Fuzzy Techniques Explain the Effectiveness of ReLU Activation Function in Deep Learning

Julio C. Urenda[1,2], Olga Kosheleva[3], and Vladik Kreinovich[2]
Departments of [1]Mathematical Science, [2]Computer Science,
and [3]Teacher Education
University of Texas at El Paso, 500 W. University
El Paso, TX 79968, USA,
{jcurenda,olgak,vladik}@utep.edu

Abstract

In the last decades, deep learning has led to spectacular successes. One of the reasons for these successes was the fact that deep neural networks use a special Rectified Linear Unit (ReLU) activation function $s(x) = \max(0, x)$. Why this activation function is so successful is largely a mystery. In this paper, we show that common sense ideas – as formalized by fuzzy logic – can explain this mysterious effectiveness.

1 Formulation of the Problem

How neural networks work: a brief general description. An artificial neural network (see, e.g., [2, 3]) is a network of computational elements called *neurons*. Each neuron transforms the inputs $x_1, \ldots, x_n$ into a value

$$y = s(w_1 \cdot x_1 + \ldots + w_n \cdot x_n + w_0),$$

where:

- w_i are real numbers known as *weights*, and
- $s(x)$ is a function – usually (non-strictly) increasing – that is called *activation function*.

Some neurons process the input data – i.e., usually, the measurement results. Other neurons take, as input, the results of processing by some other neurons, etc.

How neural networks work: some specifics. It is known that neural networks have the *universal approximation property*, i.e., that for a sufficient number of neurons and for an appropriate selection of weights, they can approximate any continuous function with any given accuracy. The process of determining the appropriate weights is known as *training*.

Which activation function should we select. The effectiveness of training depends on the selection of the activation function.

- Traditionally, neural networks used the function $s(x) = 1/(1 + \exp(-x))$ coming from biological neurons [2, 3].
- However, in the last decades, it turns out that the use of a *Rectified Linear Unit (ReLU)* $s(x) = \max(0, x)$ leads to a much more effective learning. The use of ReLU was one of the factors contributing to the spectacular successes of deep learning; see, e.g., [3].

Remaining problem. Why ReLU is so effective is, however, to a large extent a mystery.

What we do in this paper. This is the problem that we deal with in this talk. Specifically, we use ideas from fuzzy logic (see, e.g., [1, 4, 5, 6, 7, 8]) to provide a possible explanation for ReLU effectiveness.

2 Possible Explanation

2.1 Plan

Our explanation will come in two steps:

- first, we will use fuzzy techniques to narrow down the class of possible activation functions;
- then, we apply general ideas from calculus to the class selected on the first step, and show that this indeed leads to the ReLU activation function.

2.2 Towards an explanation: first step

Data is usually known with some uncertainty. Inputs to data processing are known with some uncertainty. Indeed, as we have mentioned, these inputs usually come from measurements, and measurements are never absolutely accurate. Because of this uncertainty, the same actual value may lead to slightly different measurement results.

For example, if the actual value of the voltage is 1.0 V and we measure it with accuracy 10%, we can get measurement results 1.09 or 0.92.

The data uncertainty should minimally affect the result of data processing. It is reasonable to require that the resulting difference should affect the result of data processing as little as possible.

What this implies for a neuron. In particular, with respect to processing by a single neuron, this means that if the values x and x' are close, then the values $s(x)$ and $s(x')$ should also be close.

How to describe this requirement in precise terms. The natural-language word "close" does not have a precise mathematical meaning; this word is imprecise ("fuzzy"). To describe knowledge conveyed by such terms, Lotfi Zadeh invented a special technique called *fuzzy logic*. This technique takes into account that:

- in contrast to precise terms like "positive" which are always either true or not,
- if you ask the person about an imprecise term like "close", this person will often say that this property is satisfied only to some degree.

Some pairs of numbers are very close, some are somewhat close, etc.

To capture these intermediate degrees of confidence – intermediate between absolutely false and absolutely true – Zadeh took into account that in a computer:

- "true" is usually represented by 1, while
- "false" is usually represented by 0.

So, it is natural to represent intermediate degrees of confidence by numbers intermediate between 0 and 1.

In general, to describe each such property, we therefore need to ask the person who uses this term, for each possible value v of the corresponding quantity, to mark, on the scale from 0 to 1, the degree $\mu(v)$ to which this value satisfies this property. The resulting function $\mu(v)$ is known as a *membership function*, or, alternatively, as a *fuzzy set*.

