

# Neutrosophic Logic Based Semantic Web Services Agent

Haibin Wang

*Department of Computer Science, Georgia State University, Atlanta, GA 30302,  
U.S.A.*

Yan-Qing Zhang\*

*Department of Computer Science, Georgia State University, Atlanta, GA 30302,  
U.S.A.*

Rajshekhar Sunderraman

*Department of Computer Science, Georgia State University, Atlanta, GA 30302,  
U.S.A.*

Florentin Smarandache

*Mathematical Department, University of New Mexico, Gallup, NM 87301, U.S.A.*

---

## Abstract

In this paper, we propose a framework called Semantic Web Services (SWS) agent for providing high QoS Semantic Web services. The SWS agent is based on the neutrosophic logic. The neutrosophic logic was recently proposed by Smarandache to model and reason with fuzzy, incomplete and inconsistent information. The SWS agent can solve two challenges facing practicability of current Web services technology. One is how to locate the services Registries having requested Web services efficiently; another is how to retrieve the requested services from these Registries with high QoS. We use neutrosophic neural networks with Genetic Algorithms (GA) to do the simulation. Simulation results show that the SWS agent is extensible and scalable to handle uncertain QoS metrics effectively.

## 1 Introduction

The concept of fuzzy sets was introduced by Zadeh in 1965 [1]. Since then fuzzy sets and fuzzy logic have been applied to many real applications to handle uncertainty. The traditional fuzzy set uses one real value  $\mu_A(x) \in [0, 1]$  to represent the grade of membership of fuzzy set  $A$  defined on universe  $X$ . The corresponding fuzzy logic associates each proposition  $p$  with a real value  $\mu(p) \in [0, 1]$  which represents the degree of truth. Sometimes  $\mu_A(x)$  itself is uncertain and hard to be defined by a crisp value. So the concept of interval valued fuzzy sets was proposed [2] to capture the uncertainty of grade of membership. The traditional fuzzy logic can be easily extended to the interval valued fuzzy logic. There are related works such as type-2 fuzzy sets and type-2 fuzzy logic [3–5]. The family of fuzzy sets and fuzzy logic can only handle “complete” information that is if a grade of truth-membership is  $\mu_A$  then a grade of falsity-membership is  $1 - \mu_A(x)$  by default. In some applications such as expert systems, decision making systems and information fusion systems, the information is both uncertain and incomplete. Traditional fuzzy sets and fuzzy logic cannot handle such situation. In 1986, Atanassov introduced the intuitionistic fuzzy set [6] which is a generalization of a fuzzy set. The intuitionistic fuzzy sets consider both truth-membership and falsity-membership. The corresponding intuitionistic fuzzy logic [7] associates each proposition  $p$  with two real values  $\mu(p)$ -truth degree and  $\nu(p)$ -falsity degree, respectively, where  $\mu(p), \nu(p) \in [0, 1], \mu(p) + \nu(p) \leq 1$ . So intuitionistic fuzzy sets and intuitionistic fuzzy logic can handle uncertain and incomplete information.

However, inconsistent information exists in a lot of real situations such as those mentioned above. It is obvious that the intuitionistic fuzzy logic cannot reason with inconsistency because it requires  $\mu(p) + \nu(p) \leq 1$ . In 1995, Smarandache introduced the concept of neutrosophic sets and neutrosophic logic which can model and reason with fuzzy, incomplete and inconsistent information at the same time. A special case of the neutrosophic sets and neutrosophic logic is studied in [8,9].

---

\* Corresponding author. Permanent address: Department of Computer Science, Georgia State University, Atlanta, GA 30302, U.S.A. Tel.: +1-404-651-0682; fax: +1-404-463-9912.

*Email addresses:* [hwang17@student.gsu.edu](mailto:hwang17@student.gsu.edu) (Haibin Wang), [yzhang@cs.gsu.edu](mailto:yzhang@cs.gsu.edu) (Yan-Qing Zhang), [raj@cs.gsu.edu](mailto:raj@cs.gsu.edu) (Rajshekhar Sunderraman), [smarand@unm.edu](mailto:smarand@unm.edu) (Florentin Smarandache).

