SIGMOID NEURONS ARE THE SAFEST AGAINST ADDITIVE ERRORS

Ongard Sirisaengtaksin (University of Houtson - Downtown)
Vladik Kreinovich (University of Texas at El Paso)
Hung T. Nguyen (New Mexico State University)

Abstract. *In this paper, we provide one more explanation of why sigmoid $1/(1 + \exp(-x))$ is successfully used in neural data processing.*

We are looking for a non-linear transformation device $x \rightarrow s(x)$ with the following property: if we have not corrected some systematic error in the input, then we will still be able to correct it in the output relatively easily. We show that a natural formalization of "easy" leads to a finite-dimensional transformation group that contains all linear functions. As a result, we get an explaanation of the standard sigmoid.

Formulation of the problem. The main application of neural networks is to process data. These data come from the measurement results. Measurement are never 100% accurate; the result $\tilde{x}$ of measuring a physical quantity x, generally speaking, differs from the actual value of x. The resulting error $\Delta x = \tilde{x} - x$ can be represented as the sum of two components (see, e.g., [Rabinovich 1993]):

- the *systematic* component, that can be defined as the *mathematical expectation* of the error, and
- the *random* component, that is defined as a difference between the error and its systematic component.

By definition, the random component cannot be predicted and therefore, it stays. However, a systematic component can and should be corrected.

In principle, this correction can be done as follows:

- First, we analyze the measurement results, compare them with the results of measuring the same quantities by more accurate measuring devices, and by comparison, determine the bias b of this particular measuring device.
- Then, we correct the values measured by this device: instead of the measurement result $\tilde{x}$, we take $\tilde{x} - b$.

This is the traditional method of correcting systematic error. However, when we apply this method to neural networks, we run into the following problem: Neural networks are intended to *learn*, not simply to process, and learning is slow. So, to speed up the data processing, we would like to start processing as soon as we start getting measurement results. In other words, we would like to start processing *before* we know the value of the additive bias (= systematic error). Since we did not take this error into consideration, the results of this processing are less accurate than we can potentially have. So, after we have determined the systematic error, we would like to *correct* the results of preliminary data processing (that was done without correcting the inputs). And here comes the problem: neurons are *non-linear* devices, so, they perform *non-linear* data transformation $x \rightarrow s(x)$ (where $s(x)$ denotes the activation function).

Let us illustrate this problem on the example of the case when the neuron processes only one input, and the input weight of that input is exactly equal to 1. In this case, the problem can be reformulated as follows:

- Initially, we have computed the value $s(\tilde{x})$.
- Now, we want to compute $s(\tilde{x} - b)$ for some given b.

For a generic non-linear function, knowing $s(\tilde{x})$ is of no big help in computing $s(\tilde{x} - b)$, so we could as well forget about the results of the preliminary computations, and start all the computations

anew. But if we do that, i.e., if we do not use the results of these preliminary computations, then we do not gain anything by performing them, and hence, we do not achieve any speed up.

The only case when we *do* achieve some speed up is when the activation function is chosen in such a way that from $s(\tilde{x})$, we can easily compute $s(\tilde{x} - b)$.

For a general case, when the neuron takes several inputs $\tilde{x}_i$ with different weights w_i, we arrive at the similar problem:

- In the preliminary data processing, we have applied the activation function s to the value $\tilde{x} = w_0 + w_1\tilde{x}_1 + ... + w_n\tilde{x}_n$.
- Now, since we know the values $b_1, ..., b_n$ of the systematic errors, we are interested in the result of applying s to the corrected value $x = w_0 + w_1(\tilde{x}_1 - b_1) + ... + (\tilde{x}_n - b_n)$.

This new situation may look more complicated than the previous one. However, as one can see, $x = \tilde{x} - b$, where $d = w_1b_1 + ... + w_nb_n$. Therefore, we end up with exactly the same problem: *to find such an activation function $s(x)$ that from $s(\tilde{x})$, we will be able to easily compute $s(\tilde{x} - b)$.*

In this paper, we will describe such functions. But before we do that, we must somehow formalize the notion of "easy".

