Why Rectified Linear Neurons: Two Convexity-Related Explanations

Jonatan Contreras, Martine Ceberio, Olga Kosheleva,
Vladik Kreinovich, and Nguyen Hoang Phuong

Abstract At present, the most efficient machine learning technique is deep learning, in which non-linearity is attained by using rectified linear functions $s_0(x) = \max(0,x)$. Empirically, these functions work better than any other nonlinear functions that have been tried. In this paper, we provide a possible theoretical explanation for this empirical fact. This explanation is based on the fact that one of the main applications of neural networks is decision making, when we want to find an optimal solution. We show that the need to adequately deal with situations when the corresponding optimization problem is feasible – i.e., for which the objective function is convex – uniquely selects rectified linear activation functions.

1 Formulation of the Problem

Rectified linear neurons are very successful. At present, the most successful machine learning technique is deep neural networks; see, e.g., [4]. In general, in neural networks, signals go through two types of transformations: linear transformations and non-linear transformation described by the so-called *activation function* $x \mapsto s(x)$.

Deep neural networks mostly used *rectified linear* (ReLU) activation functions

$$s_0(x) = \max(0,x). \tag{1}$$

Jonatan Contreras, Martine Ceberio, Olga Kosheleva, Vladik Kreinovich
Department of Computer Science, University of Texas at El Paso, 500 W. University
El Paso, TX 79968, USA,
e-mail: jmcontreras2@utep.edu, mceberio@utep.edu, olgak@utep.edu, vladik@utep.edu

Nguyen Hoang Phuong
Division Informatics, Math-Informatics Faculty, Thang Long University, Nghiem Xuan Yem Road
Hoang Mai District, Hanoi, Vietnam, e-mail: nhphuong2008@gmail.com

The main reason for this choice is that empirically, these activation function have been most successful.

But why are they successful? From the theoretical viewpoint, this empirical success is a challenge: why are these activations functions more successful than others? Are there activation functions that we have not tried yet – which will be even more successful?

Important comment. Before we start analyzing this question, we should mention that the fact that we have linear transformations before and after each application of an activation function implies that the same results that we obtain by using rectified linear activation function $s_0(x)$ can also obtained by neurons that use shifted and scale versions of this function:

$$s_1(x) = b_0 + b_1 \cdot x + b_2 \cdot s_0(a_0 + a_1 \cdot x), \tag{2}$$

i.e., if we use functions of the type:

$$s_1(x) = b_0 + a_- \cdot (x - x_0) \text{ for } x \le x_0; \tag{3}$$

$$s_1(x) = b_0 + a_+ \cdot (x - x_0) \text{ for } x \ge x_0, \tag{4}$$

corresponding to some values b_0, a_-, a_+, and x_0.

What is known and what we do in this paper. There are some theoretical explanations of why rectified linear neurons are so successful: e.g., in [2, 7, 8], it was proven that the rectified linear activation functions are, in some reasonable sense, optimal. This explanation is based on the idea that the relative quality of different data processing techniques – in particular, the relative quality of neural networks using different activation functions – should not change if we change all the numerical values by changing the measuring units and/or the starting points for measuring the corresponding quantities.

In this paper, we provide yet another theoretical explanation for this empirical success – this time, an explanation based on computational efficiency and convexity.

2 Why Convexity

Need for optimization. In practice, we always want to find the best possible solution. In precise terms, which solution is better and which is worse is usually described in numerical terms, by assigning a number to each possible solution, so that a solution with the largest (or smallest) value of this numerical characteristic is the best. The mapping that assigns such a number to each alternative x is known as the *objective function* $f(x)$. For example, a company tries to maximize its profit, an environmental agency tries to minimize the overall pollution, etc.

In general, as the above examples show, we can have both maximization and minimization problems. However, the problem of maximizing an objective function

$f(x)$ is equivalent to minimizing the function $g(x) \stackrel{\text{def}}{=} -f(x)$. Thus, all optimization problems can be easily reduced to minimizations. So, without losing generality, mathematicians usually only talk about minimization problems.

Need for convex optimization. In general, optimization is NP-hard (see, e.g., [9, 14]), meaning that unless P=NP (which most computer scientists believe to be impossible), no feasible algorithm can solve all optimization problems. There is an important class of optimization problems for which optimization is feasible: the class of all *convex* optimization problems (see, e.g., [10, 11, 14]), in which the minimized functions $f(x)$ is convex, i.e., satisfies the condition

$$f(\alpha \cdot x + (1-\alpha) \cdot x') \le \alpha \cdot f(x) + (1-\alpha) \cdot f(x') \tag{5}$$

for all x, x', and for all $\alpha \in [0,1]$.

Moreover, it has been proven that convex functions are, in some reasonable sense, the largest class of functions for which optimization is feasible: once we add some non-convex functions to this problem, the optimization problem becomes NP-hard; see [6].

This result will underlie our two explanations.