In this paper, we propose a framework called Semantic Web services (SWS) agent based on neutrosophic logic to provide high QoS Semantic Web services based on specific domain ontology such as gnome. The SWS agent can solve two challenges existing for automatic discovery and invocation of Web services. One is how to locate the services Registries advertising requested Web services efficiently; another is how to retrieve the requested services from these Registries with the highest quality of service (QoS). The Semantic Web services technologies can be exploited to solve the first challenge. For the second challenge, we believe the QoS of Semantic Web services should cover both functional and non-functional properties. Here we must be aware that on the one hand, the degree of capability matching [10–13] and non-functional properties are all fuzzy, incomplete and even inconsistent; and on the other hand, different application domains have different requirements on non-functional properties. It is not flexible to use classical mathematical modelling methods to evaluate the whole QoS of Semantic Web services.

The paper is organized as follows. In section 2, we present the necessary background knowledge of neutrosophic logic and QoS model. Section 3 provides details of design of architecture of the SWS agent. Section 4 gives the design of neutrosophic neural network with GA and simulation result. In section 5, we present the related work. And finally, in section 6, we conclude this paper and give the future research direction.

## 2 Background

This section details the background material related to this research. We give a brief review of neutrosophic logic and QoS model.

### 2.1 *Neutrosophic Propositional Logic*

In this section, we introduce the elements of the neutrosophic propositional logic based on the definition of neutrosophic sets [14] by using the notation from the theory of classical propositional logic.

#### 2.1.1 *Syntax of Neutrosophic Propositional Logic*

Here we give the formalization of syntax of the neutrosophic propositional logic.

**Definition 1** *An alphabet of the neutrosophic propositional logic consists of three classes of symbols:*

- (1) A set of neutrosophic propositional variables, denoted by lower-case letters, sometimes indexed;
- (2) Five connectives  $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$  which are called conjunction, disjunction, negation, implication, and bimplication symbols respectively;
- (3) The parentheses ( and ).

□

The alphabet of the neutrosophic propositional logic has combinations obtained by assembling connectives and neutrosophic propositional variables in strings. The purpose of the construction rules is to have the specification of distinguished combinations, called formulas.

**Definition 2** *The set of formulas (well-formed formulas) of the neutrosophic propositional logic is defined as follows.*

- (1) Every interval neutrosophic propositional variable is a formula;
- (2) If  $p$  is a formula, then so is  $(\neg p)$ ;
- (3) If  $p$  and  $q$  are formulas, then so are
  - (a)  $(p \wedge q)$ ,
  - (b)  $(p \vee q)$ ,
  - (c)  $(p \rightarrow q)$ , and
  - (d)  $(p \leftrightarrow q)$ .
- (4) No sequence of symbols is a formula which is not required to be by 1, 2, and 3.

□

To avoid having formulas cluttered with parentheses, we adopt the following precedence hierarchy, with the highest precedence at the top:

$$\begin{array}{c} \neg, \\ \wedge, \vee, \\ \rightarrow, \leftrightarrow. \end{array}$$

Here is an example of the neutrosophic propositional logic formula:

$$\neg p_1 \wedge p_2 \vee (p_1 \rightarrow p_3) \rightarrow p_2 \wedge \neg p_3$$

**Definition 3** *The language of interval neutrosophic propositional calculus given by an alphabet consists of the set of all formulas constructed from the symbols of the alphabet.* □

### 2.1.2 Semantics of Neutrosophic Propositional Logic

The study of neutrosophic propositional logic comprises, among others, a *syntax*, which has the distinction of well-formed formulas, and a *semantics*, the purpose of which is the assignment of a meaning to well-formed formulas.

To each neutrosophic proposition  $p$ , we associate it with an ordered triple components  $\langle t(p), i(p), f(p) \rangle$ , where  $t(p), i(p), f(p) \in [0, 1]$ .  $t(p), i(p), f(p)$  is called truth-degree, indeterminacy-degree and falsity-degree of  $p$ , respectively. Let this assignment be provided by an *interpretation function* or *interpretation*  $NL$  defined over a set of propositions  $P$  in such a way that

$$NL(p) = \langle t(p), i(p), f(p) \rangle.$$

Hence, the function  $NL$  gives the truth, indeterminacy and falsity degrees of all propositions in  $P$ . We assume that the interpretation function  $NL$  assigns to the logical truth  $T : NL(T) = \langle 1, 1, 0 \rangle$ , and to  $F : NL(F) = \langle 0, 0, 1 \rangle$ .