Formalization of "easy". To formalize this notion, let us write down natural properties of "easiness". It turns out that there properties will be sufficient to determine the class of "easy" transformations E uniquely.

These properties are as follows:
- If f and g are easy transformations, then their composition should also be easy. In mathematical terms, this means that the class E must be *closed under composition*.
- Suppose that we accidentally applied the wrong transformation. Then, it would be nice to be able to easily undo it. If for every b, going from $s(\tilde{x})$ to $s(x)$, where $x = \tilde{x} - b$, is easy, then undoing this transformation is also easy, because this undoing will mean going from $s(x)$ to $s(x - (-b))$ (i.e., the transformation of exactly the same type). So, we can assume that for every easy transformation, its *inverse* is also easy. In mathematical terms, this means that the class E is *closed under inversion*.

In mathematical terms, the fact that we have these two conditions (closeness w.r.t. compositions and inversions) means that these transformations form a *group*. So, the family of all easy transformations is a transformation group.

- Linear transformations are definitely easy, so, we will assume that an arbitrary linear transformation $f : R \to R$ belongs to the class E.
- Finally, "easy" means easily computable. So, there must be a simple program with a few parameters such that by choosing different values of these parameters, we will have different easy transformations. In mathematical terms, this means that this family E is *finite-dimensional*. A natural formalization of the notion of a finite-dimensional transformation group is a *connected Lie group* (see, e.g., [Chevalley 1946]).

So, we arrive at the following definition:

Definition 1.
- Let E denote a connected Lie group of transformations of R that contains all linear transformations. Elements of the set E will be called *easy transformations*.
- We say that a non-linear function $s : R \to R$ is *easily correctable for additive errors*, if for every real number b, there exists an easy transformation s such that for every x, $s(x - b) = e(s(x))$.

Definition 2. By a *(logistic)* sigmoid function, we mean a function $s_0(y) = 1/(1 + \exp(-y))$.

MAIN RESULT. *If a function $s(x)$ is easily correctable for additive errors, then either $s(x) = a + bs_0(Ky + l)$ for some a, b, K and l, or $s(x) = a + b\exp(Kx)$ for some a, b, and K.*

Comments.

1. If we require (as it is usually done in neural networks) that the range of the function s is bounded, then we are left with only standard sigmoid.

2. Both functions s_0 and exp are easily correctable.

3. Cowan [Cowan 1967] has chosen the logistic function, $s_0(x) = 1/(1 + \exp(-x))$, because it leads to a good approximation of the behavior of real (i.e. biological) neurons. The properties of neural networks with different f were studied by Grossberg (see, e.g., [Grossberg 1988]) who showed that the logistic function has several nice properties useful for learning and is therefore an adequate choice.

Proof. The idea of this proof is as follows: first, we prove that all easy transformations are fractionally-linear functions (in part 1), and then, we conclude that the correctable function s satisfies some functional equation, whose solutions are known (in part 2).

1. We know that E is a finite-dimensional Lie group of transformations of the set of real numbers R onto itself that contains all linear transformations. Norbert Wiener asked to classify such groups for an n-dimensional space with arbitrary n, and this classification was obtained in [Guillemin Sternberg 1964] and [Singer Sternberg 1965]. In our case (when $n = 1$) the only possible groups are the group of all linear transformations and the group of all fractionally-linear transformations $x \to (ax + b)/(cx + d)$. In both cases the group consists only of fractionally linear transformations (the simplified proof for the 1-dimensional case is given in [Kreinovich 1987]; for other applications of this result see [Kreinovich Kumar 1990, 1991], [Kreinovich Corbin 1991], [Kreinovich Quintana 1991a, 1991b]).

