness. Fuzzy logic was introduced by Jan Lukasiewicz in the 1920s While classical logic operates with only two values 1 (true) and 0 (false), Lukasiewicz introduced logic that extended the range of truth values to all real numbers in the interval between 0 and 1; he used a number in this interval to represent the possibility that a given statement was true or false. For example, the possibility that a man 181c, tall is really tall might be set to a value of 0.85. It is likely that the main is tall. Later in 1937, Max Black published a paper called "Vagueness: an exercise in logical analysis'; in this paper he argued that a continuum implies degree. In 1965 Lofti Zadeth, Professor and Head of Electrical Engineering Department at the University of California at Berkeley, published his famous paper "Fuzzy logic". In fact, Zadeth rediscovered fuzziness, identified and explored it, and promoted and fought for it. He extended the work of possibility theory into a formal system of mathematical logic, and even more importantly, he introduced a new concept for applying natural language terms. This new logic for representing and manipulating fuzzy terms was called fuzzy logic.

2.2.5 Artificial Neural network

An artificial neural network consists of a number of very simple and highly interconnected processors, also called neurons, which are analogous to the biological neurons in he brain. The neurons are connected by weighted links passing signals from one neuron to another. Each neuron receives a number of signals through its connections; however, it never produces more than a single output signal. The output signal is transmitted through the neuron's outgoing connection. The outgoing connection splits into a number of branches that transmit the same signal, and these braches terminate at the incoming connections of the neurons in the network.

Solving a classification problem using neural network involves several steps

1. Determine the number of input and output nodes, the number of hidden layer as well as the number of node per hidden layer
2. Determine weights and activation functions
3. For each tuple in the training set, propagate it through the network ad evaluate the output prediction to the actual result. Adjust the weight
4. For each tuple t_i in D, propagate through the network and make appropriate classification

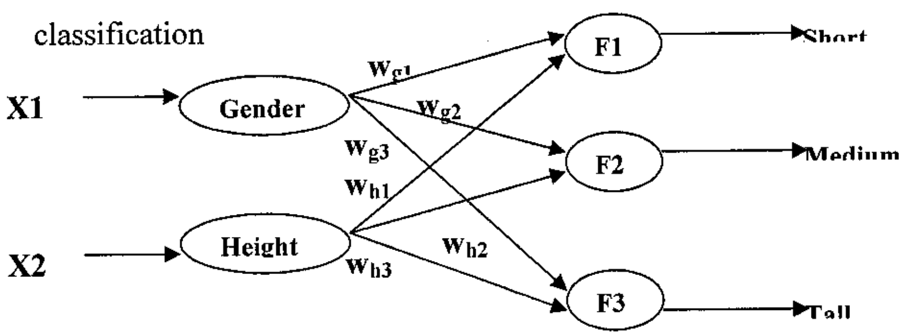


Figure 2.3: Artificial Neural Network

Issues in neural network

- Attributes: number of input and output node
- Number of hidden layer
- Number of hidden nodes per hidden layer
- Training data
- Interconnection: the connection between nodes, in the simplest case, each node is connected to all the nodes in next level
- Weights
- Activation function
- Learning technique
- Stop condition

Advantages to use neural network for classification

- Neural networks are more robust than decision tree because of the weights
- Neural networks are more robust than decision tree in noisy environment
- Neural network improve its performance by learning
- There is low error rate and high degree of accuracy

Disadvantage of neural network for classification

- Neural network is difficult to understand, while it's easy to explain decision tree
- Generating rules from decision tree is more straightforward, as compare with neural network
- Input attribute must be numeric
- Testing
- Verification
- The learning phase may fail to converge

2.3 Literature Review on Fuzzy Logic and Fuzzy rules induction

2.3.1 Fuzzy Set

A classical set is a set with crisp boundary. For example, a classical set A of real numbers greater than 6 can be expressed as

$$A = \{x|x>6\}, \tag{2.1}$$

where there is a clear, an ambiguous boundary 6 such that if x is greater than this number than x belongs to the set A; otherwise x does not belong to the set. Although classical sets are suitable for various applications, they do not reflect the nature of human concepts and thoughts, which tend to be abstract and imprecise. As an illustration, mathematically we can express the set of tall persons as a collection of persons whose height is equal or more than 6ft; this could classify a person 6 ft tall as a tall person, but not a person 5.999 ft tall. This distinction is intuitively unreasonable. The flaws come from the sharp transition between inclusion and exclusion in a set.

In contrast to a classical set, a fuzzy set is a set without a crisp boundary. That is the transition from "belong to a set" to "not belong to a set" is gradual, and this smooth transition is characterized by membership functions that give fuzzy sets flexibility in modeling commonly used linguistic expression, such as "the water is hot" or the "temperature is high".

Definition:

If X I a collection of objects denoted generically by x, then a fuzzy set A in X is defined as a set of ordered pairs:

$$A = \{(x, \mu_A(x)) \mid x \in X\}, \tag{2.2}$$

Where $\mu_A(x)$ is called the membership function (MF) for the fuzzy set A. The MF maps each element of X to a membership grade between 0 and 1.

2.3.2 Linguistic variables:

At the root of fuzzy set theory lies the idea of linguistic variables. A linguistic variable is a fuzzy variable. For example, the state 'Tom is tall' implies that the linguistic variable Tom takes the linguistic value tall. In fuzzy expert systems, linguistic variables are used in fuzzy rules. For example, 'If wind is strong then sailing is good'

2.3.3 Operation of fuzzy set

- **Complement:** The complement of a set is an opposite of this set. For example, if we have a set of tall man, its complement is the set of not tall man. If A is the fuzzy set, its complement ¬A can be found as follows:

  $\mu_{\neg A}(x) = 1 - \mu_A(x)$ (2.3)

- **Intersection:** In classical set theory, an intersection between two sets contains the elements shared by these sets. In fuzzy sets, an element may partly belong to both sets of each element:

  $\mu_{A\cap B}(x) = \min[\mu_A(x), \mu_B(x)] = \mu_A(x) \cap \mu_B(x)$ where $x \in X$ (2.4)

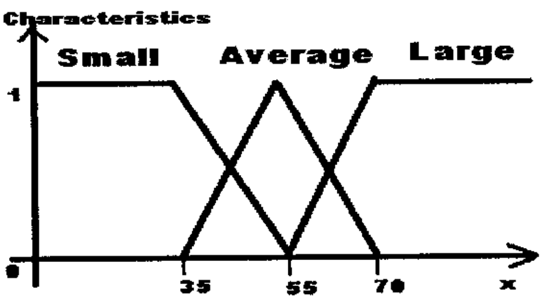
- **Union:** The union of two crisp sets consists of every element that falls into either set. In fuzzy set, the union is reverse of intersection. That is the union is the largest membership value of the element in either set:

  $\mu_{AUB}(x) = \min[\mu_A(x), \mu_B(x)] = \mu_A(x) \cup \mu_B(x)$ where $x \in X$ (2.5)

2.3.4 Membership Formulation

A fuzzy set is completely characterized by its membership function. The fuzzy membership functions can be determined by several possible methods. The most popular method is the heuristic method where pre-defined shapes will be chosen to represent certain linguistic terms. The most popular functions are piecewise linear functions such as triangular and trapezoidal membership functions due to their computational efficiency.

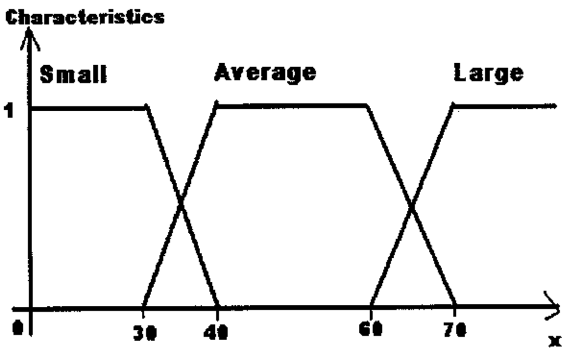
- **Triangular membership function:** A triangular membership function is specified by three parameters {a, b, c; a ≤b ≤c} as follow



$$\Delta(x,a,b,c) = \begin{cases} 0, x \le a \\ \dfrac{x-a}{b-a}, a \le x \le b \\ \dfrac{c-x}{c-b}, b \le x \le c \\ 0, c \le x \end{cases}$$

Figure 2.4: Triangular membership function

- **Trapezoidal membership function:** A trapezoidal membership function is specified by four parameters {a, b, c, d; a ≤b ≤c ≤ d} as follow



$$Trapezoid(x,a,b,c,d) = \begin{cases} 0, x \le a \\ \dfrac{x-a}{b-a}, a \le x \le b \\ 1, b \le x \le c \\ \dfrac{d-x}{d-c}, c \le x \le d \\ 0, d \le x \end{cases}$$

Figure 2.5: Trapezoidal membership function

2.3.5 Fuzzy Rule

A fuzzy if-then rule assumes the form

If x is A then y is O, (2.6)

where A and B are linguistic values defined by fuzzy sets on universe of discourse X and Y, respectively. "x is A" is called the antecedent or premise, while "y is O" is called the consequence or conclusion. Example of fuzzy if-then rule is "if mother is tall then the son is tall'

Fuzzy rule can have multi antecedent as the following form

If x_1 is A_1 and/or x_2 is B_1 then y_1 is O_1, (2.7)

Example: "if mother is tall and father is tall then the son is tall'

2.3.6 Fuzzy rule Induction – Subsethood based rule generation Algorithm

Although human experts have played an important role in the development of conventional fuzzy systems, automatically generating fuzzy rules from data is very helpful when human experts are not available and may even provide information not previously known by experts. Several approaches have been devised to develop data-driven learning for fuzzy rule based systems. They involve the use of a method that automatically generates membership functions of fuzzy rule structures of both from training data. Chen [18] proposed a method for generating fuzzy rules for classification problems; this method uses fuzzy subsethood values for generating fuzzy rules.

