How to Interpret Neural Networks In Terms of Fuzzy Logic?

Sompong Dhompongsa[1], Vladik Kreinovich[2], and Hung T. Nguyen[3]

[1]Chiang Mai University, 239 Huay Kaew Road,
Chiang Mai 50200, Thailand, sompongd@chiangmai.ac.th
[2]Dept. Computer Science, University of Texas,
El Paso, TX 79968, USA, vladik@cs.utep.edu
[3]Mathematics Dept., New Mexico State University,
Las Cruces, NM 88003, USA, hunguyen@nmsu.edu

Abstract

Neural networks are a very efficient learning tool, e.g., for transforming an experience of an expert human controller into the design of an automatic controller. It is desirable to reformulate the neural network expression for the input-output function in terms most understandable to an expert controller, i.e., by using words from natural language. There are several methodologies for transforming such natural-language knowledge into a precise form; since these methodologies have to take into consideration the uncertainty (fuzziness) of natural language, they are usually called fuzzy logics.

1 Introduction: It Is Important to Be Able to Reformulate the Neural Network Input-Output Expression in Terms Understandable to a Human Controller

Neural networks are a very efficient learning tool which can input a sequence of input-output patterns $(x^{(k)}, y^{(k)})$ and return a neural network which produces the desired output $y^{(k)}$ for each input $x^{(k)}$. The resulting neural network provides us with the output $f(x)$ for every possible input, not just for the inputs $x^{(k)}$ for which we already know the answer.

In particular, we can use a neural network to transform an experience of an expert human controller into the design of an automatic controller: namely, we record what control values $y^{(k)}$ the expert controller applied for different inputs $x^{(k)}$, and then we train a neural network so that it will be able to generate the same outputs as the expert for all given inputs.

From the mathematical viewpoint, a neural network is a tool which extends (interpolates and extrapolates) the (unknown) input-output function $f(x)$ from its known values $f(x^{(k)}) = y^{(k)}$. There are many possible extensions of a function from finitely many points, so, before we use the neural network for actual control, we would like to check with an expert whether this particular extension is consistent with his/her expertise. A natural way to check this neural network is to apply it to some input x different from $x^{(k)}$ and to ask the expert whether the resulting control value $f(x)$ is reasonable. If it is not reasonable, we ask the expert what a reasonable control is, and re-train the network so that it will produce a reasonable control value for this new input as well. If for this input, the control value provided by a neural network is reasonable, then we test this neural network on other inputs.

In this checking, although we know the exact form of the corresponding input-output function $f(x)$, we still treat the neural network as a "black box", because the expert cannot easily understand the meaning of this function. It would be nice if, in addition to asking the expert to test specific outputs, we could also ask him/her to test the entire expression for $f(x)$. For this to be possible, we must reformulate the neural network expression for $f(x)$ in terms understandable to an expert controller.

The most natural way of representing a knowledge in a way understandable to a human is to represent this knowledge by using words from natural language. There are several methodologies for transforming such natural-language knowledge into a precise form; since these methodologies have to take into consideration the uncertainty (fuzziness) of natural language, they are usually called *fuzzy logics*. So, our problem is to reformulate the description of a neural network in terms of fuzzy logic.

Such a description is also very important to better understand efficient neuro-fuzzy systems such as the Adaptive Network-Based Fuzzy Inference System ANFIS (see, e.g., [9]). This system has a lot of very successful applications (see, e.g., [6, 7, 21, 25, 26]), and

to better understand these successes, it is desirable to understand the neural part of it in commonsense terms.

In this paper, we propose a new interpretation of neural networks in fuzzy logical terms.

2 Existing Representations and Why They Are Not Sufficient

A most widely used neural network uses the following input-output relation:

$$y = W_1 \cdot y_1 + \ldots + W_K \cdot y_K + W_0,$$

where

$$y_k = s_0(w_{k1} \cdot x_1 + \ldots + w_{kn} \cdot x_n + w_{k0}) \quad (1 \le k \le n),$$

and $s_0(z) = (1 + \exp(-z))^{-1}$. In mathematical terms, the resulting function is a composition of linear functions and a non-linear function $s_0(z)$. It is known that linear functions (and, more generally, piece-wise linear functions) can be naturally interpreted in fuzzy logic; see, e.g., [23]. Since an arbitrary continuous function (including the function $s_0(z)$) can be approximated, with any given accuracy, by piece-wise linear functions, A. Di Nola et al. [4, 5] proposed to approximate the neural network input-output function $f(x)$ by functions with can be interpreted in terms of fuzzy logic.

