Selecting the Most Adequate Fuzzy Operation for Explainable AI: Empirical Fact and Its Possible Theoretical Explanation

Orsolya Csiszár[a], Gábor Csiszar[b], Martine Ceberio[c], and Vladik Kreinovich[c]
[a]Aalen Univ. of Applied Sciences, Germany, orsolya.csiszar@hs-aalen.de
[b]Inst. of Material Phys., Univ. of Stuttgart, Stuttgart, Germany, gaborcsiszar@gmail.com
[c]Comp. Sci., Univ. of Texas at El Paso, El Paso, TX 79968, USA, \{mceberio,vladik\}@utep.edu

Abstract

A reasonable way to make AI results explainable is to approximate the corresponding deep-learning-generated function by a simple expression formed by fuzzy operations. Experiments on real data show that out of all easy-to-compute fuzzy operations, the best approximation is attained if we use an operation $a+b-0.5$ (limited to the interval $[0,1]$). In this paper, we provide a possible theoretical explanation for this empirical result.

Keywords: Explainable AI, Fuzzy logic, Fuzzy operations

1 Formulation of the General Problems

Need for Explainable AI (XAI). Modern deep-learning based systems achieve spectacular results; see, e.g., [4]. Not only they place chess and Go much better than any human player – which is great but not very useful, they also perform many human tasks better then human specialists. For example, when analyzing X-rays, for many diseases, an AI system makes much fewer diagnostic mistakes than most human doctors.

Based on these highly publicized successes, it may seems, at first glance, that we should start replacing medical doctors – and many other specialists – with computers. But there is an important reason why we are not rushing to perform this replacement, and the gist of this reason is that AI systems sometimes make mistakes.

Yes, human doctors also make mistakes, but these mistakes are somewhat repairable. Indeed, when a medical doctor describes his/her diagnosis, he/she explains the reasons for this decision. A patient can ask for a second opinion, and the two doctors can compare their reasons and come up with a more reliable conclusion.

In contrast, a usual deep-learning-based AI system does not provide any reasons for its decisions. So, in contrast to opinions of human experts, for AI-based decisions, there is no way to argue and thus to improve the situation.

To improve this situation, and to fully utilize the potential of AI systems, it is important to make sure that these system provide us:

- not only with decisions,
- but also with justifications for these decisions.

In order words, the important task is to transform current AI systems into *Explainable* AI (XAI, for short).

A natural approach to explainable AI. On the one hand, the task of converting AI to XAI is a recent important challenge, unlike any challenges that we humans encountered before. However, on the other hand, this challenge is not that unusual. Similar challenges happen in physics and in all other sciences all the time:

- experimentalists find empirical laws that describe a certain physical phenomenon,
- then comes theoreticians who provide an explanation for these laws; see, e.g., [3, 9].

Such examples are plentiful; let us give just two examples:

- Kepler discovered laws that describe the planet motion, and Newton provided the general theory that explained all these laws.
- Physicists discovered empirical formulas that describe the atomic spectra, and Schroedinger came up with a general equation that explains all these formulas.

This analogy provides a natural path to XAI: since deep learning provides us with an effective algorithm for solving the problem, let us come up with a human-understandable explanation for this algorithm.

Which explanations are human-understandable: enter fuzzy techniques. To implement this idea, we need to clarify what it means to have a human-understandable explanation. To clarify this, let us again use medical doctors as an example – or any other human experts.

One of the reasons why we need to use medical doctors as an example is to avoid a widely spread misunderstanding – that comes from the fact that many researchers working in AI are mathematicians by training, and in mathematics, every statement is either true or false. This fundamental idea is at the core of mathematical reasoning. Because of this, quite a few AI researchers try to apply the same true-false dichotomy to computer reasoning as well.