(a) Identify the nature of classification problem

(b) Define fuzzy partitions for the input variables and output variables according to the type of data and the nature of classification problems

(c) Transform crisp value into fuzzy input value

(d) Generate a set of fuzzy rules based on a fuzzy linguistics model

(e) Apply the rule sets for classification

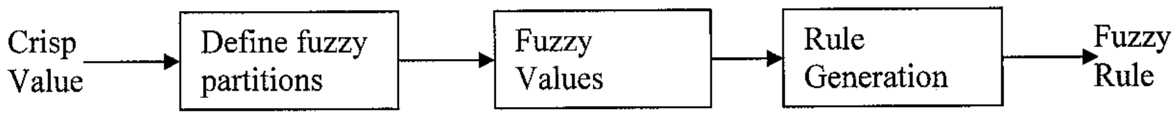


Figure 2.6 Subsethood based fuzzy rule induction process

The method for generating fuzzy rule models based on fuzzy subsethood values which are formally defined as follow. Let A and B be two fuzzy set defined on the universe. The fuzzy subsethood value of A with regard to B, **S (B, A)** represents the degree to which A is subset of B

$$S(B,A)=\frac{\sum_{x\in U}\nabla(\mu_B(x),\mu_A(x))}{\sum_{x\in U}\mu_B(x)}, \tag{2.8}$$

where S(B,A) Є [0,1] and ▼is the t-norm min operation

The purpose of generated rules is to handle classification problems. Subsethood based algorithm involves three main steps

i) Classify data into subgroups according to the underlying classification results

ii) Calculate fuzzy subsethood values for every variable in each subgroup

iii) Create rules

2.3.7 Weighted Subsethood based rule generation algorithm

Regarding the relative weight for linguistic term, Khairul (2002) has proposed the weighted subsethood based algorithm for data classification. Weighted subsethood based algorithm is the use of subsethood values as relative weights over the significance of different conditional attributes. Suppose that the subsethood value for certain linguistics tem A_1 of linguistics variable A with regard to classification X is $S(S,A_1)$ and that the linguistic variable A has the following possible linguistics terms $A_1, A_2, \ldots, A_n$. Then the relative weight for linguistics term A_i with regard to classification X is

$$w(X, A_i) = \frac{S(X, A_i)}{\max\limits_{j=1..n} S(X, A_{j)}}, \tag{2.9}$$

Clearly $w(X, A_i) \in [0, 1]$ with $i = 1...n$. This allows the creation of weight for each linguistic term per condition attribute. Intuitively the linguistic term with the highest subsethood value will be the most important and that with the lowest will be the least important

Consider the fuzzy rules with multiple inputs and a single output. These rules can be written in the following form

IF X_1 is (A_{11} OR A_{12} OR ... OR A_{1n}) **AND** X_2 is (A_{21} OR A_{22} OR ... OR A_{2n}) **AND** ... **AND** X_m is (A_{m1} OR A_{m2} OR ... OR A_{mn}) **THEN** Y is O_1 (2.10)

We can see that all the linguistic terms of each attribute are used to describe the antecedent of each rule initially. This may look tedious, but the reason keeping this complete form is that every linguistic term may contain important information that should be taken into account.

However, we can apply the relative contributions made by individual linguistic term of each variable towards the eventual conclusion drawn by multiplying each linguistic term by its respective weight; the fuzzy rule to be generated will be of the form

IF X_1 is ($w(X_1,A_{11})*A_{11}$ OR $w(X_1,A_{12})*A_{12}$ OR ... OR $w(X_1,A_{1n})*A_{1n}$) **AND** X_2 is ($w(X_1,A_{21})*A_{21}$ OR $w(X_1,A_{22})*A_{22}$ OR ... OR $w(X_1,A_{2n})*A_{2n}$) **AND** ... **AND** X_m is ($w(X_1,A_{m1})*A_{m1}$ OR $w(X_1,A_{m2})*A_{m2}$ OR ... OR $w(X_1,A_{mn})*A_{mn}$) **THEN** Y is O_1 (2.11)

For the rule evaluation, "OR" is interpreted as t-norm max operation and "AND" is interpreted as t-norm min operation

2.4 Fuzzy neural network

Fuzzy logic and neural networks are complementary tools in building intelligent systems. While neural networks are low-level computational structures that perform well when dealing with raw data, fuzzy logic deals with reasoning on a higher level, using linguistic information acquired from domain experts.

Fuzzy systems lack the ability to learn and cannot adjust themselves to a new environment. On the other hand, although neural networks can learn, they are opaque to the user.

The merger of a neural network with fuzzy system into one integrated system therefore offers a promising approach to building intelligent systems. Integrated neuro-fuzzy systems can combine the parallel computational and learning abilities of neural networks with the human-like knowledge representation and explanation abilities of fuzzy system. As a result neural networks become more transparent, while fuzzy systems become capable of learning.

2.4.1 Structure of Fuzzy Neural System

The structure of fuzzy neural system is similar to a multi-layer neural network. It has input and output layers and three hidden layers the represent membership function and fuzzy rules

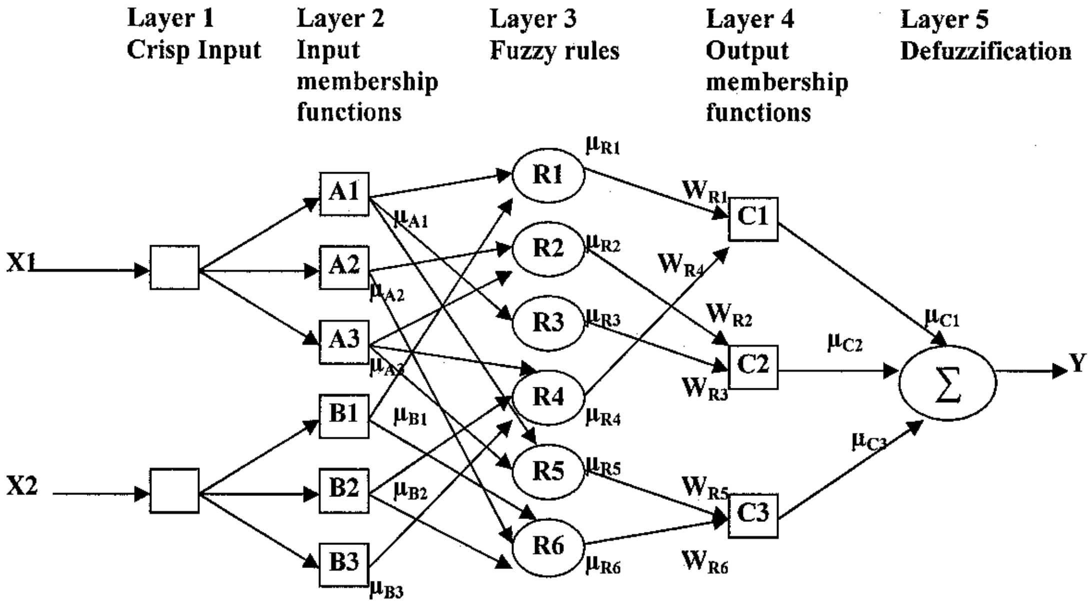


Figure 2.7: fuzzy neural network

- **Layer 1 Input layer:** Each neuron in this layer transmits external crisp signals directly to the next layer $Y^{(1)}_I = X^{(1)}_I$, where $X^{(1)}_I$ is the input, $Y^{(1)}_I$ is the output of neuron I in layer 1
- **Layer 2 input membership layer:** Neurons in this layer represent fuzzy sets used in the antecedents of fuzzy rules. A fuzzification neuron receives a crisp input and determines the degree to which this input belongs to the neuron's fuzzy set
- **Layer 3 fuzzy rule layer:** Each neuron in this layer corresponds to a single fuzzy rule
- **Layer 4 output membership layer:** Neurons in this layer represent fuzzy sets used in the consequent of fuzzy rules
- **Layer 5 Defuzzification layer:** Each neuron in this layer represents a single output of the neuron-fuzzy system.

2.4.2 Learning in fuzzy neural network

Multilayer neural network can learn, more than a hundred different learning algorithms are available, but the most popular method is back- propagation. This method was first proposed in 1969 (Bryson and Ho, 1969), but was ignored because of its demanding computations. Only in the mids-1980s was the back-propagation learning algorithm rediscovered

Learning with back-propagation algorithm has two phases. First, a training input pattern is presented to the network input layer. The network then propagates the input pattern from layer to layer until the output pattern is generated by the output layer. If this pattern is different from the desired output, an error is calculated and then propagated backwards through the network from the output layer to the input layer. The weights are modified as the error is propagated.