An interpretation which makes a formula true is a *model* of the formula.

The semantics of four neutrosophic propositional connectives is given in Table 1. Note that  $p \leftrightarrow q$  if and only if  $p \rightarrow q$  and  $q \rightarrow p$ .

Table 1  
Semantics of Four Connectives in Neutrosophic Propositional Logic

Connectives	Semantics
$NL(\neg p)$	$\langle f(p), 1 - i(p), t(p) \rangle$
$NL(p \wedge q)$	$\langle \max(t(p), t(q)), \max(i(p), i(q)), \min(f(p), f(q)) \rangle$
$NL(p \vee q)$	$\langle \max(t(p), t(q)), \max(i(p), i(q)), \min(f(p), f(q)) \rangle$
$NL(p \rightarrow q)$	$\langle \min(1, 1 - t(p) + t(q)), \min(1, 1 - i(p) + i(q)), \max(0, f(q) - f(p)) \rangle$

### 2.2 Neutrosophic Predicate Logic

In this section, we will extend our consideration to the full language of first order neutrosophic predicate logic. First we give the formalization of syntax of first order neutrosophic predicate logic as in classical first-order predicate logic.

### 2.2.1 Syntax of Neutrosophic Predicate Logic

**Definition 4** An alphabet of the first order neutrosophic predicate logic consists of seven classes of symbols:

- (1) variables, denoted by lower-case letters, sometimes indexed;
- (2) constants, denoted by lower-case letters;
- (3) function symbols, denoted by lower-case letters, sometimes indexed;
- (4) predicate symbols, denoted by lower-case letters, sometimes indexed;
- (5) Five connectives  $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$  which are called the conjunction, disjunction, negation, implication, and bimplication symbols respectively;
- (6) Two quantifiers, the universal quantifier  $\forall$  (for all ) and the existential quantifier  $\exists$  (there exists);
- (7) The parentheses ( and ).

□

To avoid having formulas cluttered with brackets, we adopt the following precedence hierarchy, with the highest precedence at the top:

$\neg, \forall, \exists$

$\wedge, \vee$

$\rightarrow, \leftrightarrow$

Next we turn to the definition of the first order neutrosophic language given by an alphabet.

**Definition 5** A term is defined as follows:

- (1) A variable is a term.
- (2) A constant is a term.
- (3) If  $f$  is an  $n$ -ary function symbol and  $t_1, \dots, t_n$  are terms, then  $f(t_1, \dots, t_n)$  is a term.

□

**Definition 6** A (well-formed) formula is defined inductively as follows:

- (1) If  $p$  is an  $n$ -ary predicate symbol and  $t_1, \dots, t_n$  are terms, then  $p(t_1, \dots, t_n)$  is a formula (called an atomic formula or, more simply, an atom).
- (2) If  $F$  and  $G$  are formulas, then so are  $(\neg F)$ ,  $(F \wedge G)$ ,  $(F \vee G)$ ,  $(F \rightarrow G)$  and  $(F \leftrightarrow G)$ .
- (3) If  $F$  is a formula and  $x$  is a variable, then  $(\forall xF)$  and  $(\exists xF)$  are formulas.

□

**Definition 7** *The first order neutrosophic language given by an alphabet consists of the set of all formulas constructed from the symbols of the alphabet .*  
□

**Definition 8** *The scope of  $\forall x$  (resp.  $\exists x$ ) in  $\forall xF$  (resp.  $\exists xF$ ) is  $F$ . A bound occurrence of a variable in a formula is an occurrence immediately following a quantifier or an occurrence within the scope of a quantifier, which has the same variable immediately after the quantifier. Any other occurrence of a variable is free.*  
□

### 2.2.2 Semantics of Neutrosophic Predicate Logic

In this section, we study the semantics of neutrosophic predicate logic, the purpose of which is the assignment of a meaning to well-formed formulas. In the neutrosophic propositional logic, an interpretation is an assignment of truth values (ordered triple component) to propositions. In the first order neutrosophic predicate logic, since there are variables involved, we have to do more than that. To define an interpretation for a well-formed formula in this logic, we have to specify two things, the domain and an assignment to constants and predicate symbols occurring in the formula. The following is the formal definition of an interpretation of a formula in the first order neutrosophic predicate logic.