Alternatively, we can use the fact – known from fuzzy control – that functions represented in fuzzy logic terms are universal approximations for arbitrary continuous functions (see, e.g., a survey [13] and references therein). By using any of the corresponding approximation schemes, we can also approximate the neural network input-output function $f(x)$ by functions with can be interpreted in terms of fuzzy logic.

This suggestion is good but not perfect. Indeed, the more accurate we want the approximation to be, the more linear pieces we need, and thus, the more complex the resulting fuzzy-logic interpretation. As a result, when the accuracy increases, the fuzzy-logic representation becomes so complex that it cannot be easily understood by an expert anyway and is, thus, not performing the task for which it was originally designed.

To overcome this difficulty, we must avoid approximation and try to represent the activation function $s_0(z)$ itself in logical terms. Such a representation is not easily possible within the standard approaches to fuzzy logic – we know all the functions which naturally appear within these approaches and none of them leads to $s_0(z)$. We therefore need a further theoretical research in developing appropriate generalizations of fuzzy logic.

In this paper, we develop such a generalization.

3 First Step: Natural Description of Degrees of Belief

We will now proceed to describing our interpretation step-by-step. On each step, we start with motivations, and then we transform this informal (commonsense) motivation into exact formulas.

The first thing we need to describe is the set of all possible degrees of belief in different statements. About some statements S, we know nothing. In other words, we posses no knowledge that would confirm or refute the corresponding statement S. What is a natural degree of uncertainty describing this situation of complete uncertainty, i.e., of "zero" knowledge? The very word "zero" (indicating the absence of knowledge) leads to the following natural choice: to describe the absence of knowledge by the value 0.

Situations when we know nothing are rare. In most situations, we have some information about the statement S whose degree of belief we are estimating. Some pieces of this information confirm S – i.e., serve as arguments in favor of S; other pieces of information may refute S – i.e., serve as arguments against S. How can we describe the corresponding degrees of belief?

These degrees of belief are the easiest to describe in the simplest situation when we only consider arguments in favor of S, and all possible arguments carry the same "weight". In this case, the natural way to describe the corresponding degree of belief is by simply counting the number of arguments that we have. In other words, if we describe the degree of belief corresponding to the single argument by a number $d_0 > 0$, then the degree of belief corresponding to two arguments is naturally described by the value $2d_0$, the degree of belief corresponding to the presence of 10 arguments is naturally described by the value $10d_0$, etc. In principle, we can have arbitrarily many arguments, so for every positive integer n, the value $n \cdot d_0$ can be a degree of belief. Thus, degrees of belief can be arbitrarily large.

In the above discussion, we considered the simplified case when all possible arguments have the same "weights". In reality, different arguments in favor of S may differ in strength. For every argument that somewhat confirms S, we can easily imaging another argument in favor of S which is much weaker than the original one. So, for each value $d_0 > 0$ that can be a possible degree of belief, there is a smaller positive number $d_1 < d_0$ which can also serve as a possible degree of belief. We can always imagine an argument so weak that not only it is weaker than the argument corresponding to d_0, but even two argument of this type, when taken together, are still weaker than the argument corresponding to d_0. Since the two argument taken together are described by the value $2d_1$, we conclude that

$2d_1 > d_0$, i.e., that $d_1 < d_0/2$.

Similarly, for $d_1 > 0$, there exists a possible degree of belief $d_2 < d_1/2$, etc. So, we get a decreasing sequence of positive degrees of belief for which $d_k < d_{k-1}/2$ and hence, $d_k < 2^{-k} \cdot d_0$. In other words, we have a decreasing sequence of positive real numbers that tends to 0.