To propagate error signals, we start at the output layer and work backward to the hidden layers. The error signal at the output of neuron k at iteration p is defined by

$$e_k(p) = y_{(d,k)}(p) - y_k(p) \tag{2.12}$$

where $y_{d,k}(p)$ and $y_k(p)$ is the desired and actual output at output neuron k, iteration p respectively

After calculating the errors at output layer then we update the weight at output layer and hidden layers:

For output layer

$w_{j,k}(p+1) = w_{j,k}(p) + \Delta w_{j,k}(p)$, where $\Delta \mathrm{w}_{\mathrm{j,k}}(p)$ is the weight correction

$\Delta w_{j,k}(p) = \alpha * y_i(p) * \delta_k(p)$, where $\delta_k(p)$ is the error gradient

The error gradient is determined as the derivative of the activation function multiplied by the errora at the neuron output

$$\delta_k(p) = \frac{\partial y_k(p)}{\partial X_k(p)} * e_k(p) \tag{2.13}$$

where $y_k(p)$ is the output of neuron k at iteration p, and $X_k(p)$ is the weighted input to neuron k at the same iteration

For hidden layer

The weight correction can be calculated as following

$$\Delta w_{i,j}(p) = \alpha * x_i(p) * \delta_j(p) \tag{2.14}$$

where $\delta_j(p)$ represents the error gradient at neuron j in the hidden layer

$$\delta_j(p) = y_j(p) * [1 - y_j(p)] * \sum_{k=1}^{l} \delta_k(p) w_{j,k}(p) \tag{2.15}$$

Where l is the number of neurons in the output layer

$Y_j p)$ is the activation function for neuron j at hidden layer

$X_j(p)$ is the input for neuron j at hidden layer

$X_i(p)$ is the input for neuron I at the previous layer

In fuzzy neural network, we are training the weight for fuzzy production rule. Fuzzy logic operation is and, or which is interpreted as min and max arithmetic operation respectively, these arithmetic operation is not differentiated. In order to apply back-propagation and get better learning performance, we establish our training algorithm by using the smooth derivative [28] which is briefly described as follows

$$\begin{gathered} \frac{\partial(y \wedge p)}{\partial y} = \begin{cases} 1, y \leq p \\ 0, y \rangle p \end{cases} \\ \frac{\partial(y \vee p)}{\partial y} = \begin{cases} 1, y \geq p \\ 0, y \langle p \end{cases} \end{gathered} \tag{2.16}$$

Are regarded as the crisp truth degree of the proposition "y is less than or equal to p" and the crisp truth degree of the proposition "y is greater than or equal to p" respectively. To improve the performance of training, these crisp behaviors will be replaced by fuzzy behaviors being able to capture the real meaning of ($y \leq p$) and ($y \geq p$) in a vague context

$$\begin{gathered} \frac{\partial(y \wedge p)}{\partial y} = \begin{cases} 1, y \leq p \\ p, y \rangle p \end{cases} \\ \frac{\partial(y \vee p)}{\partial y} = \begin{cases} 1, y \geq p \\ y, y \langle p \end{cases} \end{gathered} \tag{2.17}$$

CHAPTER 3

METHODOLOGY

3.1 Software Development Cycle

The "waterfall model' is chosen as the methodology of this project. The phases of waterfall model are illustrated as in figure 3.1:

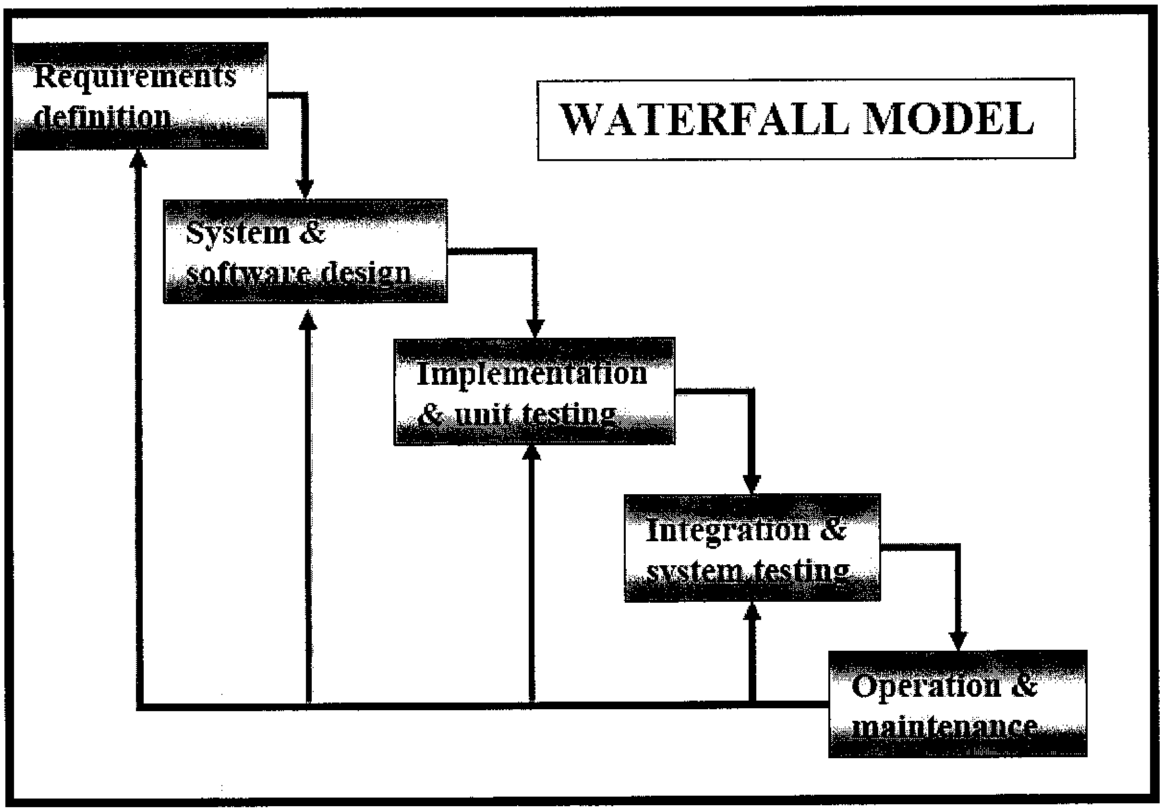


Figure 3.1 Waterfall model

The waterfall approach emphasizes a structured progression between defined phases. Each phase consists of a definite set of activities and deliverables that must be accomplished before the following phase can begin. The first phase tries to capture *What* the system will do (its requirements), the second determines *How* it will be designed, in the middle is the actual programming, the fourth phase is the full system *Testing*, and the final phase is focused on *Implementation* tasks such as go-live, training, and documentation. (Marks, 2002). For the purpose of this project, only the first four stages in this model will be used to develop the system prototype. The following sections will further explain activities involved phases of this model.

3.1.1 Requirement Definition

It is important that the requirements of the project be clearly analyzed and defined in detail so that they can serve as project specification. This phase involves with the identification of the background of the study and the definition of the objective and scope of the project.

From this specification, a literature review is conducted to identify the available data classification methods and suitable algorithm to tackle the identified requirements. The required prototype will classify data with the accuracy of 90% and above; and also generate rules which can help to interpret the output results. Requirements specification also helps in the design and planning of the software development.

3.1.2 System and Software Design

During the design stage, the system architecture will need to be established to identify and describe the fundamental software abstractions and their relationships.

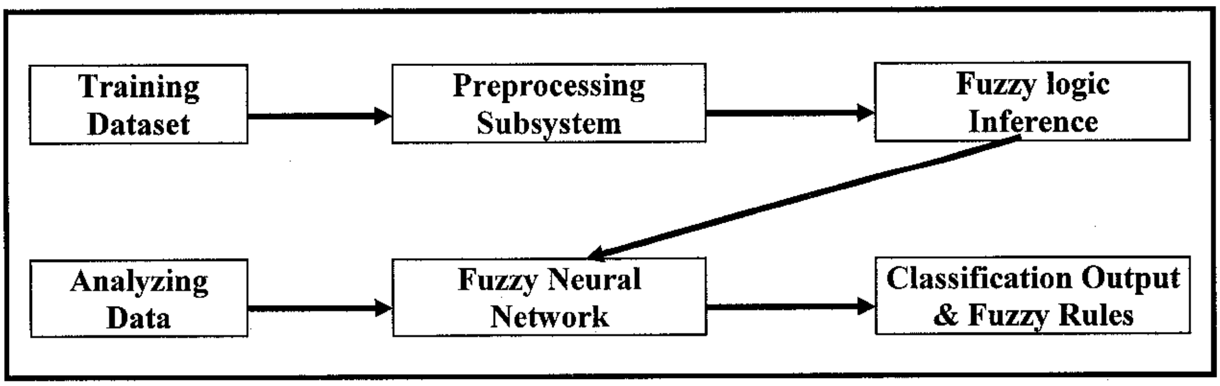


Figure 3.2 System architecture

The developing system is a supervised learning system. We need to feed the training dataset to train the system in order to generate the weighted fuzzy production rules (WFPR), and then these weighted fuzzy production rules will be trained by a min-max neural network to achieve the better classification accuracy. The system shall produce a new set of weighted fuzzy production rule so that we can feed the analyzing data into the system and apply the new set of WFPR to classify data.

3.1.2.1 Weighted fuzzy production rules induction subsystem

i. Define input fuzzy member class

To illustrate the methodology, we use the iris data set. Iris data set is the basic data set that every data mining algorithms at first will be tested on. Iris dataset has for candidate input the sepal length, sepal width, petal length and petal width, these candidate inputs are numerical data, based on those we can classify iris flower into 3 kinds: setosa, versicolor, and virginica.

We need to define linguistic term for linguistic variable. This step depends on the target classification class and the data type of linguistic variable.

- If the data type of input attribute is nominal, so the available category will become the linguistic term and the number of linguistic term is independent with the number of target class. For example, if we have attribute gender, which is nominal data type, then we only have 2 linguistic terms male and female. In this case we will have crisp membership values, a boy will have $\{(\text{male}, 1), (\text{female}, 0)\}$.
- If the data type of input attribute is continuous numerical, then the number of linguistic term equal to number of target class, and the membership function is trapezoid. For example, with iris data set base on the petal length, petal width, sepal length and sepal width we will classify iris flower into setosa, versicolor, virginica; which mean we have 3 target classes so we have 3 linguistic terms for all the input attribute which is small, average and large. We must find the value range of the input attribute value to define the trapezoid membership.

With application on iris data set, we used the trapezoidal membership function. The iris data set has 4 attributes petal length, petal width, sepal length and sepal width we will classify iris flower into setosa, versicolor, virginica.