**Definition 9** *An interpretation function (or interpretation) of a formula  $F$  in the first order neutrosophic predicate logic consists of a nonempty domain  $D$ , and an assignment of “values” to each constant and predicate symbol occurring in  $F$  as follows:*

- (1) *To each constant, we assign an element in  $D$ .*
- (2) *To each  $n$ -ary function symbol, we assign a mapping from  $D^n$  to  $D$ . (Note that  $D^n = \{(x_1, \dots, x_n) | x_1 \in D, \dots, x_n \in D\}$ ).*
- (3) *Predicate symbols get their meaning through evaluation functions  $E$  which assign to each variable  $x$  an element  $E(x) \in D$ . To each  $n$ -ary predicate symbol  $p$ , there is a function  $NP(p) : D^n \rightarrow N$ . So  $NP(p(x_1, \dots, x_n)) = NP(p)(E(x_1), \dots, E(x_n))$ .*

□

That is,  $NP(p)(a_1, \dots, a_n) = \langle t(p(a_1, \dots, a_n)), i(p(a_1, \dots, a_n)), f(p(a_1, \dots, a_n)) \rangle$ , where  $t(p(a_1, \dots, a_n)), i(p(a_1, \dots, a_n)), f(p(a_1, \dots, a_n)) \in [0, 1]$ . They are called truth-degree, indeterminacy-degree and falsity-degree of  $p(a_1, \dots, a_n)$  respectively. We assume that the interpretation function  $NP$  assigns to the logical truth  $T : NP(T) = \langle 1, 1, 0 \rangle$ , and to  $F : NP(F) = \langle 0, 0, 1 \rangle$ .

The semantics of four neutrosophic predicate connectives and two quantifiers is given in Table 2. For simplification of notation, we use  $p$  to denote  $p(a_1, \dots, a_i)$ . Note that  $p \leftrightarrow q$  if and only if  $p \rightarrow q$  and  $q \rightarrow p$ .

Table 2

Semantics of Four Connectives and Two Quantifiers in Neutrosophic Predicate Logic

Connectives	Semantics
$NP(\neg p)$	$\langle f(p), 1 - i(p), t(p) \rangle$
$NP(p \wedge q)$	$\langle \min(t(p), t(q)), \min(i(p), i(q)), \max(f(p), f(q)) \rangle$
$NP(p \vee q)$	$\langle \max(t(p), t(q)), \max(i(p), i(q)), \min(f(p), f(q)) \rangle$
$NP(p \rightarrow q)$	$\langle \min(1, 1 - t(p) + t(q)), \min(1, 1 - i(p) + i(q)), \max(0, f(q) - f(p)) \rangle$
$NP(\forall x F)$	$\langle \min t(F(E(x))), \min i(F(E(x))), \max f(F(E(x))), E(x) \in D$
$NP(\exists x F)$	$\langle \max t(F(E(x))), \max i(F(E(x))), \min f(F(E(x))), E(x) \in D$

### 2.3 QoS Model

Different applications generally have different requirements of QoS dimensions. [15,16] investigated the features with which successful companies assert themselves in the competitive world markets. Their result showed that success is based on three essential dimensions: time, cost and quality. [17] associates eight dimensions with quality, including performance and reliability.

In order to be more precise, we give our definitions of the three dimensions. (1) For a Semantic Web services, the capability can be defined as the degree that functional properties a Semantic Web service provide match with the functional properties a Semantic Web service requestor requires; (2) The response time of a Semantic Web service represents the time that elapses between service requests arrival and the completion of that service request. Response time is the sum of waiting time and actual processing time; (3) The trustworthiness of bioinformatics Semantic Web services should capture the consistency, reliability, competence and honesty of the service.

## 3 Architecture of neutrosophic logic based SWS agent

The neutrosophic logic based SWS agent can provide high QoS SWS based on specific ontology. The extensible SWS agent uses centralized client/server architecture internally. But itself can also be and should be implemented as a Semantic Web service based on specific service ontology. The neutrosophic



logic based SWS agent comprises of six components: (a)Registries Crawler; (b)SWS Repository; (c)Inquiry Server; (d)Publish Server; (e)Agent Communication Server; (f)Intelligent Inference Engine. The high level architecture of the neutrosophic logic based SWS agent is shown in Figure 1.

