3-1-2022

One More Physics-Based Explanation for Rectified Linear Neurons

Jonatan Contreras
The University of Texas at El Paso, jmcontreras2@utep.edu

Martine Ceberio
The University of Texas at El Paso, mceberio@utep.edu

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

Comments:

Technical Report: UTEP-CS-22-39

One More Physics-Based Explanation for Rectified Linear Neurons

Jonatan Contreras, Martine Ceberio, and Vladik Kreinovich

Abstract The main idea behind artificial neural networks is to simulate how data is processed in the data processing devoice that has been optimized by million-years natural selection – our brain. Such networks are indeed very successful, but interestingly, the most recent successes came when researchers replaces the original biology-motivated sigmoid activation function with a completely different one – known as rectified linear function. In this paper, we explain that this somewhat unexpected function actually naturally appears in physics-based data processing.

1 Formulation of the Problem

Neural networks: main idea. One of the objectives of Artificial Intelligence is to enrich computers with intelligence, i.e., with the ability to intelligently process information and make decisions. A natural way to do it is to analyze how we humans make intelligent decisions, and to borrow useful features of this human-based data processing and decision making.

We humans process information in the brain, by using special cells called *neurons*. Different values of the inputs are presented as electric signals – in the first approximation, the frequency of pulses is proportional to the value of the corresponding quantity. In the first approximation, the output signal y of a neuron is related to its inputs $x_1, \ldots, x_n$ by a relation

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

Jonatan Contreras, Martine Ceberio, and Vladik Kreinovich
Department of Computer Science, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA
e-mail: jmcontretas2@miners.utep.edu, mceberio@utep.edu, vladik@utep.edu

where w_i are constants and $s(z)$ is a non-linear function called the *activation function*. Outputs of most neurons serve as inputs to other neurons.

Devices that simulate such biological networks of neurons are known as *artificial neural networks*, or simply *neural networks*, for short.

Choice of activation functions. In the first approximation, the activation function of biological neurons has the form $s(z) = 1/(1+\exp(-z))$; this function is known as a sigmoid. Because of this fact, at first, artificial neural networks used this activation function; see, e.g., [1].

However, later, it turns out that it much more efficient to use different activation function – so-called Rectified Linear (ReLU) function $s(z) = \max(0,z)$; see, e.g., [3].

Comment. It is known that the use of rectified linear activations in neural networks is equivalent to using similar functions $s(z) = \max(a,z)$ and $s(z) = \min(a,z)$ for some constant a – functions that can be obtained from the rectified linear function by shifts of the input and the output and, if needed, by changing the signs of input and/or output.

Question. How can we explain the success of rectified linear activation functions?

What we do in this paper. In this paper, we show from the physics viewpoint, the rectified linear function is a very natural transformation.

2 Analysis of the Problem

Many physical quantities are bounded. In most cases, when we write down formulas relating physical quantities, we implicitly assume that thee quantities can take any possible real values. A good example of such a formula is a formula $d = v \cdot t$ that related the distance d traveled by an inertial object, the object's velocity v, and the duration t of this object's travel.

In reality, all these quantities are bounded:

- the distance cannot be larger than the current size of the Universe,
- the velocity cannot be larger than the speed of light, and
- the duration cannot be larger than the lifetime of the Universe;

see, e.g., [2, 4].

Of course, in most practical problems we can safely ignore these fundamental bounds, but in practical applications, there are usually more practical bounds that need to be taken into account. For example, if we plan a schedule for a truck driver:

- the velocity v is bounded by the speed limits on the corresponding roads, and
- the duration t is limited by the fact that a long drive, the driver will be too tired to continue – and the driver's reaction time will decrease to a dangerous level, because of which regulations prohibit long workdays.

How can we take these bounds into account? Suppose that by analyzing a simplified model, a model that does not take the bound a on a quantity z into account, we came up with an optimal-within-this-model value z_0 of this quantity.

How can we now take the bound into account? One possibility would be to start from scratch, and to re-solve the problem while taking the bound into account. However, this is often too time-consuming: if we could do this, there would be no need to first solve a simplified model.

Since we cannot re-solve this problem in a more realistic setting, a more practical approach is to transform the possible un-bounded result z_0 of solving the simplified problem into a bounded value z. Let us describe the corresponding transformation by $z = f(z_0)$.

Which transformation $f(z)$ should we choose? To answer this question, let us consider natural requirements on the function $f(z)$.

The result of this transformation must satisfy the desired constraint. The whole purpose of the transformation $z = f(z_0)$ is produce the value z that satisfies the desired constraint $z \le a$. Thus, we must have $f(z_0) \le a$ for all possible values z_0.

The transformation must preserve optimality. If the solution z_0 of the simplified model – the model that does not take bounds into account – already satisfies the desired constraint $z_0 \le a$, this means that this value is optimal even if we take the constraint into account.

So, in this case, we should return this same value, i.e., take $z = z_0$.

Monotonicity. In many practical situations, the order between different values of quantity has a physical meaning; e.g.:

- larger velocity leads to faster travel,
- smaller energy consumption means saving money and saving environment, etc.

It is therefore reasonable to require that the desired transformation $f(z_0)$ preserve this order, i.e., that if $z_0 \le z_0'$, then we should have $f(z_0) \le f(z_0')$.

Let us see what we can conclude based on these three requirements.

3 Main Result

Proposition. *Let a be a real number, and let $f(z_0)$ be a function from real numbers to real numbers that satisfies the following three requirements:*

- *for every z_0, we have $f(z_0) \le a$;*
- *for all $z_0 \le a$, we have $f(z_0) = z_0$; and*
- *if $z_0 \le z_0'$, then we have $f(z_0) \le f(z_0')$.*

Then, $f(z_0) = \min(a, z_0)$.

Comment. Vice versa, it is easy to show that the function $f(z_0) = \min(a, z_0)$ satisfies all three requirements.

Proof. To prove this proposition, let us consider two possible cases: $z_0 \leq a$ and $z_0 > a$.

In the first case, when $z_0 \leq a$, then from the second requirement, it follows that $f(z_0) = z_0$. In this case, we have $\min(a, z_0) = z_0$, so indeed $f(z_0) = \min(a, z_0)$.

In the second case, when $z_0 > a$, then from the first requirement, it follows that $f(z_0) \leq a$. On the other hand, since $a < z_0$, the third requirement implies that $f(a) \leq f(z_0)$. The second requirement implies that $f(a) = a$, thus we get $a \leq f(z_0)$. From $f(z_0) \leq a$ and $a \leq f(z_0)$, we conclude that $f(z_0) = a$. In this case, we have $\min(a, z_0) = a$, so we indeed have $f(z_0) = \min(a, z_0)$.

The proposition is proven.

Conclusion. The above proposition shows that the function $f(z_0) = \min(a, z_0)$ naturally follows from a simple physical analysis of the problem.

We have mentioned that the use of this function is equivalent to using the rectified linear activation function. Thus, the rectified linear activation function also naturally follows from physics.

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), 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).

References

1. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.
2. R. Feynman, R. Leighton, and M. Sands, *The Feynman Lectures on Physics*, Addison Wesley, Boston, Massachusetts, 2005.
3. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, Massachusetts, 2016.
4. 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.