Table 3.1: Linguistic term for iris dataset

Attributes	Linguistic term		
Sepal Length	Small	Average	Large
Sepal Width	Small	Average	Large
Petal Length	Small	Average	Large
Petal Width	Small	Average	Large

ii. Calculate the subsethood value

After define the fuzzy member class, Follow (2.8) we will calculate the subsethood value of each linguistic term over the classification class. Following is the application on iris data set

Table 3.2 Subsethood values for iris dataset

		setosa	versicolor	virginica
Sepal-length	Small	S(set,S_SL)	S(ver,S_SL)	S(vir,S_SL)
	Average	S(set,A_SL)	S(ver,A_SL)	S(vir,A_SL)
	Large	S(set,L_SL)	S(ver,L_SL)	S(vir,L_SL)
sepal-width	Small	S(set,S_SW)	S(ver,S_SW)	S(vir,S_SW)
	Average	S(set,A_SW)	S(ver,A_SW)	S(vir,A_SW)
	Large	S(set,L_SW)	S(ver,L_SW)	S(vir,L_SW)
petal-length	Small	S(set,S_PL)	S(ver,S_PL)	S(vir,S_PL)
	Average	S(set,A_PL)	S(ver,A_PL)	S(vir,A_PL)
	Large	S(set,L_PL)	S(ver,L_PL)	S(vir,L_PL)
petal-width	Small	S(set,S_PW)	S(ver,S_PW)	S(vir,S_PW)
	Average	S(set,A_PW)	S(ver,A_PW)	S(vir,A_PW)
	Large	S(set,L_PW)	S(ver,L_PW)	S(vir,L_PW)

Where S(set, S_SL) is the subsethood of Small_SepalLength with regard to setosa

iii. Calculate the weighted subsethood value

Apply (2.9) we can calculate the weighted subset hood value

Table 3.3: Weighted subsethood values for iris dataset

		setosa	versicolor	virginica
Sepal-length	Small	W(set,S_SL)	W(ver,S_SL)	W(vir,S_SL)
	Average	W(set,A_SL)	W(ver,A_SL)	W(vir,A_SL)
	Large	W(set,L_SL)	W(ver,L_SL)	W(vir,L_SL)
sepal-width	Small	W(set,S_SW)	W(ver,S_SW)	W(vir,S_SW)
	Average	W(set,A_SW)	W(ver,A_SW)	W(vir,A_SW)
	Large	W(set,L_SW)	W(ver,L_SW)	W(vir,L_SW)
petal-length	Small	W(set,S_PL)	W(ver,S_PL)	W(vir,S_PL)
	Average	W(set,A_PL)	W(ver,A_PL)	W(vir,A_PL)
	Large	W(set,L_PL)	W(ver,L_PL)	W(vir,L_PL)
petal-width	Small	W(set,S_PW)	W(ver,S_PW)	W(vir,S_PW)
	Average	W(set,A_PW)	W(ver,A_PW)	W(vir,A_PW)
	Large	W(set,L_PW)	W(ver,L_PW)	W(vir,L_PW)

Where W(set, S_SL) is the weighted subsethood of Small_SepalLength with regard to setosa

iv. Weighted fuzzy rule generation

After get the weighted subsethood values we can generate the set of weighted fuzzy production rule

- **If** Sepal length is [W(set,S_SL) x Small OR W(set,A_SL) x Average OR W(set,L_SL) x Large] **AND** Sepal width is [W(set,S_SW) x Small OR W(set,A_SW) x Average OR W(set,L_SW) x Large] **AND** Petal Length is [W(set,S_PL) x Small OR W(set,A_PL) x Average OR W(set,L_PL) x Large] **AND** Petal width is [W(set,S_PW) x Small OR W(set,A_PW) x Average OR W(set,L_PW) x Large] **Then** Class is **SETOSA**

- **If** Sepal length is [W(ver,S_SL) x Small OR W(ver,A_SL) x Average OR W(ver,L_SL) x Large] **AND** Sepal width is [W(ver,S_SW) x Small OR W(ver,A_SW) x Average OR W(ver,L_SW) x Large] **AND** Petal Length is [W(ver,S_PL) x Small OR W(ver,A_PL) x Average OR W(ver,L_PL) x Large] **AND** Petal width is [W(ver,S_PW) x Small OR W(ver,A_PW) x Average OR W(ver,L_PW) x Large] **Then** Class is **VERSICOLOR**

- **If** Sepal length is [W(vir,S_SL) x Small OR W(vir,A_SL) x Average OR W(vir,L_SL) x Large] **AND** Sepal width is [W(vir,S_SW) x Small OR W(vir,A_SW) x Average OR W(vir,L_SW) x Large] **AND** Petal Length is [W(vir,S_PL) x Small OR W(vir,A_PL) x Average OR W(vir,L_PL) x Large] **AND** Petal width is [W(vir,S_PW) x Small OR W(vir,A_PW) x Average OR W(vir,L_PW) x Large] **Then** Class is **VIRGINICA**

3.1.2.2 Fuzzy Neural network subsystem

The task of fuzzy neural network subsystem is training the weighted fuzzy production rules by modifying the weight so that the rule can classify data with more accuracy. Through the literature review we know that the fuzzy neural network generally has the input membership layer, fuzzy rule layer and output layer. The activation function for the input membership layer is the trapezoid membership function; the activation for the fuzzy rule layer is the fuzzy rule, and the activate function for the output layer is the logic operation function.

We can see that our generated weighted production fuzzy rules are the combination of "And" and "Or" operation, which cannot be used to activate the neuron. Thus we need to separate the rule into 2 sub rules; 1 sub rule perform the "OR" operation and 1 sub rule performs the "AND" operation. The initial rule can be rewrite as following

- **If** W'(set,SL) x {Sepal length is [W'(set,S_SL) x Small OR W'(set,A_SL) x Average OR W'(set,L_SL) x Large]} **AND** W'(set,SW) x {Sepal width is [W'(set,S_SW) x Small OR W'(set,A_SW) x Average OR W'(set,L_SW) x Large]} **AND** W'(set,Pl) x {Petal Length is [W'(set,S_PL) x Small OR W'(set,A_PL) x Average OR W'(set,L_PL) x Large]} **AND** W'(set,PW) x {Petal width is [W'(set,S_PW) x Small OR W'(set,A_PW) x Average OR W'(set,L_PW) x Large]} **Then** Class is **SETOSA**

Where W'(set,SL) x W'(set,S_SL) = W(set,S_SL); initially for neural network weight we can have W'(set,S_SL) = W(set,S_SL) and W'(set,SL) =1, similarity for the other weights

The rule is a logic And operation of sets of OR logic operation. After rewire the rule we can have a set of sub rule which has Or operation and "And" operation rule

- **Sub rule 1**: **if** Sepal length is [W'(set,S_SL) x Small OR W'(set,A_SL) x Average OR W'(set,L_SL) x Large] **Then** Class is **SETOSA valued at A1**

- **Sub rule 2 if** Sepal width is **[**W'(set,S_SW) x Small OR W'(set,A_SW) x Average OR W'(set,L_SW) x Large**] Then** Class is **SETOSA valued at A2**
- **Sub rule 3 if** Petal Length is [W'(set,S_PL) x Small OR W'(set,A_PL) x Average OR W'(set,L_PL) x Large**] Then** Class is **SETOSA valued at A3**
- **Sub rule 4 if** Petal width is [W'(set,S_PW) x Small OR W'(set,A_PW) x Average OR W'(set,L_PW) x Large**] Then** Class is **SETOSA value at A4**

Then we can have: **If W'(set,SL) x A1AND W'(set,SW) x A2 AND W'(set,Pl) x A3 AND W'(set,PW) x A4 then Class is Setosa**

By simplifying the rule as above, this will help us to modify the fuzzy rule layer in fuzzy neural network into 2 sub layer AND (Min) layer and OR (Max) Layer.

i. Fuzzy neural network design:

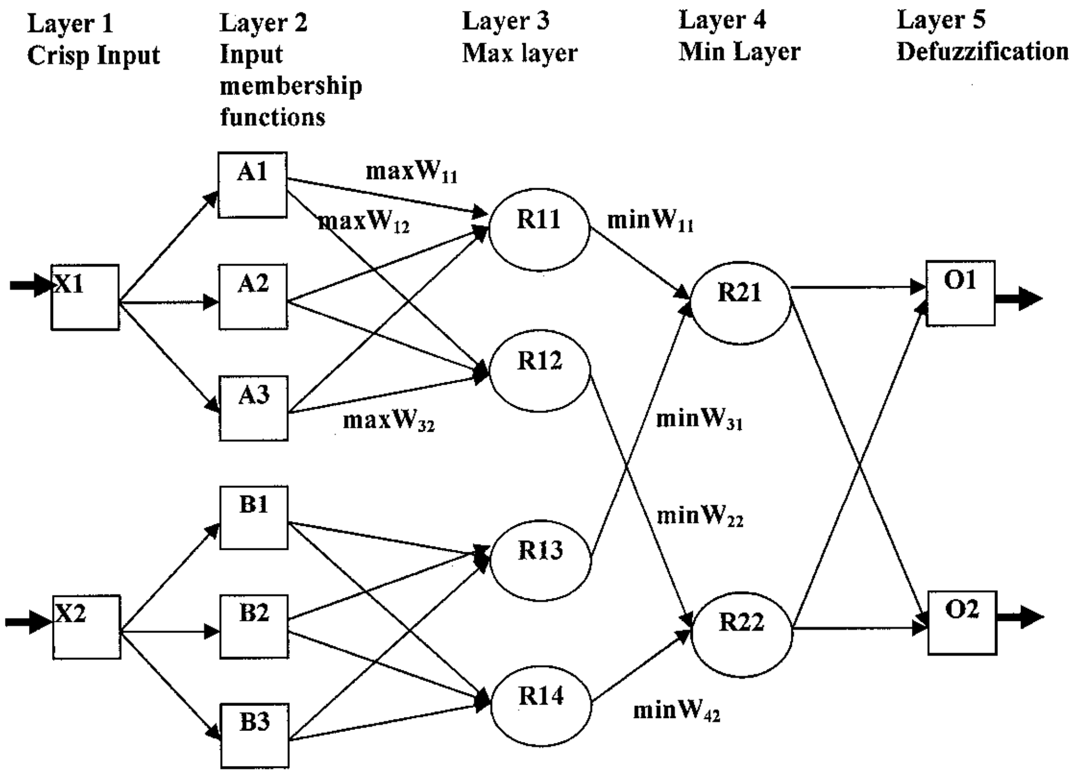


Figure 3.2 Fuzzy neural network systems

Layer 1 Input layer:

Each neuron in this layer transmits external crisp signals directly to the next layer

$Y^{(1)}_I = X^{(1)}_I$

Where $X^{(1)}_I$ is the input, $Y^{(1)}_I$ is the output of neuron I in layer 1

Layer 2 input membership layer:

Neurons in this layer represent fuzzy sets used in the antecedents of fuzzy rules. A fuzzification neuron receives a crisp input and determines the degree to which this input belongs to the neuron's fuzzy set. The activation function is the membership function.