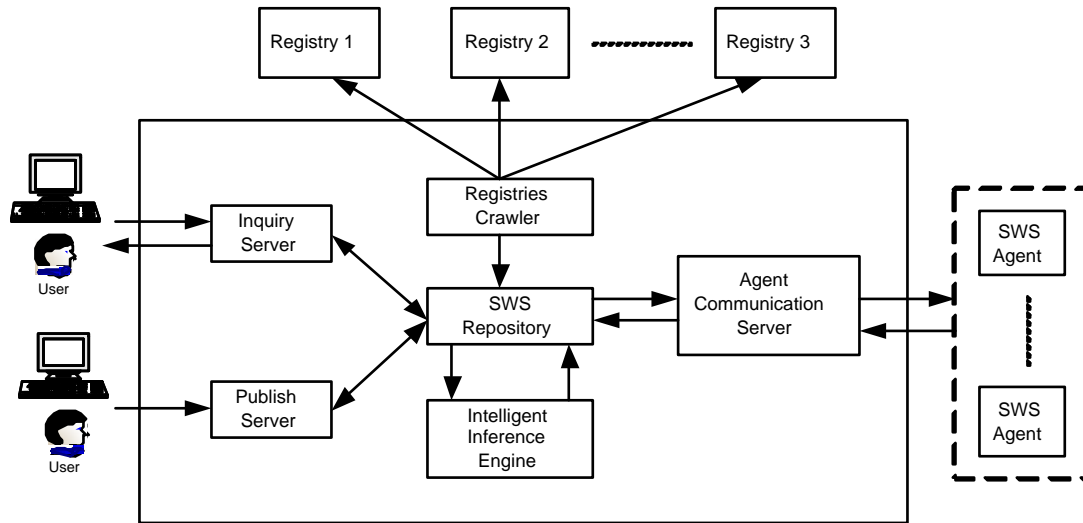


Fig. 1. Architecture of the neutrosophic logic based SWS agent

The Intelligent Inference Engine (IIE) is the core of the neutrosophic logic based SWS agent. The neutrosophic logic based SWS agent is extensible because IIE uses neutrosophic logic inference system to calculate the QoS of the Semantic Web services with multidimensional QoS metrics . IIE gets the degree of capability matching and non-functional properties' values from OWL-S Matching Engine and return back the whole QoS to OWL-S Matching Engine. In the next section, we show the design of the IIE using neutrosophic logic, neural networks and genetic algorithm.

### 3.1 Design of Intelligent Inference Engine

This section shows one implementation of IIE based on neutrosophic logic, neural network and genetic algorithm. A schematic diagram of the four-layered neutrosophic neural network is shown in Figure 3. Nodes in layer one are input nodes representing input linguistic variables. Nodes in layer two are membership nodes. Membership nodes are truth-membership node, indeterminacy-membership node and falsity-membership node, which are responsible for mapping an input linguistic variable into three possibility distributions for that variable. The rule nodes reside in layer three. The last layer contains the output variable nodes [18].

As we mentioned before, the metrics of QoS of Semantic Web services are multidimensional. For illustration of specific ontology based Semantic Web

