Why rectified linear neurons: a possible interval-based explanation

Jonatan Contreras, Martine Ceberio, and Vladik Kreinovich

University of Texas at El Paso, USA

jmcontreras2@utep.edu, mceberio@utep.edu, vladik@utep.edu

Keywords: neural networks, rectified linear neurons, interval uncertainty

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, and a non-linear transformation of each value $v \to s(v)$. The corresponding nonlinear function $s(v)$ is called an *activation function*. In deep neural networks, most nonlinear layers use the function $s(v) = \max(0, v)$ which is called the *rectified linear (ReLU) activation function*.

Comment. Taking into account that we also have linear layers, what can be represented by the ReLU function can also be represented if we use any piece-wise linear activation function.

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]), but none of them is fully convincing, so yet another explanation is always welcome.

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

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: $|\widetilde{v} - v| \le \Delta$. In this case, possible values of v form an interval $[\widetilde{v} - \Delta, \widetilde{v} + \Delta]$.

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 increases 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: $|s'(v)| \le 1$.

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 $|\widetilde{v} - v| = |s(\widetilde{v}) - s(v)|$. Taking into account that we want to uniquely reconstruct v from $s(v)$, this implies that either $s(v) = v$ or $s(v) = -v$. However, we wanted the function $s(v)$ to be nonlinear, since otherwise we will only be able to represent linear dependencies.

Second natural requirement made realistic. Since we cannot accurately reconstruct the input v from $s(v)$, 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 $s_i(v)$.

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 $s_1(v) = \pm v$, on another one $s_2(v) = \pm v$.

Third natural requitement. Since many real-life dependencies are linear, it is desirable to require that a linear function – e.g., the function $f(v) = v$ – can be represented as a linear combination of the two activation functions, i.e., that $v = c_0 + c_1 \cdot s_1(v) + c_2 \cdot s_2(v)$.

What we can now conclude. For values v for which $s_1(v) = \pm v$, we conclude that $s_2(v) = c_2^{-1} \cdot (v - c_0 - c_1 \cdot s_1(v))$ is linear. Similarly, for remaining values v – for which $s_2(v) = \pm v$ – 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 – which is exactly what we wanted to explain.

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.