Artificial Neural Network Activation Functions in Exact Analytical Form (Heaviside, ReLU, PReLU, ELU, SELU, ELiSH)

Rami Alkhatib¹

¹ Mechanical and Mechatronics Engineering Department, Rafik Hariri University, Lebanon, Mechref, Chouf2010

Abstract—Activation functions are fundamental elements in artificial neural networks. The mathematical formulation of some activation functions (e.g. Heaviside function and Rectified Linear Unit function) are not expressed in an explicit closed form. This made them numerically unstable and computationally complex during estimation. This paper introduces a novel explicit analytic form for those activation functions. The proposed mathematical equations match exactly the original definition of the studied activation function. The proposed equations can be adapted better in optimization, forward and backward propagation algorithm employed in an artificial neural network.

Index Terms—Activation Functions, Analytic Function, Machine Learning, Neural Networks

I. INTRODUCTION

Activation functions are bio-inspired mathematical equations to represent the firing action potential in a neuron or node. They are a transformation and nonlinear mapping of inputs from one layer into the next layer. Moreover, the set of interconnected neurons in an intricate manner forms diverse kinds of neural networks. Remarkably, decisions are made at the non-linearity of those functions, which exist at zero most of the time. This property of activation functions enabled neural networks to learn from complex and higher-order data to provide precise predictions and classifications. Moreover, their nonlinearity affects the convergence of neural networks and plays a vital role in specifying the convergence speed and computational efficiency. Models of neural networks include, but are not limited to, Feedforward Neural Network [1], Radial basis function Neural Network [2], Kohonen Self Organizing Neural Network [3], Recurrent Neural Network(RNN) – Long Short Term Memory [4], Convolutional [5] and Modular Neural Network.

Accordingly, activation functions control the activation of neurons and thus the output of a neural network to perform complicated tasks like handwritten character recognition [6], Image Processing Applications [7], Application for Stock Market Prediction [8], Signature Verification Application [9], cancer detection [10], Thunderstorms Forecasting and Weather prediction [11], Predicting Driving Fatigue [12], Self-Driving Car Steering Angle Prediction [13], Navigating Self-Driving car [14], Online Advertising [15], Improvement of Chatbot in Trading System [16], and many more applications. Moreover, various activation functions have been employed in generating neural-based controllers to play a video game [17].

Researchers raised questions about the characteristics of continuous nonlinear activation functions to act in the hidden layer of the neural network and examined if they can belong to a Banach space [18]. Moreover, it can be ineffective learning to tune the parameters of activation function in a neural network and difficult to train in the existence of non-differential activation functions as in the threshold networks [19].

That is why researchers introduced various approximations for the activation function. Mostafa et al. approximated the activation function for vector equalization with recurrent neural networks by a sum of shifted hyperbolic tangent functions [20]. Timmons et al. used function approximation techniques as a replacement of networks using hyperbolic tangent and sigmoid activation functions. Their result shows an improvement in training on the CPU by 10% to 37% [21]. In another research, two properties are emphasized to be good practice in approximating activation function based on good approximation of binary threshold and polynomial [22]. In another work, the authors considered the Complex-valued neural networks at which non-parametric activation functions are designed to be flexible in the complex domain. Therefore, its degree of freedom can adapt its shape given the training data [23]. Ohn et al. derived the essential width, depth and sparsity of a deep neural network that can estimate general activation functions in both regression and classification problems up to a certain approximation error [24]. To be more precise, Heaviside is expressed as a summation of two inverse trigonometric functions [25]. Furthermore, appropriate smoothness class has been introduced under approximation properties of networks with rectified linear unit activation function which are not positive definite [26].

This paper introduces mathematical formulas that match the exact representation of several activation functions including: Heaviside or Step Function, Rectified Linear Unit (ReLU), Parametric Rectified Linear Unit (PReLU) and thus the Leaky Rectified Linear Unit (Leaky ReLU), Exponential Linear Unit (ELU), Scaled Exponential Linear Units (SELU) and Exponential linear Squashing (ELiSH).

The remainder of this paper is organized as follows: In Section 2, exact and defined analytic forms of traditional and widely used activation functions are presented. Section 3 introduces the activation functions and their formulated exact mathematical equations. The Matlab code of some plotted functions is given with the newly formulated equations. Finally, a conclusion of the novel method has been developed.

II. DEFINED ANALYTIC EXPRESSIONS OF ACTIVATION FUNCTIONS

Traditionally, the sigmoid function and hyperbolic tangent are widely used as differential activation functions in feedforward neural networks. The sigmoid (logistic) activation function is defined in (1) [27] and a plot is shown in Fig. 1.

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (1)$$

The sigmoid function is characterized by it being computationally expensive, converges slowly and outputs not zero centered. The hyperbolic tangent (tanh) activation function [28] is defined in (2):

$$Y(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (2)$$

The hyperbolic tangent (tanh) activation function is a shifted version of the sigmoid function to become zero-centered as shown in Fig. 2.