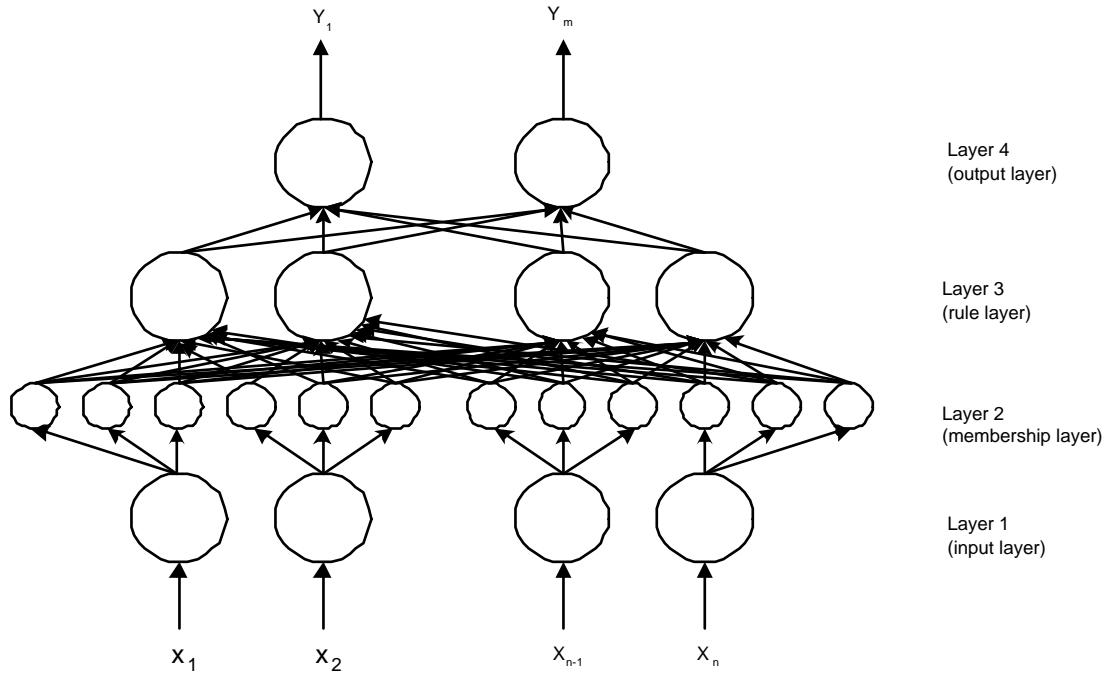


Fig. 2. Schematic diagram of Neutrosophic Neural Network

services for bioinformatics, we decide to use capability, response time and trustworthiness as our inputs and whole QoS as output. The neutrosophic logic system is based on TSK model.

### 3.2 Input neutrosophic sets

Let  $x$  represent capability,  $y$  represent response time and  $z$  represent trustworthiness. We scale the capability, response time and trustworthiness to  $[0,10]$  respectively. The graphical representation of membership functions of  $x$ ,  $y$ , and  $z$  are shown in Figure 4.

### 3.3 Neutrosophic rule bases

Here, we design the neutrosophic rule base based on the TSK model. A neutrosophic rule is shown below:

IF  $x$  is  $I_1$  and  $y$  is  $I_2$  and  $z$  is  $I_3$  THEN  $O$  is  $a_{i,1} * x + a_{i,2} * y + a_{i,3} * Z + a_{i,4}$ .

where,  $I_1, I_2$  and  $I_3$  are in low, middle, and high respectively and  $i$  in  $[1,27]$ . There are totally 27 neutrosophic rules. The  $a_{i,j}$  are consequent parameters which will be obtained by training phase of neutrosophic neural network using genetic algorithm.