For each strength level d_k, and for each positive integer n, we can have n arguments of this strength d_k. Thus, for each of these degrees of belief d_k and for every integer n, the value $n \cdot d_k$ also serves as a possible degree of belief. Hence, we arrive at the following situation: we have a decreasing sequence of positive numbers $d_0 > d_1 > \ldots > d_k > \ldots$, $d_k > 0$, that tends to 0 ($d_k \to 0$), and we know that for every positive integer n, the value $n \cdot d_k$ belongs to the set D of possible degrees of belief. The following simple result is true:

Proposition 1. *Let d_k be a decreasing sequence of positive real numbers that tends to 0, and let D be a set that contains all the values $n \cdot d_k$ for every positive integer n and for every k. Then, D is everywhere dense in the set of all positive real numbers, i.e., for every positive real number $r \in R$ and for every $\varepsilon > 0$, there exists a value $d \in D$ for which $|d - r| \le \varepsilon$.*

From the practical viewpoint, we can interpret this proposition as stating that every positive real number can serve as a degree of belief. Indeed, from the practical viewpoint, we cannot describe a real number exactly, and we can probably only approximately describe degrees of belief. Whatever positive real number we pick, when we represent this real number in the computer, we do it with a certain accuracy $\varepsilon > 0$. This possible representation error ε can be very small, but it is always positive. It mean that when we have a value $\widetilde{r}$ inside the computer, the actual real number represented by this value can be any value from the interval $[\widetilde{r} - \varepsilon, \widetilde{r} + \varepsilon]$. According to Proposition 1, no matter how small ε is, within this interval, there always exists a possible degree of belief. So, no matter how accurately we represent a real number, it is always possible that the corresponding computer-represented number represents a possible degree of belief. In other words, from the practical viewpoint, whatever real number we represent in the computer, this number is a possible degree of belief.

In view of this practical fact, there is no practical reason to distinguish between the set of all positive real numbers and the set of all possible positive degrees of belief. Thus, we can simply assume that every positive real number is a positive degree of belief.

Similarly, if we have an argument which has the same "weight" but which is against S, then it is natural to describe the degree of belief corresponding to such a negative argument by the corresponding negative number $-d_0$. Thus, every negative real number is also a possible degree of belief.

We already know that 0 is a possible degree of belief – it corresponds to the absence of knowledge. Thus, every real number – no matter whether it is positive, negative, or equal to 0 – can be viewed as a possible degree of belief. So, we arrive at the following definition:

Definition 1. *By a* degree of belief, *we mean a real number.*

4 Second Step: The Corresponding "Or" Operation

Suppose that we have two statements A and B, we know the degrees of belief a and b assigned to these statements, and these degrees of belief are the only information that we have about A and B. In this situation, what is the natural degree of belief in disjunction $A \vee B$? Since the only information we have about the expert's belief in A and B are the two numbers a and b, these two numbers are the only information that we can use to compute the "natural" degree of belief in $A \vee B$. Thus, this degree of belief should be a function of a and b. This function is called an *"or"-operation.* An "or"-operation is usually denoted by $f_\vee(a, b)$, or, when there is no risk of confusion, simply by $a \vee b$.

What is the natural choice of an "or"-operation? To answer this question, let us again start with the simplest case when all arguments are in favor of S and all arguments have the same strength d_0. In this case, the fact that $a = n \cdot d_0$ and $b = m \cdot d_0$ means that there are n arguments in favor of A and m arguments in favor of B. To describe $a \vee b$, we must count arguments in favor of $A \vee B$. Every argument is favor of A or in favor of B is the argument in favor of $A \vee B$.

In principle, there exist infinitely many potential arguments, so in general, it is hardly probable that when we pick n arguments out of infinitely many and then m out of infinitely many, the corresponding sets will have a common element. Thus, when we have other information about A and B, it is reasonable to assume that every argument in favor of A is different from every argument in favor of B. Under this assumption, the total number of arguments in favor of A and arguments in favor of B is equal to $n + m$. Hence, the natural degree of belief in $A \vee B$ is equal to $(n + m) \cdot d_0 = n \cdot d_0 + m \cdot d_0 = a + b$.

Thus, when the values a and b are *commeasurable*, i.e., can be represented as $n \cdot d_0$ and $m \cdot d_0$ for some integers n and m, then $a \vee b = a + b$. Since within any given accuracy $\varepsilon > 0$, every two real numbers can be approx-

imated by two commeasurable ones (e.g., by rational ones), we can therefore conclude that $a + b$ is a natural "or" operation for arbitrary real numbers.

