11-1-2021

Why rectified linear neurons: a possible interval-based explanation

Jonathan 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-21-101

Why rectified linear neurons: a possible interval-based explanation*

Jonathan Contreras, Martine Ceberio, and Vladik Kreinovich
Department of Computer Science, University of Texas
at El Paso, El Paso, TX 79968, USA
jmcontreras2@utep.edu,mceberio@utep.edu,vladik@utep.edu

Abstract

At present, the most efficient machine learning techniques are deep neural networks. In these networks, a signal repeatedly undergoes two types of transformations: linear combination of inputs, and a non-linear transformation of each value $v \mapsto s(v)$. Empirically, the function $s(v) = \max(v, 0)$ – known as the *rectified linear* function – works the best. There are some partial explanations for this empirical success; however, none of these explanations is fully convincing. In this paper, we analyze this why-question from the viewpoint of uncertainty propagation. We show that reasonable uncertainty-related arguments lead to another possible explanation of why rectified linear functions are so efficient.

Keywords: Interval uncertainty, neural networks, rectified linear neurons
AMS subject classifications: 68T07, 65G20, 65G30, 65G40

1 What are rectified linear neurons

At present, the most efficient machine learning techniques are deep neural networks; see, e.g., [1]. In general, in a neural network, a signal repeatedly undergoes two types of transformations:

- linear combination of inputs, and
- a non-linear transformation of each value $v \mapsto s(v)$.

The corresponding nonlinear function $s(v)$ is called an *activation function.*

In deep neural networks, most nonlinear layers use the function $s(v) = \max(0, v)$. This function is called the *rectified linear (ReLU) activation function.*

2 Why rectified linear neurons?

Empirically, rectified linear activation functions work the best. There are some partial explanations for this empirical success; see, e.g., [2]. However, none of these explanations is fully convincing. So yet another explanation is always welcome.

*Submitted: November 30, 2021; Revised: ?; Accepted: ?.

In this paper, we analyze this why-question from the viewpoint of uncertainty propagation. We show that some reasonable uncertainty-related arguments indeed lead to a possible (partial) explanation.

3 It does not matter which 2-piece-wise linear activation function we use

A 2-piece-wise linear activation function means that we have two different linear functions $s_1(x) = a_1 \cdot x + b_1$ and $s_2(x) = a_2 \cdot x + b_2$, and we have a nonlinear continuous function $s(x)$ for which, for every x:

- either $s(x) = s_1(x)$
- or $s(x) = s_2(x)$.

We cannot have $a_1 = a_2$ – then we never have $s_1(x) = s_2(x)$, so $s(x)$ cannot switch from one to another. Thus, $a_1 \neq a_2$.

Without losing generality, we can assume that $a_1 < a_2$. The only point where $s(x)$ can switch is when $s_1(x_0) = s_2(x_0)$, i.e., when:

$$y \stackrel{\text{def}}{=} a_1 \cdot x_0 + b_1 = a_2 \cdot x_0 + b_2.$$

So:

$$x_0 = \frac{b_1 - b_2}{a_2 - a_1}.$$

Then, $s_1(x) = y + a_1 \cdot (x - x_0)$ and $s_2(x) = y + a_2 \cdot (x - x_0)$. So, $s_1(x + x_0) = y + a_1 \cdot x$ and $s_2(x + x_0) = y + a_2 \cdot x$.

If $s(x) = s_1(x)$ for $x < x_0$ and $s(x) = s_2(x)$ for $x > x_0$, then

$$s(x + x_0) - (y + a_1 \cdot x) = (a_2 - a_1) \cdot \max(x, 0), \text{ so}$$

$$\max(x, 0) = \frac{1}{a_2 - a_1} \cdot s(x + x_0) - \frac{y}{a_2 - a_1} - \frac{a_1}{a_2 - a_1} \cdot x.$$

So, by using a single neuron and linear transformations, we can get ReLU.

Similarly, by using ReLU, we can get this neuron as

$$s(x) = a_1 \cdot x + b_1 + (a_2 - a_1) \cdot \max(0, x - x_0).$$

Similar equivalence occurs if $s(x) = s_2(x)$ for $x < x_0$ and $s(x) = s_1(x)$ for $x > x_0$.

4 Need to take interval uncertainty into account

The activation function transforms the input v into the output $y = s(v)$. The input v comes:

- either directly from measurements,
- or from processing measurement results.

Measurements are never absolutely accurate. The measurement result $\widetilde{v}$ is, in general, different from the actual (unknown) value of the quantity v. In many practical situations, all we know about the measurement error $\Delta v \stackrel{\text{def}}{=} \widetilde{v} - v$ is the upper bound Δ on its absolute value: $|\widetilde{v} - v| \le \Delta$; see, e.g., [3]. In this case, possible values of v form an interval $[\widetilde{v} - \Delta, \widetilde{v} + \Delta]$.