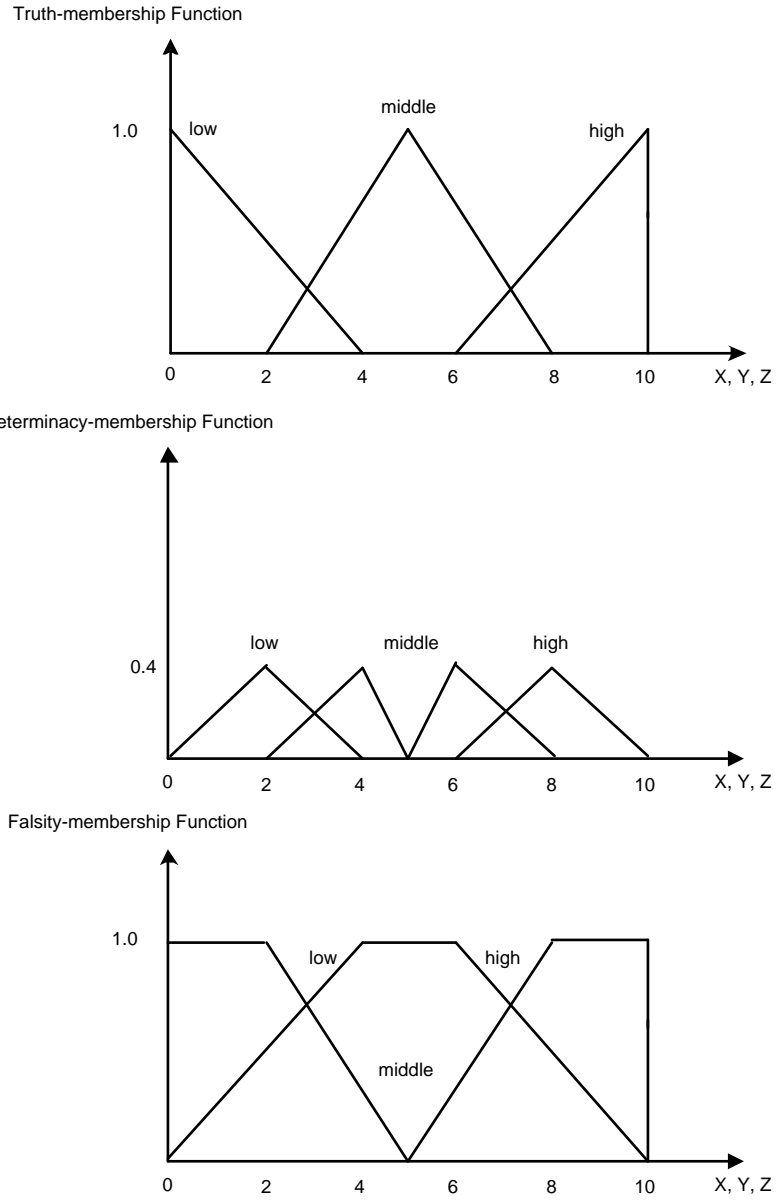


Fig. 3. Membership functions of inputs

### 3.4 Design of deneutrosophication

Suppose, for certain inputs  $x$ ,  $y$  and  $z$ , there are  $m$  fired neutrosophic rules. To calculate the firing strength of  $j$ th rule, we use the formula:

$$W^j = W_x^j * W_y^j * W_z^j, \quad (1)$$

where

$$W_x^j = (0.5 * t_x(x) + 0.35 * (1 - f_x(x)) + 0.025 * i_x(x) + 0.05),$$

$$W_y^j = (0.5 * t_y(y) + 0.35 * (1 - f_y(y)) + 0.025 * i_y(y) + 0.05),$$

$W_z^j = (0.5 * t_z(z) + 0.35 * (1 - f_z(z)) + 0.025 * i_z(z) + 0.05)$ ,  
where  $t_x, f_x, i_x, t_y, f_y, i_y, t_z, f_z, i_z$ , are the truth-membership, falsity-membership, indeterminacy-membership of neutrosophic inputs  $x, y, z$ , respectively.

So the crisp output is:

$$O = \sum_{j=1}^m W^j * (a_{j,1} * x + a_{j,2} * y + a_{j,3} * z + a_{j,4}) / (\sum_{j=1}^m W^j) \quad (2)$$

### 3.5 Genetic algorithms

GA is a model of machine learning which derives its behavior form a metaphor of the processes of evolution in nature. This is done by creation within a machine of a population of individuals represented by chromosomes. Here we use real-coded scheme. Given the range of parameters (coefficients of linear equations in TSK model), the system uses the derivate-free random search-GA to learn to find the near optimal solution by the fitness function through the training data.

- (1) Chromosome: The genes of each chromosome are 108 real numbers (there are 108 parameters in the fuzzy fule base) which are initially generated randomly in the given range. So each chromosome is a vector of 108 real numbers.
- (2) Fitness function: The fitness function is defined as

$$E = 1/2 \sum_{j=1}^m (d_i - o_i)^2 \quad (3)$$

- (3) Elitism: The tournament selection is used in the elitism process.
- (4) Crossover: The system will randomly select two parents among the population, then randomly select the number of cross points, and simply exchange the corresponding genes among these two parents to generate a new generation.
- (5) Mutation: For each individual in the population, the system will randomly select genes in the chromosome and replace them with randomly generated real numbers in the given range.

### 3.6 Simulations

There are two phases for applying a fuzzy neural network: training and predicting. In the training phase, we use 150 data entries as training data set. Each entry consists of three inputs and one expected output. We tune the

Table 3  
 Prediction Result of Neutrosophic Neural Network

Input x	Input y	Input z	Desired output	Real output o
1	0	1	0	0.51
1	2	5	1	1.71
1	4	7	2	2.59
3	2	9	3	3.52
3	6	7	4	3.81
3	10	7	5	4.92
5	8	9	6	5.43
7	10	7	7	5.90
7	10	9	8	6.45
9	10	9	9	7.36

performance of the system by adjusting the size of population, the number of generation and probability of crossover and mutation. Table 1 gives the part of prediction results with several parameters for output o.