The input is the output of crisp input layer $X^{(2)}_j = Y^{(1)}_i$

The output of neuron in this layer is the membership value of crisp input value

$Y^{(2)}_j = \mu(X^{(2)}_j)$

Layer 3 and Layer 4 are the fuzzy rule layers:

These layer perform the "and" and "or" operation in the weighted fuzzy rule. The activation for the "and" layer is the min operation, and the activation function for the "or" layer is the max operation

$Y_j^{(3)} = \vee_{i=0}^{\max C} (\max W_{i,j} * Y_i^{(2)}),$

where $\max C$ *is number of neuron in layer (3)*

$\max W_{i,j}$ is the weight of *connection* from neuron i inlayer (2) with *neuron j in layer (3)*

$Y_j^{(4)} = \wedge_{i=0}^{\min C} (\min W_{i,j} * Y_i^{(3)}),$

where minC is number of neuron in layer (4)

$\min W_{i,j}$ is the weight of connection from neuron i in layer (3) to neuron j in layer (4)

Layer 5 defuzzification

Classify the output. For the classification purpose, we make a minor modification for the computed output of network

$$Y_i^{(5)} = \begin{cases} 1, \text{if } y_i^{(4)} = \text{Max}_{i=1..n} y_i^{(4)} \\ 0, \text{if } y_i^{(4)} \neq \text{Max}_{i=1..n} y_i^{(4)} \end{cases}$$

For 1 iteration the error will be

$$E(p) = \frac{1}{2}\sum_{k=1}^{OC}(y_k^{(4)} - y_k)^2$$

Where OC is number of output neurons, y_k is the desired output for neuron k and $y^{(5)}_k$ is the actual output of neuron k. we can see that the error E(p) is the function with respect to minW, maxW. The main objective is to adjust these weights such that error function reaches minimum or is less than a given threshold

ii. Learning in fuzzy neural network

For learning phase we apply the back propagation equations. According to the principle of gradient descent, the back propagation equations can be written as

$$\min W_{i,j}(p+1) = \min W_{i,j}(p) - \alpha * \frac{\partial E(p)}{\partial \min W_{i,j}}$$

$$\max W_{i,j}(p+1) = \max W_{i,j}(p) - \alpha * \frac{\partial E(p)}{\partial \max W_{i,j}}$$

Where α is the learning rate

We can see that our network is not fully connected, the connection is predetermined

$$\frac{\partial E(p)}{\partial \min W_{i,j}} = \frac{\partial E(p)}{\partial Y_j^{(4)}} * \frac{\partial Y_j^{(4)}}{\partial \min W_{i,j} * Y_i^{(3)}} * \frac{\partial \min W_{i,j} * Y_i^{(3)}}{\partial \min W_{i,j}}$$

$$\frac{\partial E(p)}{\partial Y_j^{(4)}} = \frac{\partial \frac{1}{2}\sum_{k=1}^{L3}(Y_k^{(4)} - Y_k)^2}{\partial Y_j^{(4)}} = (Y_j^{(4)} - Y_j) = O_j$$

$$\frac{\partial Y_j^4}{\partial \min W_{i,j} * Y_i^{(3)}} = \frac{\partial \wedge_{k=0}^{L2}(\min W_{k,j} * Y_k^{(3)})}{\partial \min W_{i,j} * Y_i^{(3)}} = \begin{cases} 1, \textit{if } \min W_{i,j} * Y_i^{(3)} \textit{is} \wedge_{k=0}^{\min C}(\min W_{k,j} * Y_k^{(3)}) \\ \wedge_{ki=0,k\neq i}^{\min C}(\min W_{k,j} * Y_k^{(3)}), \\ \textit{if } \min W_{i,j} * Y_i^{(3)} \textit{isnot} \wedge_{k=0}^{\min C}(\min W_{k,j} * Y_k^{(3)}) \end{cases}$$

$$\frac{\partial \min W_{i,j} * Y_i^{(3)}}{\partial \min W_{i,j}} = Y_i^{(3)}$$

$$\frac{\partial E(p)}{\partial \min W_{i,j}} = \begin{cases} (Y_j^{(4)} - Y_j) * Y_i^{(3)}, \textit{IF } \min W_{i,j} * Y_i^{(3)} \textit{is} \wedge_{k=0}^{\min C}(\min W_{k,j} * Y_k^{(3)}) \\ (Y_j^{(4)} - Y_j) * Y_i^{(3)} * \wedge_{ki=0,k\neq i}^{\min C}(\min W_{k,j} * Y_k^{(3)}), \textit{IF } \min W_{i,j} * Y_i^{(3)} \textit{is not} \wedge_{k=0}^{\min C}(\min W_{k,j} * Y_k^{(3)}) \end{cases}$$

For $\max W_{i,j}$ which is the connection from neuron i in layer 2 to neuron j in layer 3. Suppose that neuron j in layer 3 is connected with neuron u in layer 4. Therefore we can calculate the error gradient as following

$$\frac{\partial E(p)}{\partial \max W_{i,j}} = \frac{\partial E(p)}{\partial Y_u^{(4)}} * \frac{\partial Y_u^{(4)}}{\partial Y_j^{(3)}} * \frac{\partial Y_j^{(3)}}{\partial \max W_{i,j} * Y_i^{(2)}} * \frac{\partial \max W_{i,j} * Y_i^{(2)}}{\partial \max W_{i,j}}$$

$$\frac{\partial E(p)}{\partial y_u^{(4)}} = \frac{\partial \frac{1}{2}\sum_{k=1}^{\max C}(Y_k^{(4)} - Y_k)^2}{\partial Y_u^{(4)}} = (Y_u^{(4)} - Y_u) = O_u$$

$$\frac{\partial Y_u^{(40}}{\partial Y_j^{(3)}} = \frac{\partial Y_u^{(4)}}{\partial \min W_{j,u} * Y_j^{(3)}} * \frac{\partial \min W_{j,u} * Y_j^{(3)}}{\partial Y_j^{(3)}}$$

$$\frac{\partial Y_u^{(4)}}{\partial \min W_{j,u} * Y_j^{(30}} = \frac{\partial \wedge_{k=0}^{\min C} (\min W_{k,u} * Y_k^{(3)})}{\partial \min W_{j,u} * Y_j^{(3)}} = \begin{cases} 1, if \min W_{j,u} * Y_j^{(3)} is \wedge_{k=0}^{\min C} (\min W_{k,u} * Y_k^{(3)}) \\ \wedge_{ki=0,k\neq j}^{\min C} (\min W_{k,u} * Y_k^{(3)}), \\ if \min W_{j,u} * Y_j^{(3)}\ is\ not \wedge_{k=0}^{\min C} (\min W_{k,u} * Y_k^{(3)}) \end{cases}$$

$$\frac{\partial \min W_{j,u} * Y_j^{(3)}}{\partial Y_j^{(3)}} = \min W_{j,u}$$

$$\frac{\partial Y_j^{(3)}}{\partial \max W_{i,j} * Y_i^{(2)}} = \frac{\partial \vee_{k=0}^{\max C} (\max W_{k,j} * Y_{ki}^{(2)})}{\partial \max W_{i,j} * Y_i^2} = \begin{cases} 1, if \max W_{i,j} * Y_i^{(2)} is \vee_{k=0}^{\max C} (\max W_{k,j} * Y_{ki}^{(} \\ \max W_{i,j} * Y_i^{(2)}, if \\ \max W_{i,j} * Y_i^{(2)} is\ not \vee_{k=0}^{\max C} (\max W_{k,j} * Y_{ki}^{(} \end{cases}$$

$$\frac{\partial \max W_{i,j} * Y_i^{(2)}}{\partial \max W_{i,j}} = Y_i^2$$

$$\frac{\partial E(p)}{\partial \max W_{i,j}} = \begin{cases} (Y_u^{(4)} - Y_u) * \min W_{j,u} * Y_i^2, if\ C1 \\ (Y_u^{(4)} - Y_u) * \wedge_{ki=0,k\neq j}^{\min C} (\min W_{k,u} * Y_k^{(3)}) * \min W_{j,u} * Y_i^2, if\ C2 \\ (Y_u^{(4)} - Y_u) * \min W_{j,u} * \max W_{i,j} * Y_i^{(2)} * Y_i^2, if\ C3 \\ (Y_u^{(4)} - Y_u) * \wedge_{ki=0,k\neq j}^{\min C} (\min W_{k,u} * Y_k^{(3)}) * \min W_{j,u} * \max W_{i,j} * Y_i^{(2)} * Y_i^2, if\ C4 \end{cases}$$

$C1$ is $\min W_{j,u} * Y_j^{(3)} is \wedge_{k=0}^{\min C} (\min W_{k,u} * Y_k^{(3)})$ and $\max W_{i,j} * Y_i^{(2)} is \vee_{k=0}^{\max C} (\max W_{k,j} * Y_{ki}^{(2)})$

$C2$ is $\min W_{j,u} * Y_j^{(3)} is$ not $\wedge_{k=0}^{\min C} (\min W_{k,u} * Y_k^{(3)})$ and $\max W_{i,j} * Y_i^{(2)} is \vee_{k=0}^{\max C} (\max W_{k,j} * Y_{ki}^{(2)})$
$C3$ is $\min W_{j,u} * Y_j^{(3)} is \wedge_{k=0}^{\min C} (\min W_{k,u} * Y_k^{(3)})$ and $\max W_{i,j} * Y_i^{(2)} is$ not $\vee_{k=0}^{\max C} (\max W_{k,j} * Y_{ki}^{(2)})$
$C4$ is $\min W_{j,u} * Y_j^{(3)} is$ not $\wedge_{k=0}^{\min C} (\min W_{k,u} * Y_k^{(3)})$ and $\max W_{i,j} * Y_i^{(2)} is$ not $\vee_{k=0}^{\max C} (\max W_{k,j} * Y_{ki}^{(2)})$