In contrast, medical and other experts do not deal with absolutely true or absolutely false statements. In most cases, even when a doctor confidently describes his/her diagnosis, the doctor understands that there is always a chance that this diagnosis is wrong. For good doctors, the chance is very small, but it is still there – and when a situation is unusual, chances are reasonably high. This is how we reason, and we expect AI-based system to come up with similar types of reasoning – reasoning about statements about which we have a certain degree of certainty.

Techniques for reasoning with such statements are known as *fuzzy techniques*; see, e.g., [1, 5, 6, 7, 8, 11]. These techniques were pioneered in the 1960s by Lotfi Zadeh who suggested to describe the expert's degree of certainty in a statement by a number from the interval $[0,1]$. This idea is very natural; indeed:

- in the computers, "true" is usually represented as 1, and "false" as 0;
- so naturally degrees of certainty intermediate between full certainty in the statement and fuzzy certainty in its negations should be described by intermediate numbers.

Reasoning involves dealing with complex statements, i.e., statements obtained from the original ones by using logical connectives such as "and", "or", "if and only if", etc. In our reasoning, we mostly use unary and binary connectives, i.e., connectives that combine two statements into a single one: e.g., combining statements A and B into a complex statement $A \,\&\, B$.

- The "true"-"false" versions of such connectives transform two truth value (both are equal to 0 or 1) into a single truth value of a complex statement.
- Correspondingly, a fuzzy analogue f of this connective should take two numbers a and b from the interval $[0,1]$ and transform them into a new number $f(a,b)$ from the same interval.

Which fuzzy operations are most adequate for this purpose? There are many functions of this type, i.e., in mathematical terms, unary functions $f : [0,1] \to [0,1]$ and binary functions $f : [0,1] \times [0,1] \to [0,1]$. Which of them is most adequate for the use in AI? This is the main questions with which we deal in this paper.

The structure of this paper is as follows.

- First, in Section 2, we describe some general ideas about selecting an appropriate fuzzy operation, and come up with a class of operations that are consistent with these ideas.
- In Section 3, we describe the empirical results of applying these operations.
- Finally, in Section 4, we provide a possible theoretical explanation for these empirical results.

2 General Ideas about Selecting Appropriate Fuzzy Operations

Main idea: we need operations that are the fastest to compute. In many applications, decisions need to be made fast. It is therefore reasonable to consider operations whose computation is as fast as possible. Similar ideas are one of the main reasons why modern neural algorithms:

- mostly use easiest-to-compute activation function $\max(0,x)$ (known as *Rectified Linear Unit*, ReLU, for short)
- instead of the previously used more-difficult-to-compute sigmoid function $1/(1+\exp(-x))$.

How can we implement this idea: general discussion. How can we achieve this goal? In a computer, every computation is performed as a sequence of arithmetic operations. The fastest arithmetic operation is addition/subtraction. So, if we want to only consider fast-to-computer fuzzy operations, it is reasonable to restrict ourselves to operations that can be computed by using only additions/subtractions.

How can we implement this idea for unary operations? If we have only one input z, then, to apply addition or subtraction, we need a second argument – either z itself or a constant.

- If we use z twice, we get either $z - z = 0$ or $z + z = 2z$. Since we want the values to always be between 0 and 1, we cannot use the expression $2z$: its value for $z = 1$ is 2.
- If we use a constant c, then we can have $z + c$ or $c - z$. In the first case, we cannot have all the values within the interval $[0,1]$. In the second case, we can – but only in one case, when $c = 1$.

So, the only appropriate unary operation is $1 - z$, which is the usual fuzzy negation.

How we can implement this idea for binary operations. Since we need a binary operation, that take into account both inputs a and b, the simplest case is to use one addition, which leads to the operation $a + b$.

To be on the safe side, let us have a family of possible operations, so that we will be able to select the most adequate one. With one addition, we cannot get anything other than $a + b$, so we need a second addition. If we add again a or b, we will not get the whole family, so instead, we need to add a constant c. This way, we will get a family of functions $f(a,b) = a + b + c$ characterized by the parameter c.