In our case, closeness depends on how different are the numbers x and x', i.e., it depends on the distance $v = |x - x'|$ between these two numbers. So, the degree of closeness has the form $\mu(|x - x'|)$ for an appropriate membership function $\mu(v)$. The further away the two points, i.e., the larger the distance between them, the less confident we are that these two points are close. Thus, the function $\mu(v)$ shouold be strictly decreasing.

This way, we for every two values x and x', we get:

- the degree of confidence $\mu(|x - x'|)$ in the statement "x and x' are close", and
- the degree of confidence $\mu(|s(x) - s(x')|)$ in the statement "$s(x)$ and $s(x')$ are close".

We need to interpret the if-then combination of these two statements. In general, a natural way to interpret the statement "if A then B" is that our degree of confidence in the statement B should be at least as large as our degree of confidence in the statement A. In our case, this means that for all possible real numbers x and x', we must have

$$\mu(|s(x) - s(x')|) \geq \mu(|x - x'|). \tag{1}$$

What this inequality implies for the activation function. Since, as we have mentioned, the membership function corresponding to "close" is strictly decreasing. Thus, the inequality (1) is equivalent to the following inequality:

$$|s(x) - s(x')| \le |x - x'|. \tag{2}$$

What this inequality implies for the derivative of the activation function. Neural network training is usually based on the gradient descent, i.e., on using the derivatives. Because of this, the activation function $s(x)$ is usually selected to be smooth (differentiable) – or at least differentiable almost everywhere. Let us analyze what the inequality (2) implies for the derivative $s'(x)$ of the activation function.

If we take $x' = x + h$ for some small $h > 0$, then $|x - x'| = h$. Since $x < x'$ and the function $s(x)$ is monotonic, we get $s(x) < s(x')$, thus $|s(x) - s(x')| = s(x + h) - s(x) \ge 0$. So, the inequality (2) takes the form

$$0 \le s(x + h) - s(x) \le h. \tag{3}$$

If we divide both sides of this inequality by h, we get

$$0 \le \frac{s(x + h) - s(x)}{h} \le 1. \tag{4}$$

In the limit $h \to 0$, we get

$$0 \le s'(x) \le 1. \tag{5}$$

2.3 Towards an explanation: final step

Let us show how the fuzzy-motivated inequality (5) leads to the explanation of ReLU effectiveness – i.e., to the explanation of why ReLU is, in some reasonable sense, optimal.

What can we say, in general, about an optimal choice. In general, according to calculus, the maximum of a function $F(X)$ in a given region is attained:

- either at the local maximum, where all partial derivatives of this function are 0s,
- or at the border of this region.

Case when we have a large number of constraints. In situations where we have many constraints, each of which decreases the region size, the resulting region is very small. So, the probability that it contains a local maximum is also very small. Thus, in most cases, the maximum is attained at the border, i.e., where (at least) one the inequality constraints turns into an equality. So,

to find the maximum of a function in a region, it is, in most cases, sufficient to find its maximum on the border of this region.

We can apply the same argument to the function $F(X)$ restricted to the border and conclude that most probably, the maximum is attained when another inequality constraint becomes an equality, etc.

In general, the maximum is most probably attained at the point where most – if not all – inequality constraints turn into equalities.

Let us apply this general conclusion to our case. Let us apply the above conclusion to our case. The set of all activation functions is determined by inequality constraints $0 \le s'(x) \le 1$ corresponding to different x. Thus, whatever optimality criterion we use, the optimal activation function most probably corresponds to the situation when each of these inequalities turns into an equality, i.e., when for each x, we either have $s'(x) = 0$ or $s'(x) = 1$. Here:

- In regions where $s'(x) = 0$, the function $s(x)$ is constant.
- In regions where $s'(x) = 1$, we have $s(x) = x + c$ for some constant c.

Thus, the optimal activation function must consists of regions in which it is either constant or have the form $s(x) = x + c$.

What are the simplest functions of this type? The fewer regions, the simpler the function.

The simplest case is when we have only one region. However, in this case, the activation function $s(x)$ would be linear and so, we will not be able to represent non-linear functions.

Thus, the simplest case when we *can* represent non-linear functions is when we have two regions:

- on one of them $s(x)$ is constant,
- on another one $s(x) = x + c$.

Each such two-region function is linearly equivalent to ReLU.

Thus, we indeed have a fuzzy-based explanation for the success of ReLU.

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] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.

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

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

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

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

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

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