In Table 1, No. of generation = 10000, No. of population = 100, probability of crossover = 0.7, probability of mutation = 0.3. The maximum error of prediction result is 1.64. The total prediction error for 150 entries is 19 functions and choosing reasonable training data set which is based on specific application domain can reduce the prediction error a lot. Here the example is just for illustration.

#### 4 Conclusion and future work

In this paper, we discuss the design of Intelligent Inference Engine of extensible neutrosophic logic based SWS agent. The neutrosophic logic based SWS agent supports both the keyword based discovery and capability based discovery of the Semantic Web services. The primary motivation of our work is to solve two challenges facing current Web services advertising and discovering techniques. One is how to locate the Registry hosting required Web service description; another is how to find the required Web service with highest QoS in the located Registry. The neutrosophic logic based SWS agent solves these two problems efficiently and effectively. The neutrosophic logic based SWS agent is built upon the Semantic Web, Web services and neutrosophic logic. The neutrosophic logic based SWS agent could be used in WWW, P2P and Grid infrastructure. The neutrosophic logic based SWS agent is flexible and extensi-

ble. In the future, we plan to extend the architecture of the neutrosophic logic based SWS agent to compute the whole QoS workflow of Semantic Web services to facilitate the composition and monitoring of complex Semantic Web services and apply it to Semantic Web-based bioinformatics applications.

## References

- [1] L. Zadeh, Fuzzy sets, *Inform. and Control* 8 (1965) 338–353.
- [2] I. Turksen, Interval valued fuzzy sets based on normal forms, *Fuzzy Sets and Systems* 20 (1986) 191–210.
- [3] N. Karnik, J. Mendel, Introduction to type-2 fuzzy logic systems, in: *Proceeding of 1998 IEEE Fuzz Conference*, 1998, pp. 915–920.
- [4] Q. Liang, J. Mendel, Interval type-2 fuzzy logic systems: theory and design, *IEEE Transactions On Fuzzy Systems* 8 (2000) 535–550.
- [5] J. Mendel, R. John, Type-2 fuzzy sets made simple, *IEEE Transactions on Fuzzy Systems* 10 (2002) 117–127.
- [6] K. Atanassov, Intuitionistic fuzzy sets, *Fuzzy Sets and Systems* 20 (1986) 87–96.
- [7] K. Atanassov, G. Gargov, Intuitionistic fuzzy logic, *Compt. Rend. Acad. Bulg. Sci.* 43 (1990) 9–12.
- [8] H. Wang, P. Madiraju, Y.-Q. Zhang, R. Sunderraman, Interval neutrosophic sets, *International Journal of Applied Mathematics and Statistics* 3 (2005) 1–18.
- [9] H. Wang, F. Smarandache, Y.-Q. Zhang, R. Sunderraman, Interval neutrosophic logic, submitted.
- [10] K. Sivashanmugam, K. Verma, A. Sheth, J. Miller, Adding semantics to web services standards, in: *International Conference on Web Services (ICWS'03)*, 2003, pp. 395–401.
- [11] J. Gonzalez-Castillo, D. Trastour, C. Bartonili, Description logics for matchmaking of services.  
URL <http://www.hpl.hp.com/techreports/2001/HPL-2001-265.pdf>
- [12] K. Sycara, M. Klusch, S. Widoff, J. Lu, Dynamic service matchmaking among agents in open information environments, *SIGMOD Record* 28 (1) (1999) 47–53.
- [13] T. Payne, M. Paolucci, K. Sycara, Advertising and matching daml-s service descriptions.  
URL  
<http://www.semanticweb.org/SWWS/Program/position/soipayne.pdf>

- [14] F. Smarandache, *A Unifying Field in Logics: Neutrosophic Logic, Neutrosophy, Neutrosophic Set, Neutrosophic Probability and Statistics*, Xiquan, Phonix, 2003, third edition.
- [15] G. Rommel, *Simplicity wins: how Germany's mid-sized industrial companies succeed*, Harvard Business School Press, Boston, Mass, 1995.
- [16] G. Stalk, T. Hout, *Competing against time: how timebased competition is reshaping global markets*, Free Press, New York, 1990.
- [17] D. A. Garvin, *Managing Quality: The Strategic and Competitive Edge*, Free Press, New York, 1988.
- [18] C.-H. Lee, J.-L. Hong, Y.-C. Lin, Type-2 fuzzy neural network systems and learning, *International Journal of Computational Cognition* 1 (4) (2003) 79–90.