Taking into account that we need fuzzy operations. Here, in contrast to the unary case, we have an additional complication, related to the fact that the values of a fuzzy operation must be within the interval $[0,1]$. Already the simplest function $a + b$ produced results outside this interval – e.g., for $a = b = 1$, we get $a + b = 2$. So, to get a fuzzy operation, we need to restrict the value of the function by the interval $[0,1]$. A natural way to do it is to replace each value which is outside the interval by the closest value from this interval, i.e.:

- replace negative values z with 0, and
- replace values z larger than 1 with 1.

The result of this replacement can be described as

$$S(z) \stackrel{\text{def}}{=} \max(\min(z,1),0).$$

Resulting selection. Our conclusion is that we should select the fuzzy operations

$$f_c(a,b) = S(a + b + c) \qquad (1)$$

corresponding to some constant c.

Comment. It is worth mentioning that:

- for $c = 0$, the expression (1) takes the form $\min(a + b, 1)$; this is one of the well known fuzzy "or"-operations (aka t-conorms);
- for $c = -1$, the expression (1) takes the form $\max(a + b - 1, 0)$; this is one of the well known fuzzy "and"-operations (aka t-norms).

Remaining question. The remaining question is: which value c should we choose? To answer this question, we ran some experiments [2].

3 Which Fuzzy Operation Is the Most Adequate for XAI: Experimental Results

What we did. We used 12 classification problems from the UCI Machine Learning Repository [10]. For each of these problems, we tried out best to approximate the results by combination of operations (1) corresponding to different values c.

Of course, the results of classification are 0 and 1, and fuzzy computations return a value between 0 and 1. So, for each values z produced by a sequence of fuzzy operations is transformed into 0 or 1 depending on whether 0 or 1 are closer to z:

- if $z < 0.5$, we transform it into 0, and
- if $z > 0.5$, we transform it into 1.

Results. Interestingly, in all the cases, the best approximation – with the smallest number of terms for given accuracy – was attained for $c = -0.5$. With this choice, we get really short expressions. Here are some examples, in which x_i is the value of the i-th input normalized to the interval $[0,1]$:

- For the Breast Cancer problem, we got the approximating expression
$$f_c(f_c(x_6, x_{34}), f_c(x_{28}, x_{34})).$$
- For the Diabetes problem, we got the approximating expression
$$1 - f_c(x_1, x_6).$$
- For the King-Rook vs. King-Pawn problem, we got the approximating expression
$$f_c(f_c(x_9, x_{34}), f_c(x_{22}, x_{34})).$$

- For the Vote problem, we got the approximating expression

$$f_c(f_c(x_{11},x_{37}),f_c(x_{25},x_{31})).$$

4 A Possible Theoretical Explanation for the Empirical Results

Idea. In binary classification problem, we have two possible results:

- one of them marked as 1: e.g., that the patient has breast cancer, and
- another one is marked as 0: e.g., that the patient does not have breast cancer.

This corresponds to the case when we ask whether a patient has breast cancer.

On the other hand, we can formulate the exact same classification problem by asking the opposite question: whether the patient is cancer-free. If we formulate the question this way, then it is natural to mark absence of breast cancer as 1 and breast cancer as 0. In other words, when the original problem leads to the value z, the reformulated problem should lead to the value $1-z$, i.e., "not z".

Of course, in the original problem, it makes sense to consider inputs whose presence increases the possibility that a patient has breast cancer as having:

- the value 1 if they are present and
- the value 0 if they are absent.

If we reformulate this problem as asking whether a patient is cancer-free then, vice versa, we should consider inputs for which, vice versa:

- the value 1 if these inputs are absent and
- the value 0 if they are present.

In other words, instead of each original input a, we should have the new input $1-a$, which is "not a".