Definition 2. *By an* "or"-operation*, we mean* $a \vee b = a + b$.

This conclusion is in good accordance with the known results about "or"-operations: under certain reasonable conditions (so-called "strict Archmedean") every "or"-operation is indeed isomorphic to $a + b$ (see, e.g., [11, 22]), and under more general conditions, every "or" can be approximated – within any given accuracy – by an operation which is isomorphic to $a + b$; see, e.g., [20].

It is worth mentioning that while in fuzzy logic, we normally only use non-negative degrees of belief, in the first expert systems such as MYCIN [2, 24], negative values were used as well, so our use of negative values follows a natural idea.

5 Third Step: The "And" Operation

When describing the "or"-operation, we implicitly assumed that all the statements come from reliable experts. In reality, different experts are reliable to different degrees.

In general, we believe in a statement made by an expert if we believe this expert *and* the expert believes in this statement. To formalize this idea, we must select an "and"-operation, i.e., a function $\&$ which, given degrees of belief a and b in two statements A and B, generates a (reasonable) degree of belief $a \,\&\, b$ in the conjunction "A and B". In terms of this operation, our degree of belief in a statement A made by an expert is equal to $w \,\&\, a$, where w is our degree of belief in this expert, and a is the expert's degree of belief in the statement A. What are the natural properties of the "and"-operation?

First, since $A \,\&\, B$ means the same as $B \,\&\, A$, it is reasonable to require that the corresponding degrees $a \,\&\, b$ and $b \,\&\, a$ should coincide, i.e., that the "and"-operation be commutative.

Second, when an expert makes two statements B and C, then our resulting degree of belief in $B \vee C$ can be computed in two different ways:

- We can first compute *his* degree of belief $b \vee c$ in $B \vee C$, and then us the "and"-operation to generate our degree of belief $w \,\&\, (b \vee c)$.
- We can also first generate our degrees $w \,\&\, b$ and $w \,\&\, c$, and then use an "or"-operation to combine these degrees, arriving at $(w \,\&\, b) \vee (w \,\&\, c)$.

It is natural to require that both ways lead to the same degree of belief, i.e., that the "and"-operation be distributive with respect to $\vee$.

Third, if we have reasons to believe in the expert (i.e., if our degree of belief w in this expert is non-negative), then the more the expert beliefs in a certain statement, the more reasons we have to believe in it. In other words, if $w \geq 0$ and $b \leq c$, then $w \,\&\, b \leq w \,\&\, c$, and if $w > 0$ and $b < c$, then $w \,\&\, b < w \,\&\, c$. It can also happen that an expert is consistently wrong (i.e., $w \leq 0$). In this case, it is reasonable to require that if $b \leq c$, then $w \,\&\, b \geq w \,\&\, c$. As a result, we arrive at the following definition:

Definition 3. *A function* $\& : R \times R \to R$ *is called an* "and"-operation *if it satisfies the following three properties:*

- *it is commutative, i.e.,* $a \,\&\, b = b \,\&\, a$;
- *it is distributive, i.e.,* $a \,\&\, (b \vee c) = (a \,\&\, b) \vee (a \,\&\, c)$; *and*
- *it is monotonic, i.e.:*
  - *if* $a \geq 0$ *and* $b \leq c$, *then* $a \,\&\, b \leq a \,\&\, c$;
  - *if* $a > 0$ *and* $b < c$, *then* $a \,\&\, b < a \,\&\, c$; *and*
  - *if* $a \leq 0$ *and* $b \leq c$, *then* $a \,\&\, b \geq a \,\&\, c$.

Proposition 2. *Every "and"-operation has the form* $a \,\&\, b = C \cdot a \cdot b$ *for some* $C > 0$.

This expression can be further simplified. Indeed, let us introduce a new scale of degrees of belief $a' = C \cdot a$, and let us see what the operations look like in the new scale. The value $a' \,\&'\, b'$ is equal to $C \cdot (a \,\&\, b)$, where $a = a'/C$ and $b = b'/C$ are the values on the old scale which correspond to a' and b'. We know that $a \,\&\, b = C \cdot a \cdot b$, hence

$$a' \,\&'\, b' = C \cdot (C \cdot (a'/C) \cdot (b'/C)) = a' \cdot b'.$$

Similarly, we can show that in the new scale, $a' \vee' b' = a' + b'$.