3 First Convexity-Related Explanation

How is all this related to neural networks. One of the main applications of neural networks is to make decisions. For this purpose, we need to train the neural network to predict, for each possible action, the consequences of this action. In other words, we want, given the parameters x that characterize the possible decision, to compute the value $f(x)$ of the objective function that characterizes this decision. For the simplest neural networks, this means that we approximate the original function $f(x_1,\ldots,x_n)$ by a linear combination of the output of non-liner neurons:

$$f(x_1,\ldots,x_n) = \sum_{k=1}^{K} W_k \cdot s\left(\sum_{i=1}^{n} w_{ki} \cdot x_i - w_{k0}\right) - W_0. \tag{6}$$

For multi-layer neural networks, the corresponding expression is more complicated.

Towards resulting natural requirements on the activation function. Once we train the neural network to compute the value of the objective function, a natural next step is to find the alternative x that minimizes this objective function. Since, as we have mentioned, optimization is only feasible for convex objective functions, it makes sense to make sure that the expression (6) – and a similar expression for multi-layer neural networks – preserve convexity as much as possible.

In other words, if the actual activation function is convex, we want to make sure that this convexity is, in some reasonable sense, preserved in an approximating expressions like (6).

First requirement. The above idea means, in particular, that for the simplest case when one neuron is sufficient, the activation function $s(x)$ itself must be convex.

Comment. The rectified linear activation function (1) itself is convex, so it satisfies this requirement.

On the other hand, there are many other convex functions, so this requirement does not uniquely determine the rectified linear function. For this unique determination, we need to come up with additional requirement(s).

Second requirement. It is known that if functions $f_1(x), \ldots, f_n(x)$ are convex, then their convex combination

$$f(x) = w_1 \cdot f_1(x) + \ldots + w_K \cdot f_K(x), \tag{7}$$

where $w_k \geq 0$ and $\sum\limits_{k=1}^{K} w_k = 1$, is also convex. Moreover, any linear combination with non-negative coefficients is convex, even when the sum of these coefficients is different from 1. On the other hand, if we allow even one of the coefficients to be negative, then we already get non-convex functions. So, the only way to make sure that a linear combination of convex functions is convex is to make sure that all the coefficients w_k are non-negative.

It is therefore reasonable to require that every convex function $f(x)$ – at least every convex function of one variable – be representable as a linear combination of activation functions with non-negative coefficients. This is our second requirement.

Let us analyze what are the activation functions that satisfy this requirement.

Let us recall the usual calculus-based characteristics of convexity. It is known that a differentiable function $f(x)$ is convex if and only if its second derivative $f''(x)$ is everywhere non-negative $f''(x) \geq 0$. Not all convex functions are everywhere differentiable – e.g., the rectified linear activation function $s_0(x)$ is not differentiable at the point $x = 0$. However, for such function, we can consider, as derivatives, *generalized functions* (also known as *Schwartz distributions*), which are limits of usual functions; see, e.g., [3, 5]. The most well-known generalized function is a *delta-function* $\delta(x)$ which is equal to 0 for all $x \neq 0$ and which tends to ∞ at $x = 0$; such functions are used in physics to describe, e.g., point-wise particles and objects; see, e.g., [1, 12]. In particular, the derivative $s_0'(x)$ of the rectified linear function is equal to 0 for $x \leq 0$ an to 1 for $x > 0$, and the second derivative is exactly the delta-function.

For a linear combination of functions (7), its second derivative is equal to the linear combination of its second derivatives, with exactly the same coefficients w_k:

$$f''(x) = w_1 \cdot f_1''(x) + \ldots + w_K \cdot f_K''(x). \tag{8}$$

So, in terms of the second derivatives, the above second requirement means that every non-negative (generalized) function can be represented as a linear combination of the functions corresponding to second derivative of the activation function $s(x)$ – and of its shifted and scaled versions $s(a_0 + a_1 \cdot x)$.

Now we can prove that only rectified linear activation function satisfies both our requirements. If the second derivative $s''(x)$ of an activation function $s(x)$ differs from 0 for at least two different values $x \neq x'$, then this property remains true for any convex combination of shifted and scaled versions of this activation function. Thus, this way, we will never get a convex function for which the second derivative is non-zero only for one value x – e.g., the rectified linear function $s_0(x)$.

On the other hand, if we select the rectified linear function $s_0(x)$ as an activation function, then we have $s_0''(x) = \delta(x)$. In this case, any non-negative function $f''(x)$ can be represented as a linear combination of shifted versions of $s_0''(x)$: indeed,

$$f''(x) = \int f''(y) \cdot \delta(x-y)\,dy = \int f''(y) \cdot s_0''(x-y)\,dy, \tag{9}$$

and thus, the function $f(x)$ can be represented as a similar linear combination of the shifted versions of $s_0(x)$ – plus possibly some linear terms:

$$f(x) = b_0 + b_1 \cdot x + \int f''(y) \cdot s_0(x-y)\,dy. \tag{10}$$

In general, our second requirement is satisfied by any convex function for which the second derivative is different from 0 only for one value $x = x_0$. This second derivative can therefore be described as

$$s''(x) = c \cdot \delta(x - x_0), \tag{11}$$

for some $c > 0$. Integrating twice the equality (11), we conclude that

$$s(x) = b_0 + b_1 \cdot x + c \cdot s_0(x - x_0), \tag{12}$$

for some values s_0 and s_1. One can check that this is exactly the expression (2-4), i.e., that indeed, the above two natural convexity-related requirements naturally lead to the rectified linear activation functions.