So, we have two formulations of the exact same problem:

- in the first formulation, we have inputs $x_1,\ldots,x_n$, and we want to produce an answer

$$z = f(z_1,\ldots,x_n);$$

- in the second formulation, we have inputs $x'_i = 1-x_i$, and we want to produce the answer

$$z' = f'(x'_1,\ldots,x'_n)$$

for which $z' = 1-z$.

Both formulations are absolutely equivalent. So, the complexity of our approximation should not depend on which of the two formulations we use. In particular, this means that:

- if we can use only one fuzzy operation for the original problem, i.e., if we have $z = f_c(x_1,x_2)$,
- then we should be able to use only one fuzzy operation for the reformulated problem as well, i.e., we should have $z' = f_c(x'_1,x'_2)$, where $z' = 1-z$, $x'_1 = 1-x_1$, and $x'_2 = 1-x_2$.

Substituting $z' = 1-z$, $x'_1 = 1-x_1$, and $x'_2 = 1-x_2$ into the expression $z' = f_c(x'_1,x'_2)$, we get

$$1-z = f_c(1-x_1,1-x_2).$$

We know that $z = f_c(x_1,x_2)$, so we get

$$1-f_c(x_1,x_2) = f_c(1-x_1,1-x_2).$$

Substituting the expression (1) for f_c into this formula and considering only the cases when the value is strictly between 0 and 1 (i.e., when $f_c(x_1,x_2) = x+1+x_2+c$), we get

$$1-(x_1+x_2+c) = 1-x_1+1-x_2+c,$$

i.e., if we open parentheses:

$$1-x_1-x_2-c = 1-x_1+1-x_2+c.$$

If we subtract $1-x_1-x_2$ from both sides, we get $-c = 1+c$, hence $2c = -1$ and $c = -0.5$.

This is indeed the empirically best value. So, its appearance can indeed be explained by the fact that, in general:

- the problem of detecting an effect (e.g., breast cancer) is equivalent to
- the problem of detecting the absence of this effect (e.g., the absence of breast cancer).

Acknowledgments

This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change

for Preparing a New Generation for Professional Practice in Computer Science), HRD-1834620 and HRD-2034030 (CAHSI Includes), EAR-2225395, and by the AT&T Fellowship in Information Technology.

It was also supported by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478, and by a grant from the Hungarian National Research, Development and Innovation Office (NRDI).

References

[1] R. Belohlavek, J. W. Dauben, and G. J. Klir, *Fuzzy Logic and Mathematics: A Historical Perspective*, Oxford University Press, New York, 2017.

[2] O. Csiszár, L. S. Pusztaházi, L. Dénes-Fazakas, M. S. Gashler, V. Kreinovich, and G. Csiszár, "Uninorm-like parametric activation functions for human-understandable neural models", *Knowledge-Based Systems*, 2023, Vol. 260, Paper 110095, https://doi.org/10.1016/j.knosys.2022.110095

[3] R. Feynman, R. Leighton, and M. Sands, *The Feynman Lectures on Physics*, Addison Wesley, Boston, Massachusetts, 2005.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, Massachusetts, 2016.

[5] G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, Upper Saddle River, New Jersey, 1995.

[6] J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions*, Springer, Cham, Switzerland, 2017.

[7] H. T. Nguyen, C. L. Walker, and E. A. Walker, *A First Course in Fuzzy Logic*, Chapman and Hall/CRC, Boca Raton, Florida, 2019.

[8] V. Novák, I. Perfilieva, and J. Močkoř, *Mathematical Principles of Fuzzy Logic*, Kluwer, Boston, Dordrecht, 1999.

[9] K. S. Thorne and R. D. Blandford, *Modern Classical Physics: Optics, Fluids, Plasmas, Elasticity, Relativity, and Statistical Physics*, Princeton University Press, Princeton, New Jersey, 2021.

[10] University of California–Irvin, *Machine Learning Depositary*, https://archive.ics.uci.edu/ml/

[11] L. A. Zadeh, "Fuzzy sets", *Information and Control*, 1965, Vol. 8, pp. 338–353.