2-1-2022

Fuzzy Or Neural, Type-1 Or Type-2 – When Each Is Better: First-Approximation Analysis

Vladik Kreinovich
The University of Texas at El Paso, vladik@utep.edu

Olga Kosheleva
The University of Texas at El Paso, olgak@utep.edu

Comments:

Technical Report: UTEP-CS-22-15

Fuzzy Or Neural, Type-1 Or Type-2 – When Each Is Better: First-Approximation Analysis

Vladik Kreinovich and Olga Kosheleva

Abstract In many practical situations, we need to determine the dependence between different quantities based on the empirical data. Several methods exist for solving this problem, including neural techniques and different versions of fuzzy techniques: type-1, type-2, etc. In some cases, some of these techniques work better, in other cases, other methods work better. Usually, practitioners try several techniques and select the one that works best for their problem. This trying often requires a lot of efforts. It would be more efficient if we could have a priori recommendations about which technique is better. In this paper, we use the first-approximation model of this situation to provide such a recommendation.

1 Formulation of the Problem

Need for data processing. In many practical situations, we are interested in a quantity y which is difficult – or even impossible – to measure directly. For example, we may be interested in the distance to a faraway star, in the amount of oil in a given well, in tomorrow's temperature, etc. Since we cannot directly measure this quantity, a natural idea is to measure it indirectly – i.e., to measure the values of easier-to-measure related quantities $x_1, \ldots, x_n$, and then provide the best estimate for y based on the results $\widetilde{x}_i$ of these measurements. For this purpose, we need to know the dependence $y = f(x_1, \ldots, x_n)$ between the desired quantity y and the auxiliary easier-to-measure quantities x_i.

The resulting computation of the estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ based on the measurement results $\widetilde{x}_i$ is what is usually called *data processing*.

Vladik Kreinovich and Olga Kosheleva
University of Texas at El Paso, 500 W. University, El Paso, TX 79968, USA
e-mail: vladik@utep.edu, olgak@utep.edu

Sometimes, the dependence is known, but not always. In some practical situations. we know the desired dependence $y = f(x_1, \ldots, x_n)$ – at least approximately, with reasonable accuracy.

For example, to estimate the distance to a faraway star, we can measure the direction to this star in two different seasons, when the Earth is at two opposite sides of the Sun. In this case, the distance can be obtained by using known trigonometric formulas. Similarly, we know the equations that describe the atmosphere's dynamics.

However, in many other situation, we do not know the dependence – or we know the approximate dependence, but this approximate dependence is too very accurate. For example, no accurate formula is known for predicting future changes in the country's economy.

Need for – generally understood – machine learning. In situations when we do not know the desired dependence, we need to determine this dependence from the experimental data. In other words, we collect situations k in which we know the values $y^{(k)}$ and $x_i^{(k)}$ of both y and x_i, and we want to find the dependence $y = f(x_1, \ldots, x_n)$ that fits all these data, i.e., for which, for all k, we have $y^{(k)} \approx f\left(x_1^{(k)}, \ldots, x_n^{(k)}\right)$. The determination of the dependence from the available data is known as *machine learning*; see, e.g., [3].

Often, in addition to the values $y^{(k)}$ and $x_i^{(k)}$, we have some information about the dependence $y = f(x_1, \ldots, x_n)$. For example, often we know that the desired dependence is linear or quadratic – or belongs to a known few-parametric class of dependencies. Often, we know some imprecise ("fuzzy") rules that describe this dependence, e.g., that if x_1 is low and x_2 is high, then y is medium.

Many techniques are used to solve this problem. Many different techniques are used to solve this "machine learning" problem of determining the dependence from the experimental data.

In many situations, neural network techniques work the best – especially techniques of deep learning; see, e.g., [5]. In other situations, better results are obtained if we use fuzzy techniques (see, e.g., [2, 6, 7, 9, 10, 13]) – i.e., when we first use fuzzy techniques to translate the imprecise of-then rules into a precise first-approximation dependence, and then adjust the parameters of this dependence so as to fit the available data. There are many version of fuzzy techniques: we can use the traditional fuzzy techniques, we can use intuitionistic fuzzy techniques (see, e.g., [1]), we can use type-2 fuzzy approach [7], etc. There are many other machine learning techniques; see, e.g., [3].

For each of these techniques:

- sometimes, this technique works well – and its results are better than others, while
- in other situations, this techniques is not working as well.

Remaining challenge. At present, it is not clear a priori which of the techniques will work better. Practitioners usually try different techniques – and then use the technique that works the best for a given situation. This trying of several different techniques takes a lot of efforts.

It is therefore desirable to be able to decide when each of the available techniques is better – and thus, in effect, come up with a new combined methodology that would enable us to utilize the best features of each technique without the need to try them all.

What we do in this paper. In this paper, on the example of fuzzy-neural dichotomy, we show how such a combination can be attained.