After training we will have a new set of weight, and the trained rule will be

If minW(set,SL) x {Sepal length is [maxW(set,S_SL) x Small OR maxW(set,A_SL) x Average OR maxW(set,L_SL) x Large]}
AND minW(set,SW) x {Sepal width is [maxW(set,S_SW) x Small OR maxW(set,A_SW) x Average OR maxW(set,L_SW) x Large]}
AND minW(set,Pl) x {Petal Length is [maxW(set,S_PL) x Small OR maxW(set,A_PL) x Average OR maxW(set,L_PL) x Large]}
AND minW(set,PW) x {Petal width is [maxW(set,S_PW) x Small OR maxW(set,A_PW) x Average OR maxW(set,L_PW) x Large]}
Then Class is **SETOSA**

Or we can have the simplified rule

If Sepal length is [newW(set,S_SL) x Small OR newW(set,A_SL) x Average OR newW(set,L_SL) x Large]
AND Sepal width is [newW(set,S_SW) x Small OR newW(set,A_SW) x Average OR newW(set,L_SW) x Large]
AND Petal Length is [newW(set,S_PL) x Small OR newW(set,A_PL) x Average OR newW(set,L_PL) x Large]
AND Petal width is [newW(set,S_PW) x Small OR newW(set,A_PW) x Average OR newW(set,L_PW) x Large]
Then Class is **SETOSA**
Where newW(set,S_SL) = maxW(set,S_SL) x minW(set,SL)

3.1.3 Implementation and Unit Testing

Based on the architectural design framework from the previous phase, the implementation task is performed. The whole application is partitioned into a set of program units or modules. Depending on the resources available, modules will be developed by a team or a person. Unit testing involves verifying that each unit/ module meets its specification. Typically, 30%-40% of project's time budget is allocated to this phase.

The implementation of this project is divided into modules consisting of the typical processes or phases of data mining, namely Data Selection, Data Preprocessing, Fuzzy Rule Induction, Fuzzy Rule Training, and Validation. Each of these processes is considered as a module of the application and is developed in sequence due to the linear nature in this tool. After modules are coded, they will be tested to ensure their conformance to the requirements specification.

3.1.4 Integration and System Testing

The individual program units or programs are integrated and tested as a complete system to ensure that the software requirements have been met. After testing, the software system is delivered to the customer (Sommerville, 2001). This is the phase whereby different modules developed and tested in this project are integrated into a complete application. An overall testing on the application is performed to ensure that the integrated modules work according to the specification. At this stage, it may lead to detection of incompatibility between developed modules and modifications are necessary to ensure that the end product is error-free.

3.2 Development Tools

The system is designed in VB .Net programming language, In order to run the system, dotnet framework is required.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Data Set Description

4.1.1 Iris data set

The iris plant dataset contains four linguistic variables: sepal length, sepal width, petal length, petal width. It consists of 150 object instances and 3 classes: iris-setosa, iris-veriscolor, iris-virginica. The data set can be obtained form UCI machine leaning website [23]

4.1.2 Credit card approval dataset

The credit card approval process is always performed to check if the applicant is qualified for a credit card. A bank will need to ask the applicants certain questions, which will assist the approval department make their decision about whether to grant the applicant a credit line or not. A bank will need to take into consideration many personal details, especially credit history, of the applicant and no simple rule of thumb is in place to assess the risk of granting credit card to any one person. This is where data mining plays an important role to dig into the historical data of approved and rejected cases to find out the patterns of good and bad credit card applicants. The knowledge gained from this data mining process, together with banking experts' judgments would decide whether to approve the credit card application and reduce the risks for the bank.

The credit card screening dataset, originated from J. R. Quinlan, was obtained from the following URL: http://www.csee.usf.edu/~mlast/credit.dat.The dataset contains records from bank credit card applications, including the credit outcomes (accept / reject). There are altogether 14 input attributes and 1 class attribute. The total number of instances is 690. The number of class labels is 2 (accept / reject). This dataset is interesting because there is a good mix of attributes -- continuous, nominal with small numbers of values, and nominal with larger numbers of values. There is no missing value in this dataset. The short description of the attributes is available in Table 4.1:

Attribute	**Domain**	**Type**	**Use in Model**
Sex	0,1	Nominal	candidate input
Age	13.75 - 80.25	Continuous	candidate input
Mean time at addresses	0 - 28	Continuous	candidate input
Home status	1,2,3	Nominal	candidate input
Current occupation	1 - 14	Nominal	candidate input
Current job status	1 - 9	Nominal	candidate input
Mean time with employers	0 - 28.5	Continuous	candidate input
Other investments	0,1	Nominal	candidate input
Bank account	0,1	Nominal	candidate input
Time with bank	0 - 67	Continuous	candidate input
Liability reference	0,1	Nominal	candidate input
Account reference	1,2,3	Nominal	candidate input
Monthly housing expense	0 - 2000	Continuous	candidate input
Savings account balance	1 - 100001	Continuous	candidate input
Class (Accept / Reject)	0,1	Nominal	target

Table 4.1 credit card approval data set

4.2 System Operation

The system operate to demonstrate the proposed algorithm which involves several steps