Thus, without losing generality, we will assume that $a \vee b = a + b$ and $a \,\&\, b = a \cdot b$.

6 Fourth Step: Crisp Truth Value

We know that "true" and "true" is "true", and that "false" and "false" is "false". Thus, we arrive at the following definition:

Definition 4. *A positive degree of belief* e_0 *is called a* crisp *value if* $e_0 \,\&\, e_0 = e_0$.

Proposition 3. $e_0 = 1$.

7 Fifth Step: Implication Operation

From the commonsense viewpoint, an implication $A \to B$ is a statement C such that if we add C to B, we get A. This understanding leads to the following natural definition of an implication operation $a \to b$:

Definition 5. *A function $\to: R \times R \to R$ is called an* implication operation *if for all a and b, we have $(a \to b) \,\&\, a = b$.*

Clearly, if $a = 0$ and $b \neq 0$, the value $a \to b$ is not defined; if $a = 0$ and $b = 0$, then any real number can be equal to $a \to b$. When $a \neq 0$, this definition uniquely defines the implication operation:

Proposition 4. *When $a \neq 0$, we have $a \to b = b/a$.*

8 Final Step: Negation Operations

The negation operation must satisfy the following natural requirements:

- first, that the negation of $A \vee B$ mean the same as "not A" and "not B", and
- second, that if we believe in B more than in A, then we should believe more in "not A" than in "not B":

Definition 6. *A function $\neg : R \to R$ is called a negation operation if it satisfies the following two properties:*

- *de Morgan property: for every a and b, $\neg(a \vee b) = (\neg a) \,\&\, (\neg b)$; and*
- *monotonicity: if $a < b$, then $\neg a > \neg b$.*

Proposition 5. *Every negation operation has the form $\neg(a) = \exp(-k \cdot a)$ for some $k > 0$.*

In addition to such negation operations, we already had a natural negation operator:

Definition 7. $\neg_2(a) = -a$.

The third negation operator comes from the fact that negation $\neg A$ can be viewed as a particular case of implication, $A \to F$, for a crisp (specifically, false) value F. Thus, we have the third negation:

Definition 8. $\neg_3(a) = a \to e_0$.

In other words, $\neg_3(a) = 1/a$.

It is worth mentioning that in many logics, several negation operations naturally appear. For example:

- weak and strong negations are used in intuitionistic logic; see, e.g., [3, 10];
- classical negation and negation as failure are used as two different negation operations in logic-programming based knowledge representation formalisms (see, e.g., [29]), and
- several implications (hence several negations) are present in linear logic [8, 28].

The last two examples are not surprising since there is a natural relationship between fuzzy logic and logic programming [12, 19, 15, 17, 18, 27] and between fuzzy and linear logics [14, 16].

9 Resulting Logical Interpretation of Neural Networks

In many applications of expert systems, it is necessary to be cautious. In this case, if we are not sure about some statement A, then, instead of using the original degree of belief a that A is true, we would rather use the degree of belief in a somewhat stronger statement "it is impossible that A is not true". How can we estimate this new degree of belief?

By definition of a negation operation, the degree of belief that A is not true is equal to $\neg a$. The very fact that we are cautious means that we have some prior reasons to suspect that A is not true. A natural way to describe the degree corresponding to these prior reasons is to use the crisp value e_0. If we combine the current degree of belief and this prior knowledge, we conclude that the combined degree of belief in A is equal to $\neg a \vee e_0$.

The above "strong negation" means, crudely speaking, that instead of counting arguments that A is possible, we prefer to count arguments that $\neg A$ is impossible, i.e., that $\neg A$ implies a contradiction. The corresponding degree of belief corresponds to the implication-based negation, so we get $(\neg a \vee e_0) \to e_0$. One can easily check that this is the standard activation function $s_0(a) = 1/(1 + \exp(-k \cdot a))$.

Since the linear combination

$$w_1 \cdot y_1 + \ldots + w_K \cdot y_K$$

is now interpreted in purely logical form, as

$$(w_1 \,\&\, y_1) \vee \ldots \vee (w_K \,\&\, y_K),$$

we thus have a natural purely logical interpretation of neural networks.