2 Analysis of the Problem

Neural and fuzzy machine learning: what is the main difference? What is the main difference between neural and fuzzy approaches? In both cases, we adjust the parameters of the model to fit all the data:

- in a neural network, we adjust the parameters of all the neurons,
- in the fuzzy case, we adjust the parameters of the membership functions (and, sometimes, the parameters of the corresponding "and"- and "or"-operations, also known as t-norms and t-conorms).

We often even use the same techniques for this adjustment: e.g., gradient descent. So what is the main different between these two techniques?

The main difference is that:

- In the neural network, a change in each parameter of each neuron, in principle, affects all possible values of the resulting function $f(x_1, \ldots, x_n)$. In this sense, all these parameters are *global*.
- In contrast, in fuzzy techniques, a change in a membership function – e.g., in a membership function describing when x_1 is small – only affects the values $f(x_1, \ldots, x_n)$ corresponding to small x_1. In this sense, all these parameters are *local*.

What is the main difference between type-1, type-2, etc.? From this general viewpoint, what is the main difference between different versions of fuzzy techniques? For example, if we use triangular membership functions – and not necessarily symmetric ones – then we need 3 parameters to describe each function:

- the value v_{0-} at which the membership function starts increasing,
- the value v_1 at which the membership function reaches its largest value 1, and
- the value v_{0+} at which the membership function reaches the value 0 and stops decreasing.

If we use symmetric membership functions, we only need two parameters to describe each membership function, since in this case,

$$v_1 = \frac{v_{0-} + v_{0+}}{2}.$$

If we use trapezoid membership functions, in the general case, we need four parameters:

- the value v_{0-} at which the membership function starts increasing,
- the value v_{1-} at which the membership function first reaches its largest value 1,
- the value v_{1+} at which the membership function starts decreasing, and
- the value v_{0+} at which the membership function reaches the value 0 and stops decreasing.

For symmetric trapezoid functions, we only need three parameters.

In the case of interval-values membership functions $\left[\underline{\mu}(x), \overline{\mu}(x)\right]$, we need, in effect, to describe two different membership functions $\underline{\mu}(x)$ are $\overline{\mu}(x)$ and thus, we need twice as many parameters. For example, if we use symmetric trapezoid membership functions, we need $2 \cdot 3 = 6$ parameters to describe each membership function.

From this viewpoint, the main difference between different fuzzy representations is in how many parameters we use to describe the local behavior of the desired function in the corresponding region.

Now, we are ready to reformulate the above challenge in precise mathematical terms. In view of the above analysis, in order to select a proper technique, we need to decide:

- first, whether it is better to describe the dependence globally (as, e.g., in the neural network approach) or to divide it into regions (as in fuzzy approach) and have a separate description in each region;
- second, if it is more efficient to divide into regions, what is the best size of a region, and how many parameters should we use to describe the dependence in each region.

Let us describe thus reformulated problem in precise terms.

3 First-Approximation Model and the Resulting Recommendation

Simplifying assumption. For simplicity, let us consider a 1-D version of the problem, when we want to find a good approximation to an unknown function of one variable $y = f(x)$.

How many parameters can we use? Of course, the more parameters we use, the more accurately we can represent a function. However, the more parameters we use, the more time it will take to process the data, the more space will be needed to store all these parameters. So, in practice, there is a limit N to how many parameters we can use.

From this viewpoint, the question is: what is the best approximation that we can attain for the given number of parameters?

Gauging approximation accuracy. Most real-life dependencies are smooth, even analytical; see, e.g., [4, 12]. Such functions can be expanded in Taylor series

$$f(x) = a_0 + a_1 \cdot (x - x_0) + a_2 \cdot (x - x_0)^2 + \ldots,$$

and a natural way to approximate such a function is to use the first few terms in its Taylor series, i.e., to use the approximation

$$f(x) \approx a_0 + a_1 \cdot (x - x_0) + a_2 \cdot (x - x_0)^2 + \ldots + a_{p-1} \cdot (x - x_0)^{p-1}. \qquad (1)$$

This is a standard way to approximate real-life dependencies in physics (see, e.g., [4, 12]), this is how computers compute functions like $\exp(x)$, $\sin(x)$, etc.

This approximation means that we ignore the following terms in the Taylor series. The largest of these terms is the term proportional to $(x - x_0)^p$. If we divide the whole interval of possible values of x – whose size we will denote by I – into regions of smaller size s, then this ignored term has the size s^p.

The expression s^p thus describes the size of the approximation error.

Resulting optimization problem. Once we select the degree of the approximating polynomial – i.e., the value p – we will know that to describe the dependence on each subregion, we need p parameters.

Overall, we have N parameters. Thus, we can have N/p subregions.

The size s of each subregion can be obtained if we divide the size I of the whole interval by the number N/p of the subregions. Thus, this size is equal to

$$s = \frac{I}{N/p} = \frac{p \cdot I}{N}.$$

So, the resulting approximation error is equal to

$$s^p = \left(\frac{p \cdot I}{N}\right)^p. \qquad (2)$$

The values I and N are given, the only parameter that we can control is p. Thus, we must select the value p for which the approximation error (2) attains the smallest possible value.