- Load dataset: Load dataset into the system

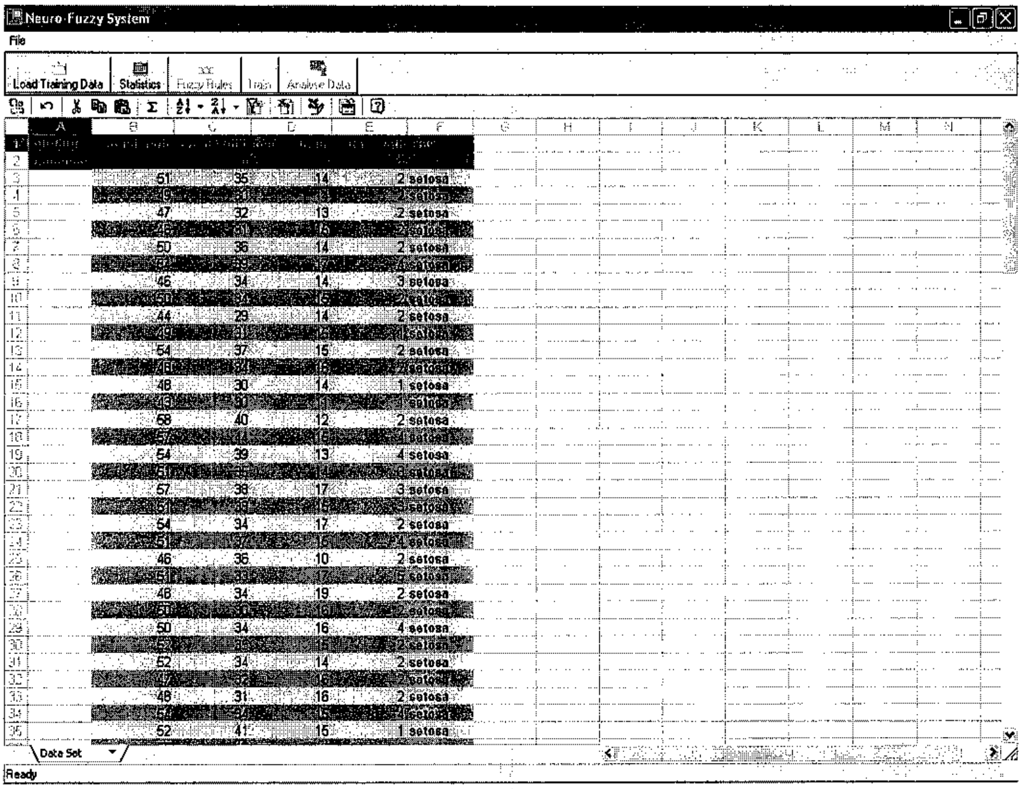


Figure 4.1 System user interface – Load data

- Pre-processing: calculate the subsethood and weighted subsethood value

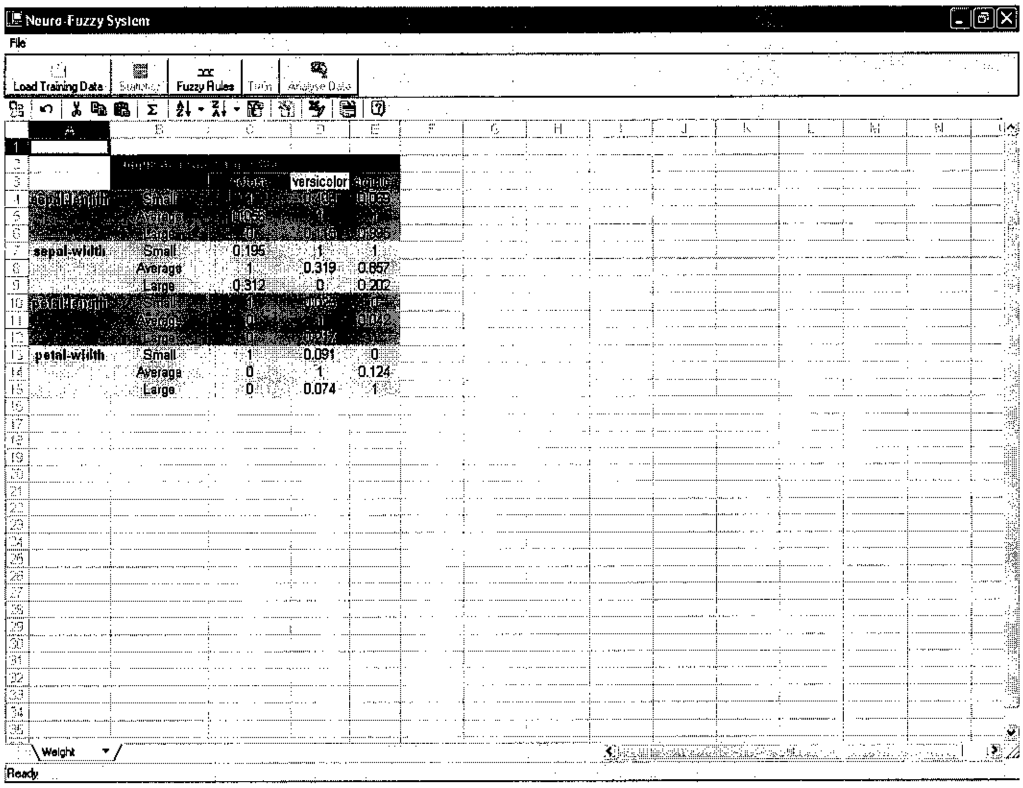


Figure 4.2 System user interface – Pre-processing

- Generating rule: generate the weighted fuzzy production rules

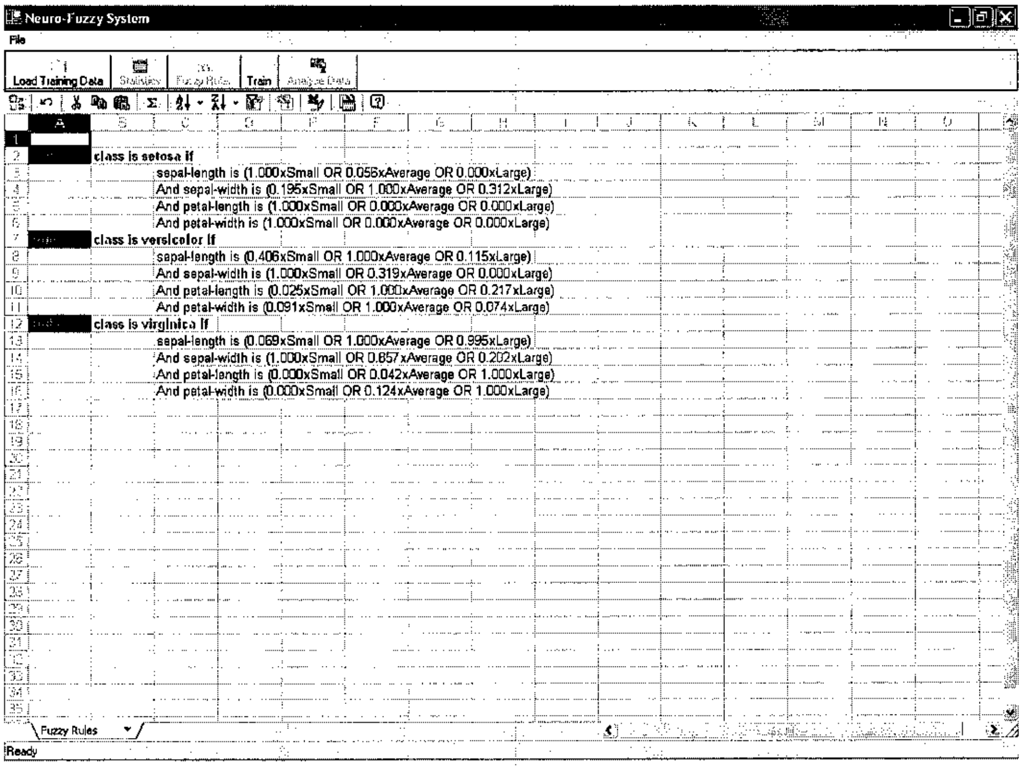


Figure 4.3 System user interface – Rule generating

- Training: Apply min-max neural network to train the weighted

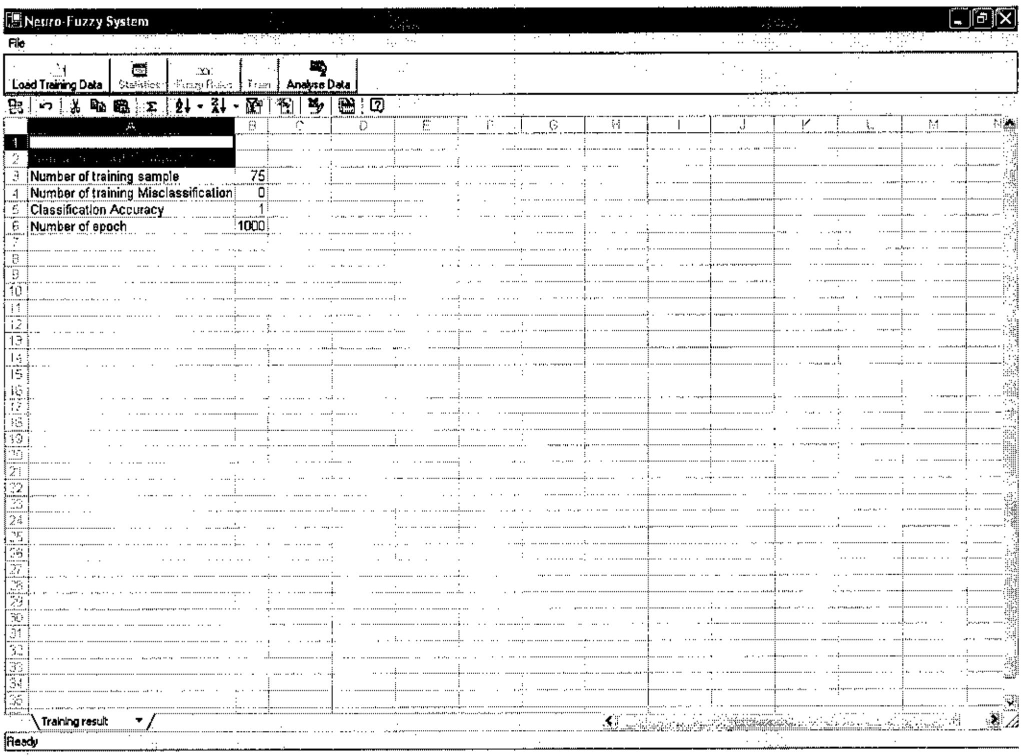


Figure 4.4 System user interface – Training

- Analyze: Base on the trained rules to analyze data

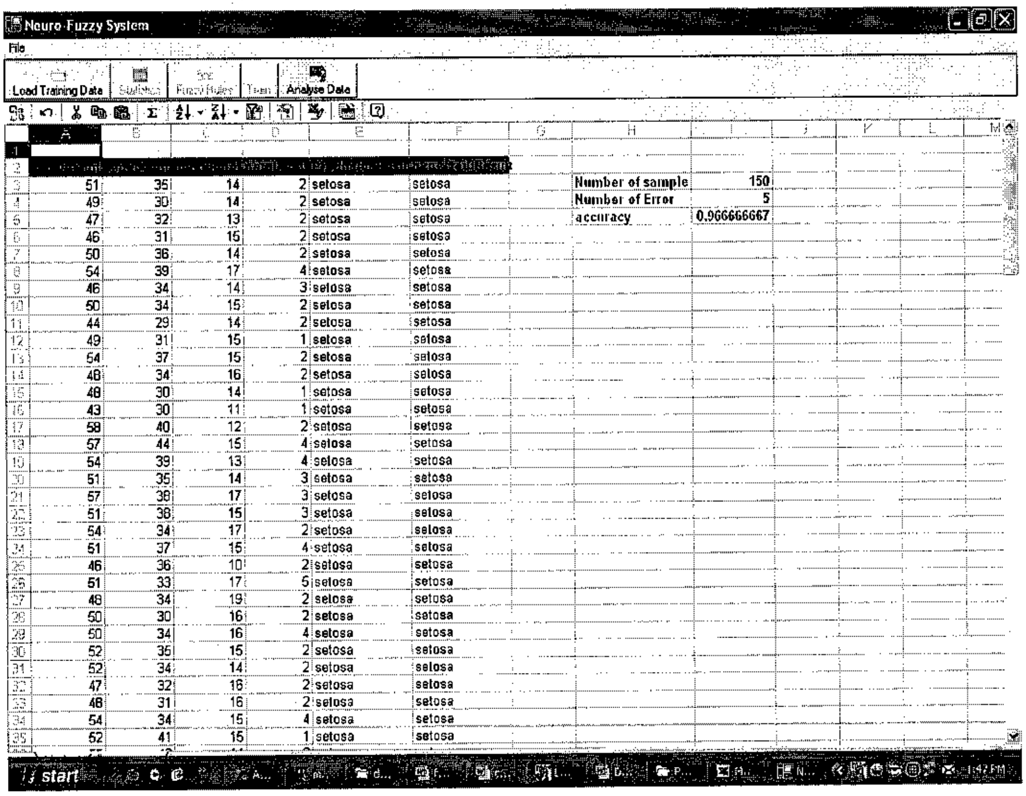


Figure 4.5 System user interface – Result

4.3 Result and Discussion

4.3.1 Result on iris data set

With iris data set, the entire dataset was labeled from 1 to 150, is divided equally into two sub-datasets: IP1 and IP2; IP1 consist of the odd numbered objects and IP2 consists of the even numbered object. Then we feed the sub-dataset to train the fuzzy neural system and analyze data. For evaluation, we use the IP1 to train the system, and then analyze the IP2 and the whole dataset; and we also use the IP2 to training the system then analyze the IP1 and the whole dataset.