This interpretation is a disjunction of conjunctions, so it is similar to well-known CNF and DNF forms of a propositional expression.

10 Proofs: Main Ideas

Proof of Proposition 2. For every a, the function $f_a(b) \stackrel{\text{def}}{=} a \,\&\, b$ is additive $f_a(b) + f_a(c) = f_a(b + c)$. Since it is also monotonic, we conclude (see, e.g., [1]) that this function is linear, i.e., $a \,\&\, b = f_a(b) = C(a) \cdot b$. Commutativity implies that $C(a) \cdot b = C(b) \cdot a$ for all a and b, hence, for $b = 1$, $C(a) \cdot 1 = C(1) \cdot a$, and $a \,\&\, b = C(a) \cdot b = C \cdot a \cdot b$. From monotonicity, we conclude that $C > 0$.

Proof of Proposition 5. De Morgan condition leads to the functional equation $\neg(a + b) = \neg(a) \cdot \neg(b)$. All monotonic solutions to this equations are known; see, e.g., [1].

Acknowledgments
This work was supported in part by NASA under cooperative agreement NCC5-209 and grant NCC 2-1232, by NSF grants CDA-9522207, ERA-0112968 and 9710940 Mexico/Conacyt, by Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-00-1-0365, and by Grant No. W-00016 from the U.S.-Czech Science and Technology Joint Fund.

References

[1] J. Aczel, *Lectures on functional equations and their applications*, Academic Press, New York, London, 1966.

[2] B. G. Buchanan and E. H. Shortliffe, *Rule-Based Expert Systems*, Addison-Wesley, Reading, MA, Menlo Park, CA, 1984.

[3] J. P. Cleave, *A study of logics*, Claredon Press, Oxford, 1991.

[4] A. Di Nola and P. Amato, *Neural networks, McNaughton Functions, and Fuzzy Systems*, University of Salerno Technical Report, June 2001.

[5] A. Di Nola, R. Tagliaferri, and R. Belohlávek, *Fuzzy Neural Networks Based on Fuzzy Logic Algebras Valued Relations*, University of Salerno Technical Report, 2001.

[6] H. Gassoumi, J. J. Ellington, H. T. Nguyen, and N. R. Prasad, "A soft computing approach to insects classification in the cotton field", *Proceedings of the International Symposium on Medical Informatics and Fuzzy Technology MIF'99*, Hanoi, Vietnam, August 27–29, 1999, pp. 454–485.

[7] H. Gassoumi, J. J. Ellington, H. T. Nguyen, and N. R. Prasad, "Integrated pest management system", In: H. Mohanty and C. Baral (eds.), *Trends in Information Technology, Proceedings of the International Conference on Information Technology ICIT'99, Bhubaneswar, India, December 20–22, 1999*, Tata McGraw-Hill, New Delhi, 2000, pp. 126–131.

[8] J.-Y. Girard, "Linear logic", *Theoretical Computer Science*, 1987, Vol. 50, pp. 1–102.

[9] J.-S. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice Hall, Upper Saddle River, NJ, 1997.

[10] *Ifs: conditionals, beliefs, decision, chance, and time*, D. Reidel Co., Dordrecht, Holland, 1981.

[11] G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, Upper Saddle River, NJ, 1995.

[12] O. Kosheleva, V. Kreinovich, and H. T. Nguyen, "Mamdani's Rule: a "weird" use of "and" as implication justified by modern logic", *Sixth International Fuzzy Systems Association World Congress*, San Paulo, Brazil, July 22–28, 1995, Vol. 1, pp. 229–232.

[13] V. Kreinovich, G. C. Mouzouris, and H. T. Nguyen, "Fuzzy rule based modeling as a universal approximation tool", In: H. T. Nguyen and M. Sugeno (eds.), *Fuzzy Systems: Modeling and Control*, Kluwer, Boston, MA, 1998, pp. 135–195.

[14] V. Kreinovich, H. T. Nguyen, and P. Wojciechowski, "Fuzzy Logic as Applied Linear Logic", *BUSEFAL*, No. 67, July 1996, p. 4–13.