5 First natural requirement

A first natural requirement is that the output y should not be too much affected by inaccuracy with which we know the input. Ideally, this inaccuracy should not increase after data processing, i.e., we should have

$$|s(\widetilde{v}) - s(v)| \le |\widetilde{v} - v|.$$

In mathematical terms, this means that the function $s(v)$ should be 1-Lipschitz. So its derivative (or generalized derivative) should be limited by 1: $|s'(v)| \le 1$.

6 Second natural requirement: first try

On the other hand, we do not want to lose information about the signal. So we must be able to reconstruct the input signal from the output as accurately as possible. This idea can be naturally described as $|\widetilde{v} - v| \le |s(\widetilde{v}) - s(v)|$.

Together with the first requirement, this means that $|\widetilde{v} - v| = |s(\widetilde{v}) - s(v)|$; see proof below. Taking into account that we want to uniquely reconstruct v from $s(v)$, this implies that either $s(v) = v + c$ or $s(v) = -v + c$. However, we wanted the function $s(v)$ to be nonlinear, since otherwise we will only be able to represent linear dependencies.

7 Proof that the first try does not work

Indeed, we have $|s(1) - s(0)| = 1$. This means that we have either $s(1) - s(0) = 1$ or $s(1) - s(0) = -1$. Let us show that in the first case, we have $s(v) - s(0) = v$ for all v.

Indeed, we have $s(v) - s(1) = \pm(v - 1)$ and $s(v) - s(0) = \pm v$. Let us show, by contradiction, that we cannot have $s(v) - s(0) = -v \ne v$. Indeed, then $s(v) - s(1) = (s(v) - s(0)) - (s(1) - s(0)) = -v - 1$. On the other hand, $s(v) - s(1) = \pm(v - 1)$, so $-v - 1 = \pm(v - 1)$.

- If $-v - 1 = v - 1$, then $-v = v$ and $v = 0$. In this case, $-v = v$.
- If $-v - 1 = -v + 1$, then we get $-1 = 1$ – a contradiction.

So, indeed, $s(v) - s(0) = v$, thus $s(v) = v + c$, where $c \stackrel{\text{def}}{=} s(0)$.

Similarly, we can prove that if $s(1) - s(0) = -1$, then $s(v) = -v + c$.

8 Second natural requirement made realistic

We showed that we cannot accurately reconstruct the input v from $s(v)$. So, a natural idea is to use *two* activation functions $s_1(v)$ and $s_2(v)$ so that for each v, we can accurately reconstruct the signal from at least one of the two outputs $s_i(v)$.

9 What we can conclude

A natural conclusion is that for (almost) all values v, we must have:

- either $|s_1'(v)| = 1$
- or $|s_2'(v)| = 1$.

In other words, the real line – the set of all possible values v – is divided into two subsets:

- on one of them $s_1(v) = \pm v + c_1$,
- on another one $s_2(v) = \pm v + c_2$.

10 Third natural requitement

Many real-life dependencies are linear. The simplest linear function is $f(v) = v$.

It is desirable to require that $f(v) = v$ can be represented as a linear combination of the two activation functions, i.e., that:

$$v = c_0 + c_1 \cdot s_1(v) + c_2 \cdot s_2(v).$$

11 What we can now conclude

- For values v for which $s_1(v) = \pm v + c_1$, we conclude that
$$s_2(v) = c_2^{-1} \cdot (v - c_0 - c_1 \cdot s_1(v)).$$
Thus, for these v, the function $s_2(v)$ is linear.
- Similarly, for remaining values v – for which $s_2(v) = \pm v + c_2$ – we can conclude that the function $s_1(v)$ is linear.

Thus, both activation functions $s_1(v)$ and $s_2(v)$ are piecewise linear.

This exactly what we wanted to explain.

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).

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.

The authors are thankful to all the participants of the 19th International Symposium on Scientific Computing, Computer Arithmetic, and Verified Numerical Computation SCAN'2021 (Szeged, Hungary, September 13–15, 2021) for valuable suggestions.

References

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

[2] V. Kreinovich, O. Kosheleva: Optimization under uncertainty explains empirical success of deep learning heuristics, In: P. Pardalos, V. Rasskazova, M. N. Vrahatis (eds.): *Black Box Optimization, Machine Learning and No-Free Lunch Theorems*, Springer, Cham, Switzerland, 2021, pp. 195–220.

[3] S. G. Rabinovich, *Measurement Errors and Uncertainties: Theory and Practice*, Springer, New York, 2005.