2. Since s is easily correctable, and all easy transformations are fractionally linear, we can conclude that for every a, $s(y+a) = (A + Bs(y))/(C + Ds(y))$ for some A, B, C and D. So we arrive at a functional equation for s. Let us reduce this equation to a one with a known solution. For that purpose, let us use the fact that fractionally linear transformations are projective transformations of a line, and for such transformations the cross ratio is preserved [Aczel 1966, Section 2.3], i.e., if $g(y) = (A + Bs(y))/(C + Ds(y))$, then

$$\frac{g(y_1) - g(y_3)}{g(y_2) - g(y_3)} \frac{g(y_2) - g(y_4)}{g(y_1) - g(y_4)} = \frac{s(y_1) - s(y_3)}{s(y_2) - s(y_3)} \frac{s(y_2) - s(y_4)}{s(y_1) - s(y_4)}$$

for all y_i. In our case this is true for $g(y) = s(y + a)$, therefore for all a the following equality is true:

$$\frac{s(y_1 + a) - s(y_3 + a)}{s(y_2 + a) - s(y_3 + a)} \frac{s(y_2 + a) - s(y_4 + a)}{s(y_1 + a) - s(y_4 + a)} = \frac{s(y_1) - s(y_3)}{s(y_2) - s(y_3)} \frac{s(y_2) - s(y_4)}{s(y_1) - s(y_4)}.$$

The most general continuous solutions of this functional equation are given by Theorem 2.3.2 from [Aczel 1966]: either s is fractionally linear, or $s(y) = (a + b\tan(ky))/(c + d\tan(ky))$ for some a, b, c, d, or $s(y) = (a + b\tanh(ky))/(c + d\tanh(ky))$, where $\tanh(z) = \sinh(z)/\cosh(z)$, $\sinh(z) = (\exp(z) - \exp(-z))/2)$ and $\cosh(z) = (\exp(z) + \exp(-z))/2)$.

If $s(y)$ is fractionally linear $s(y) = (a + by)/(c + dy)$ and $d \neq 0$, then the denominator is equal to zero for $y = -c/d$. The only way for the function to be defined for this y is that the numerator should also be zero, i.e., $a + by = a + b(-c/d) = 0$. But in this case $a = b(c/d)$, therefore $a + by = b(c/d + y) = (b/d)(c + dy)$, and the fraction $s(y)$ is always equal to a constant b/d. But we assumed that s is a non-linear, so, it cannot be a constant. Hence, $d = 0$. In this case, s is linear, which contradicts to our assumption. So, s cannot be fractionally linear.

Let us prove that the expressions with tangent are impossible. Indeed, the denominator must be not identically equal to zero, therefore either $c \neq 0$, or $d \neq 0$. If $d \neq 0$, then for $ky = \arctan(-c/d)$

3

we have $\tan(ky) = -c/d$, and the denominator is equal to zero. As in the linear case we can then conclude that in this case f is constant, and that contradicts to our assumption that it is not. So $d = 0$ and $s(y) = (a/d) + (b/d)\tan(ky)$. Hence either $b = 0$ and $s = const$, or $b \neq 0$, and s is not defined, when $\tan(ky) = \infty$, i.e., when $ky = \pi/2$ and $y = \pi/(2k)$. So expressions with tangent are really impossible.

Let us now consider the case of hyperbolic tangent. If $k = 0$, then s is constant, which is impossible. So $k \neq 0$. If $k < 0$, then we can take $\bar{k} = -k$ and use the fact that tanh is an odd function, so $\tanh(ky) = -\tanh(\bar{k}y)$. Therefore, in the following we can assume that $k > 0$. Multiplying both the denominator and the numerator by $\cosh(z)$, we conclude that $s(y) = (a\cosh(ky) + b\sinh(ky))/(c\cosh(ky) + d\sinh(ky))$. We then substitute the expressions for sinh and cosh in terms of exp, and conclude that $s(y) = (A\exp(ky) + B\exp(-ky))/(C\exp(ky) + D\exp(-ky))$ for some A, B, C, D. Multiplying both denominator and numerator by $\exp(-ky)$, we arrive at $f(y) = (A + B\exp(-2ky))/(C + D\exp(-2ky))$. If $D = 0$, then we get a linear transformation of the exponential function. If $C = 0$, then $f(y) = (B/D) + (A/D)\exp(2ky)$, which is also a linear transformation of the exponential function. Let us now consider the case, when both C and D are different from 0.