Solving the resulting optimization problem. The function $\ln(x)$ is strictly increasing. Thus, minimizing the expression (2) is equivalent to minimizing its logarithm

$$\ln(s^p) = p \cdot \left(\ln(p) + \ln\left(\frac{I}{N}\right)\right). \qquad (3)$$

Differentiating this expression by p and equating the derivative to 0, we conclude that

$$\ln(p) + \ln\left(\frac{I}{N}\right) + 1 = 0.$$

By applying $\exp(x)$ to both sides of this equality, we conclude that

$$p \cdot \frac{I}{N} \cdot e = 1,$$

hence

$$p = \frac{N}{I \cdot e}. \tag{4}$$

In this case, the optimal size of the subregion is equal to

$$s = \frac{p \cdot I}{N} = \frac{1}{e}. \tag{5}$$

So, we can make the following conclusions.

First conclusion: local or global? The optimal size of the region does not depend on the desired accuracy:

- if this size is sufficiently small – smaller than $1/e$ in some natural units – then it is better to use global techniques like neural networks;
- if this size is larger, then it is better to use local techniques like fuzzy.

In other words, in qualitative terms:

- if the inputs vary a lot, then local (e.g., fuzzy) techniques are better;
- on the other hand, if the inputs do not deviate much, global (e.g., neural) techniques are better.

Second conclusion: how to reach better accuracy? Suppose that we have a local model – e.g., a fuzzy model with a fixed number of triangular membership functions. This model provides some accuracy. What should we do if we want higher accuracy?

Of course, for this purpose, we will need to use more parameters. In general, these parameters can be allocated differently:

- we can divide the original interval into a larger number of subintervals – i.e., use a larger number of membership functions,
- or we can keep the same number of subintervals, but provide a more sophisticated description of the function on each subinterval, description that requires more parameters – e.g., use type-2 fuzzy, or use a more general class of membership functions.

Our analysis shows that the second idea leads to better results. If we had 3 or 5 membership functions before, we should continue to use the same number of membership functions – but make these functions more sophisticated.

Comment. Our result may explain why we humans always divide all objects into 7 ± 2 classes – i.e., in effect, use a subdivision into a fixed number of subregions; see, e.g., [8, 11].

Thus, we provide a natural explanation of this feature of human data processing.

4 Future Work

What we have analyzed is the first approximation, an approximation in which:

- we did not take into account specific features of each application area, and
- we only took into account number of regions and number of parameters.

Additional analysis is needed:

- Additional analysis is needed to decide between different local techniques with the same number of parameters: e.g., should we go to type-2 or should we use more sophisticated membership functions.
- Additional analysis is also needed to decide which of "global" machine learning techniques would work better in a given situation.

We hope that this simple paper will inspire researchers to extend our results and to solve remaining challenges.

Acknowledgements 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), and HRD-1834620 and HRD-2034030 (CAHSI Includes), 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).

The authors are greatly thankful to Yuriy Kondratenko for his encouragement, and, of course, to Janusz Kacprzyk for his numerous ideas and discussions that motivated this work.

References

1. K. Atanassov, *Intuitionistic Fuzzy Sets: Theory and Applications*, Springer Verlag, Berlin, Heidelberg, 1999.
2. R. Belohlavek, J. W. Dauben, and G. J. Klir, *Fuzzy Logic and Mathematics: A Historical Perspective*, Oxford University Press, New York, 2017.
3. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.
4. R. Feynman, R. Leighton, and M. Sands, *The Feynman Lectures on Physics*, Addison Wesley, Boston, Massachusetts, 2005.
5. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Leaning*, MIT Press, Cambridge, Massachusetts, 2016.
6. G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, Upper Saddle River, New Jersey, 1995.
7. J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions*, Springer, Cham, Switzerland, 2017.
8. G. A. Miller, "The magical number seven plus or minus two: some limits on our capacity for processing information", *Psychological Review*, 1956, Vol. 63, No. 2, pp. 81–97.
9. H. T. Nguyen, C. L. Walker, and E. A. Walker, *A First Course in Fuzzy Logic*, Chapman and Hall/CRC, Boca Raton, Florida, 2019.
10. V. Novák, I. Perfilieva, and J. Močkoř, *Mathematical Principles of Fuzzy Logic*, Kluwer, Boston, Dordrecht, 1999.

11. S. K. Reed, *Cognition: Theories and Application*, Wadsworth Cengage Learning, Belmont, California, 2010.
12. K. S. Thorne and R. D. Blandford, *Modern Classical Physics: Optics, Fluids, Plasmas, Elasticity, Relativity, and Statistical Physics*, Princeton University Press, Princeton, New Jersey, 2017.
13. L. A. Zadeh, "Fuzzy sets", *Information and Control*, 1965, Vol. 8, pp. 338–353.