[15] H. T. Nguyen, O. M. Kosheleva, and V. Kreinovich, "Is the success of fuzzy logic really paradoxical? Or: Towards the actual logic behind expert systems", *International Journal of Intelligent Systems*, 1996, Vol. 11, No. 5, pp. 295–326.

[16] H. T. Nguyen and V. Kreinovich, "Fuzzy Logic, Logic Programming, and Linear Logic: Towards a New Understanding of Common Sense", *Proceedings of NAFIPS'96, Biennial Conference of the North American Fuzzy Information Processing Society*, Berkeley, CA, June 20-22, 1996, pp. 546–550.

[17] H. T. Nguyen and V. Kreinovich, "Using Gelfond-Przymusinska's Epistemic Specifications to Justify (Some) Heuristic Methods Used in Expert Systems and Intelligent Control", *Soft Computing*, 1997, Vol. 1, No. 4, pp. 198–209.

[18] H. T. Nguyen, V. Kreinovich, and B. Bouchon-Meunier, "Soft Computing Explains Heuristic Numerical Methods in Data Processing and in Logic Programming", *Working Notes of the AAAI Symposium on Frontiers in Soft Computing and Decision Systems*, Boston, MA, November 8–10, 1997, pp. 40–45.

[19] H. T. Nguyen, V. Kreinovich, D. E. Cooke, Luqi, and O. Kosheleva, "Towards combining fuzzy and logic programming techniques", In: H. P. Nguyen and A. Ohsato (eds.), *Proc. Vietnam-Japan Bilateral*

Symposium on Fuzzy Systems and Applications VJ-FUZZY'98, HaLong Bay, Vietnam, 30th September–2nd October, 1998, pp. 482–489.

[20] H. T. Nguyen, V. Kreinovich, and P. Wojciechowski, "Strict Archimedean t-Norms and t-Conorms as Universal Approximators", *International Journal of Approximate Reasoning*, 1998, Vol. 18, Nos. 3–4, pp. 239–249.

[21] H. T. Nguyen, N. R. Prasad, V. Kreinovich, and H. Gassoumi, "Some Practical Applications of Soft Computing and Data Mining", In: A. Kandel, H. Bunke, and M. Last (eds.), *Data Mining and Computational Intelligence*, Springer-Verlag, Berlin, 2001, pp. 273–307.

[22] H. T. Nguyen and E. A. Walker, *First Course in Fuzzy Logic*, CRC Press, Boca Raton, FL, 1999.

[23] I. Perfilieva and A. Tonis, "Functional system in fuzzy logic formal theory", *Busefal*, 1995, Vol. 64, pp. 42–50.

[24] E. H. Shortliffe, *Computer-based medical consultation: MYCIN*, Elsevier, New York, 1976.

[25] M. Siddaiah, M. A. Lieberman, S. E. Hughs, and N. R. Prasad, "A soft computing approach to classification of trash in ginned cotton", *Proceedings of the 8th International Fuzzy Systems Association World Congress IFSA'99*, Taipei, Taiwan, August 17–20, 1999, pp. 151–155.

[26] M. Siddaiah, M. A. Lieberman, S. E. Hughs, and N. R. Prasad, "Identification of trash types in ginned cotton using neuro fuzzy techniques", *Proceedings of the 8th IEEE International Conference on Fuzzy Systems FUZZ-IEEE'99*, Seoul, Korea, August 22–25, 1999, Vol. 2, pp. 738–743.

[27] S. A. Starks, H. T. Nguyen, V. Kreinovich, H. P. Nguyen, and M. Navara, "Strong Negation: Its Relation to Intervals and Its Use in Expert Systems", In: G. Alefeld and R. A. Trejo (eds.), *Interval Computations and its Applications to Reasoning Under Uncertainty, Knowledge Representation, and Control Theory, Proceedings of MEXICON'98, Workshop on Interval Computations, 4th World Congress on Expert Systems*, Mexico City, Mexico, 1998.

[28] A. S. Troelstra, *Lectures on linear logic*, CSLI, Stanford, 1992.

[29] G. Wagner, "Logic programming with strong negation and inexact predicates", *J. Logic Computat.*, 1991, Vol. 1, No. 6, pp. 835–859.