With the above experience we have achieved the results as following

Table 4.2: Iris data set result

Training dataset	Training Accuracy	Testing dataset	SBA	WSBA	Fuzzy neural
IP1	100%	IP2	80% (15)	93.33% (5)	94.67% (4)
		Whole	78.69% (32)	94.67% (8)	96.67% (5)
IP2	97.30%	IP1	78.67% (16)	93.33% (5)	93.33% (5)
		Whole	78% (33)	93.33% (10)	96.77% (5)

SBA: subsethood based algorithm

WSBA: Weighted subsethood based algorithm

4.3.2 Result on credit card approval dataset

For evaluating the system, we apply the credit card approval for testing, we divided the dataset into 2 sub-datasets, one is used for training and the other is used for evaluating the system. We randomly select 358 samples from the credit card approval data set for training set (Cr1) and we use the rest 332 sample (Cr2) for testing.

The achieved training accuracy is 80% and the evaluation accuracy is 81%

Table 4.3: Credit card data set result

Training dataset	Training Accuracy	Testing dataset	SBA	WSBA	Fuzzy neural
Cr1	80%	Cr2	70%	76%	81%

SBA: subsethood based algorithm

WSBA: Weighted subsethood based algorithm

The hybrid system, fuzzy neural network has improved the accuracy of the weighted fuzzy production rules. For the iris dataset, the improvement is about 1-2 % increasing in accuracy and for the credit card approval dataset we have 5% accuracy improving. We can realize that with the hybrid system we can have the high accuracy of the neural network classifier and we also can generate the rules which inherited from fuzzy logic classification.

In conclusion, the proposed algorithm, the fuzzy neural network helps to increase the accuracy of the fuzzy logic classifier. Besides that compare to neural network classifier, the fuzzy neural approach can help to reduce processing time, because the initial weight is taken from the weighted fuzzy rules which has accuracy quite high; thus we don't have to randomly generate the weight and start training from without knowledge. And the fuzzy neural approach can generate rules can be interpreted which is not applicable in neural network. The proposed algorithm is still in the developing stage, it has improved the fuzzy classifier and as well the neural network training; it provides a promising area in data classification.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Data mining has emerged to be a very important research area that helps organizations make good use of the tremendous amount of data they have. In the past, it was almost an impossible task to dig for information from the huge amount of data due to technology, manpower and time constraints. Data mining has unlocked those limitations. With the combination of many other research disciplines, data mining turns raw data into useful information rather than just raw, meaningless data.

This project has proposed a new algorithm for data classification task. The algorithm has achieves some of the targeted objectives. Fuzzy logic and neural networks are complementary tools in building intelligent systems. While neural networks are low-level computational structures that perform well when dealing with raw data, fuzzy logic deals with reasoning on a higher level, using linguistic information acquired from domain experts.

Fuzzy systems lack the ability to learn and cannot adjust themselves to a new environment. On the other hand, although neural networks can learn, they are opaque to the user. By developing a hybrid system, the merger of a neural network with fuzzy system into one integrated system, the proposed algorithm has the advantages of neural network which is high accuracy, and also can overcome the weakness of neural network which is can not generate rules by applying fuzzy logic inference. Therefore the proposed algorithm offers a promising approach to building intelligent systems. Integrated fuzzy neural systems can combine the parallel computational and learning abilities of neural networks with the human-like knowledge representation and explanation abilities of fuzzy system. As a result neural networks become more transparent, while fuzzy systems become capable of learning.

In conclusion, the proposed algorithm has achieved the target objectives; it can gain high accuracy for data classification task, the algorithms also can generate rules which can help to interpret the output results. The algorithms is also robust as it applies the weighted from fuzzy productions rules as initial weights, which already good in classifying, this reduce the training processing. But the proposed algorithm still require high computation, the processing time will be long if the dataset is huge. However the proposed algorithm offers a promising approach to building intelligent systems.

5.2 Recommendations

Since this project serves as a demonstration of the basic features of a typical data mining process, there exist many other extensions to this work to make it become a more and more powerful tool.

Firstly, this tool depends on the hardware, particularly the memory and the CPU, to work successfully. Therefore, when it comes to very large dataset, the capacity of the memory and the computation power of the CPU become critical. This is because the whole dataset currently has to reside in the memory. Large dataset would require bigger memory capacity and it also requires the CPU to perform more computation during its induction process. Therefore, there should be a new design of the application such that it can interface with a database server, for example, Oracle DBMS, or Microsoft SQL Server, etc. This design would offload the whole dataset to the database server, and also the necessary computations can also be performed by the database server.

Secondly, the proposed algorithm is still in developing stages. At this stage it generates rules from all available linguistic terms. There are many areas for improvement such as:

- Define better membership functions; this involves more on probability and statistics research
- Prune the weighted fuzzy rules before training, because it's not every linguistic terms is responsible for the target result, and some linguistic term is so less

important that we can ignore, this can help to improve processing time and reduce number of rules but get better accuracy

- Learning algorithm: the min-max neural network use the min and max operation which is not differentiated, the algorithms has applied the smooth derivative for back propagation learning, but this learning algorithm is still request high computation which is need to improve further

Lastly, more preprocessing features, such as scaling data, converting data in date format to other useful format, etc. can be implemented to allow for higher quality data. Robust preprocessing mechanism would improve significantly the performance of the classifier

REFERENCES

[1] Margaret H. Dunham, 2003, "Data Mining Introductory and Advanced Topics", New Jersey, Prentice Hall.

[2] Mehmeh Katardzic, 2003, "Data Mining: Concepts, Models, and Algorithm", John Wiley & Sons.

[3] Olivia Parr Rud, 2001, "Data Mining Cookbook: Modeling Data for Marketing, Risk, and Customer Relationship Management", John Wiley & Sons.

[4] David Hand, Heikki Mannila, Padhraic Smuth, 2002, "Principles of Data Mining", The MIT Press.

[5] Jiawei Han, Micheline Kamber, 2000, "Data Mining: Concepts and Techniques", Morgan Kaufmann.

[6] Freitas, A.A. "Data Mining and Knowledge Discovery With Evolutional Algorithms", Neural Computing series, 2002.

[7] John W., 2003, "Data Mining: Oppoetunities and Challenges", Idea Group Publishing.

[8] Michalski R. et al, 1998, "Machine Learning and Data Mining: Methods and Application" John Wiley & Son.

[9] Walpole et al: Probability & Statistics for Engineers & Scientists, 7th Edition, Prentice Hall, 2002.

[10] Karuna Pande Joshi, 1997, "Analysis of Data Mining Algorithms", <http://userpages.umbc.edu/~kjoshi1/data-mine/proj_rpt.htm> accessed on January 2005

[11] Minaei-Bidgoli, B.; Kashy, D.A.; Kortmeyer, G.; Punch, W.F.; "Predicting student performance: an application of data mining methods with an educational Web-based system" Frontiers in Education, 2003. FIE 2003. 33rd Annual , Volume: 1 , 5-8 Nov. 2003 Pages:T2A - 13-18 Vol.1

[12] Myller, N.; Suhonen, J.; Sutinen, E.; "Using data mining for improving web-based course design" Computers in Education, 2002. Proceedings. International Conference on , 3-6 Dec. 2002 Pages:959 - 963 vol.2

[13] Ma, Y., Lee, S.M., Liu, B., Yu, P.S., Wong, C.K., "Targeting the Right Students Using Data Mining", Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, Massachusetts, United States, 2000, 457-464.

[14] Castro J.L. and Zurita J.M. (1997) "An inductive Learning Algorithm in fuzzy Systems", Fuzzy Sets and Systems, vol. 89, pp 193-203

[15] D. S. Yeung and E. C. C. Tsang, "Weighted Fuzzy Production rules", Fuzzy Sets and Systems, vol. 88 pp229-313, 1997.

[16] E. C. C. Tsang and D. S. Yeung, "Refining Local Weights and Certainty Factors using a Neural Network", IEEE International. Conference on Systems, Man and cybernetics (San Diego, CA, October), pp 1512-16\517, 1998.

[17] N. K. Kasabov, "Learning Fuzzy Rules and approximate reasoning in fuzzy networks and hybrid systems", fuzzy Sets and Systems, vol. 82, pp135-149, 1996.

[18] Chen S.M., Lee S.H., and Lee C.H.(2001) "A new Method for generating Fuzzy Rules From numerical data for handling Classification Problems", Applies Artificial Intelligence, vol. 15, pp 645-664.

[19] Hong T.P. and Lee. C.Y. (1996) "Induction of Fuzzy Rules and Membership Functions from Training Examples", Fuzzy sets and Systems, vol. 84, pp33-47.

[20] Nauck D., Nauck U. and Kruse R. (1996) " generating Classification rules with the Neuro-fuzzy System NEFCLASS, Proceeding of Bienmal conference of the North American Fuzzy Information Processing society NAFIP's 96, Berkeley, CA>

[21] Nozaki K., Ishibuchi H. and Tanaka H. (1997) "A Simple but powerful Heuristics Method for generating fuzzy Rules from Numerical data", fuzzy Sets and Systems, vol. 86, pp251-270

[22] Shen Q and Chouchoulas A. (2000) "A fuzzy-Rough approach for generating Classification Rules," Pattern recognition, vol 35, pp341-354.

[23] UCI Machine Learning Databases. Available online [http://ftp.ics.uci.edu/pub/machine-learning-databases/]

[24] Zadeth L. A> (1988) Fuzzy Logic IEEE – Computer Science, vol. 21, pp83-93

[25] Yuan Y. and zhuang H> (1996) " A genetic Algorithm foe generating Fuzzy Classification rules" Fuzzy sets and Systems, vol. 84, pp1-19

[26] Khairul Anwar Rasmani, (2002), " A Data –Driven Fuzzy rule Based Approach for Student Academic Performance Evaluation"

[27] Eric C.C. Tsang, Daniel So Yeung and Xi-Zhao Wang,(2002) "Learning Weights of Fuzzy Production Rules by A Max-min neural network"

[28] A. Blanco, M. Delgado, I. Requena, "Identification of fuzzy relational equations by fuzzy neural networks", fuzzy sets and systems, vol. 71, pp215-226, 1995.

[29] Micheal Negnevitsky, " Artificial Intelligence, Aguid to Intelligencew systems", Addison Wesley 2002