4 Second Convexity-Related Explanation

Let us consider a more general setting. Out of the above two requirements, the first one looks more convincing, the second one is somewhat less convincing. Let us therefore consider a more general setting, when we still postulate the first requirement – i.e., we still consider only convex activation functions – but instead of postulating the second requirement, we want to find the activation function which is the best in some sense, i.e., for which the corresponding objective functional $F(s)$

describing the relative qualities of different convex activation functions $s(x)$ – attains its smallest possible value.

What calculus tells us. In general, a maximum or minimum of a function on a multi-D domain is attained either inside this domain – in which case it is a stationary point of this function – or on its boundary. When the domain is relatively small, the probability that a global stationary point is inside this domain is very small, so it is reasonable to assume that the minimum is attained on the boundary.

This general conclusion can be applied to our case when we optimize a functional $F(s)$ on the domain of all convex functions s. Indeed, most functions are not convex. So, in the space of all possible functions, the domain of all convex functions is indeed small.

Similarly, if the domain's boundary contains a flat face-type part – as when the domain is a polytope – then it is reasonable to assume that the minimum is attained not in the interior of this face, but on its boundary. If this boundary also contains a flat part – as in the case of a polytope where the boundary of a face consists of edges – we can similarly conclude that the minimum is most probably attained at the boundary of this part – e.g., for a 3-D polytope, at one of the vertices.

In general, we can conclude that the minimum is most probably attained at one of the *extreme points* of the original domain – i.e., a point that cannot be represented as a convex combination of other points from this domain.

Comment. For a precise mathematical description of this idea, see [13].

What this implies for optimal activation functions. We want to select an activation function. In this case, the domain is the set of all convex functions. What are the extreme elements of this domain?

We have already shown that any convex function $s(x)$ whose second derivative differs from 0 at least 2 different points can be represented as a convex combination of other convex functions – namely, shifted rectified linear functions. Hence, such functions $s(x)$ are not extreme elements of our domain. Thus, the only extreme elements of this domain are convex functions whose second derivative differs from 0 only at one point – which are, as we have shown, exactly rectified linear functions.

Since, with high probability, only extreme elements can be optimal, we conclude that with high probability, only rectified linear functions can be optimal – no matter what optimality criterion we used. Thus, we have indeed provided a second theoretical justification for the success of rectified linear activation functions.

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.

References

1. R. Feynman, R. Leighton, and M. Sands, *The Feynman Lectures on Physics*, Addison Wesley, Boston, Massachusetts, 2005.
2. O. Fuentes, J. Parra, E. Anthony, and V. Kreinovich, "Why rectified linear neurons are efficient: a possible theoretical explanations", In: O. Kosheleva, S. Shary, G. Xiang, and R. Zapatrin (eds.), *Beyond Traditional Probabilistic Data Processing Techniques: Interval, Fuzzy, etc. Methods and Their Applications*, Springer, Cham, Switzerland, 2020, pp. 603–613.
3. I. M. Gel'fand and G. E. Shilov, *Generalized functions*, Academic Press, New York, 1966–1969.
4. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, Massachusetts, 2016.
5. G. Grubb, *Distributions and Operators*, Springer, New York, 2009.
6. R. B. Kearfott and V. Kreinovich, "Beyond convex? global optimization is feasible only for convex objective functions: a theorem", *Journal of Global Optimization*, 2005, Vol. 33, No. 4, pp. 617–624.
7. V. Kreinovich and O. Kosheleva, "Deep learning (partly) demystified", *Proceedings of the 2020 4th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence ISMSI'2020*, Thimpu, Bhutan, April 18–19, 2020.
8. V. Kreinovich and O. Kosheleva, "Optimization under uncertainty explains empirical success of deep learning heuristics", In: P. Pardalos, V. Rasskazova, and M. N. Vrahatis (eds.), *Black Box Optimization, Machine Learning and No-Free Lunch Theorems*, Springer, Cham, Switzerland, 2021, pp. 195–220.
9. V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht, 1998.
10. G. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 2006.
11. R. T. Rockafeller, *Convex Analysis*, Princeton University Press, Princeton, New Jersey, 1997.
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. B. S. Tsirel'son, "A geometrical approach to maximum likelihood estimation for infinite-dimensional Gaussian location. I", *Theory of Probability and its Applications*, 1982, Vol. 27, pp. 411–418.
14. S. A. Vavasis, *Nonlinear Optimization: Complexity Issues*, Oxford University Press, New York, 1991.