If C and D have different signs, then for $\exp(2ky) = -D/C$ the denominator equals to zero, and so, just like in the tangent case, we conclude that f is either identically constant, or not defined in this point $y = \ln(-D/C)/(2k)$. If C and D have the same signs, then for $l = -\ln(D/C)$ we have $C + D\exp(-2ky) = C(1 + (D/C)\exp(-2ky)) = C(1 + \exp(-(2ky + l))$. If we substitute $\exp(-2ky) = \exp(-(2ky + l))\exp(l) = (C/D)\exp(-(2ky + l))$ into the numerator, we get

$$A + \frac{BC}{D}\exp(-(2ky + l)),$$

and therefore $s(y) = (A + (BC/D)\exp(-(2ky + l)))/(C(1 + \exp(-(2ky + l))))$. One can check (by substituting the expression of the logistic function s_0 in terms of exp) that this expression is equal to $(A/C) + (B/D - A/C)s_0(2ky + l)$. So we get the desired expression for $K = 2k$. Q.E.D.

References

Aczel, J. *Lectures on functional equations and their applications.* Academic Press, NY-London, 1966.

Chevalley, C. *Theory of Lie groups*, Princeton University Press, Princeton, NJ, 1946.

Cowan, J. D. *A mathematical theory of central nervous activity.* Ph. D. Dissertation, Univ. London, 1967.

Guillemin, V. M. and S. Sternberg. *An algebraic model of transitive differential geometry*, Bulletin of American Mathematical Society, 1964, Vol. 70, No. 1, pp. 16–47.

Grossberg, S. *Nonlinear neural networks: Principles, mechanisms and architectures.* Neural Networks, 1988, Vol. 1, pp. 17–61.

Kosko, B. *Neural networks and fuzzy systems.* Prentice Hall, Englewood Cliffs, NJ, 1992.

Kreinovich, V. A mathematical supplement to the paper: I. N. Krotkov, V. Kreinovich and V. D. Mazin. *A general formula for the measurement transformations, allowing the numerical methods of analyzing the measuring and computational systems*, Measurement Techniques, 1987, No. 10, pp. 8–10.

Kreinovich, V. *Group-theoretic approach to intractable problems.* Lecture Notes in Computer Science, Springer-Verlag, Berlin, Vol. 417, 1990, pp. 112–121.

Kreinovich, V. *Arbitrary nonlinearity is sufficient to present all functions by neural networks: a theorem.* Neural Networks, 1991, vol. 4, pp. 381–383.

Kreinovich, V. and J. Corbin, *Dynamic tuning of communication network parameters: why fractionally linear formulas work well.* University of Texas at El Paso, Computer Science Department, Technical Report UTEP-CS-91-4, 1991.

Kreinovich, V. and S. Kumar, *Optimal choice of &- and ∨- operations for expert values.* Proceedings of the 3rd University of New Brunswick Artificial Intelligence Workshop, Fredericton, New Brunswick, Canada, 1990, pp. 169–178.

Kreinovich, V. and S. Kumar, *How to help intelligent systems with different uncertainty representations communicate with each other.* Cybernetics and Systems: International Journal, 1991, vol. 22, No. 2, pp. 217–222.

[Kreinovich Quintana 1991a] Kreinovich, V. and C. Quintana. *How does new evidence change our estimates of probabilities: Carnap's formula revisited.* Submitted to Cybernetics and Systems.

[Kreinovich Quintana 1991b] Kreinovich, V. and C. Quintana. *Neural networks: what nonlinearity to choose.* Proceedings of the 4th University of New Brunswick Artificial Intelligence Workshop, Fredericton, New Brunswick, 1991, pp. 627–637.

Rabinovich, S. *Measurement errors: theory and practice*, American Institute of Physics, N.Y., 1993.

Singer, I.M. and S. Sternberg. *Infinite groups of Lie and Cartan*, Part 1, Journal d'Analyse Mathematique, 1965, Vol. XV, pp. 1–113.

Wiener, N. *Cybernetics, or Control and Communication in the animal and the machine*, MIT Press, Cambridge MA, 1962.