Fig. 1. Sigmoid Function

Fig.2. Hyperbolic Tangent Function

The saturation at output boundaries of the above two nonlinear activation functions make the neural network unable to update weights during back-propagation learning and thus the vanishing gradient problem appears.

III. EXACT MATHEMATICAL FORMULATION OF OTHER SPECIAL ACTIVATION FUNCTIONS

In this section, each activation function is defined as given in the literature. After, the exact closed-form analytical formulas is expressed for Heaviside, Rectified Linear Unit, Parametric Rectified Linear unit, Exponential Linear Unit, and Scaled Exponential Linear Units activation functions.

A. Heaviside Function

The Heaviside or step function is given [29] as in (3):

$$U(z) = \begin{cases} 0, & z < 0 \\ A, & z \geq 0 \end{cases} \quad (3)$$

As seen from (3), the step function is zero for negative values of the input and some constant for positive input values. If the peak-to-peak amplitude is set to one ($A=1$), then it is called unit step function. Step function is highly used in operational calculus to solve differential equations which are widely used to model dynamics and engineering systems. The Heaviside exact closed-form analytical formula is derived as shown in (4):

$$U(z) = A0^{-z+\sqrt{z^2}} \quad (4)$$

Again “A” is defined to be the peak-to-peak amplitude of the step function and $z \in]-\infty, +\infty[$.

If $A=1$ then the plot of the step function will match exactly its original definition as shown in Fig. 3. The derivative of the Heaviside can then be derived and look like in (5):

Fig.3. Heaviside Function

$$U' = \frac{dU}{dz} = 0 \quad \forall z \quad (5)$$

As it can be observed in (5) that the derivative of Heaviside step function didn't yield a Dirac delta function at $z=0$. This issue will be discussed in a further investigation. The unilateral Laplace transform of the formulated step function will remain $(\frac{1}{s})$, where “s” is the complex-frequency parameter. The step function can be shifted on vertical y-axis by adding a subtraction parameter denoted by “B” as given in (6). An example is shown in Fig. 4 where the peak-to-peak amplitude is chosen $A=2$ and a shift of $B=1$:

$$U(z) = A0^{-z+\sqrt{z^2}} - B \quad (6)$$

Fig.4. Heaviside Function being shifted on the y-axis

Fig.5. Unit Step Function being shifted on the x-axis

Alternatively, the Heaviside function can be shifted on the horizontal x-axis by adding a parameter denoted by “C” as in (7):

$$U(z) = A0^{-(z-C)+\sqrt{(z-C)^2}} \quad (7)$$

Fig. 5 is an illustration of a unit step function ($A=1$) being shifted on the x-axis by $C=2$. Note that the “shift” mechanism is applicable to all upcoming derived formulas of activation functions.

B. Rectified Linear Unit (ReLU)

The ReLU is defined mathematically [30] as a combination of two linear part as in (8):

$$R(z) = \max\{0, z\} = \begin{cases} 0, & z < 0 \\ z, & z \geq 0 \end{cases} \quad (8)$$

The linear part of ReLU is easier to be optimized with gradient methods. In fact, it over comes the vanishing gradient problem because of it is linearity on some domain of definition. In the other hand, it keeps nonlinearity allowing an efficient learning from complex relationships within the data. This actually led to an exploding gradient. Moreover, ReLU simplifies the model by having an output with exact value equal to zero and not approximating to zero like in the sigmoid function. This reduced the computational complexity and characterize it with representational sparsity. ReLU is used in many interesting applications like in switching linear

encoding with designed rectified linear autoencoders models [27].

In (9), the ReLU exact closed form analytical formulation is presented in an elegant formula.

$$R(z) = Az0^{-z+\sqrt{z^2}} \quad (9)$$

Where A is the slope of the ReLU ramp part. A is fixed to one and a plot of ReLU will be then as in Fig. 6.

Fig.6. ReLU activation function

The derivative of ReLU function can then be obtained as (10) which matches the unit step function.

$$R' = \frac{dR}{dz} = 0^{-z+\sqrt{z^2}} = \begin{cases} 0, & z < 0 \\ 1, & z \geq 0 \end{cases} \quad (10)$$

The negative values substituted in (10) are strictly zero which will be highly affecting the vanishing gradient problem. However, the derivative of ReLU at zero is now defined. The unilateral Laplace transform of the formulated ReLU function in (9) will remain $(\frac{1}{s^2})$.

C. Parametric Rectified Linear Unit (PReLU)

The PReLU [32] is a modified version of ReLU to enable backpropagation at negative input values. Mathematically it is given as in (11):

$$P(\alpha, z) = \begin{cases} \alpha z, & z < 0 \\ z, & z \geq 0 \end{cases} \quad (11)$$

α is the slope at the negative part of the function. α can be optimized during learning at the backpropagation. However, if α is fixed to a certain value, say 0.01, which is entitled Leaky ReLU [33] then it doesn't provide accurate predictions for negative input values.

The PReLU exact closed form analytical formulation is given in (12) and sketched in Fig. 7 with $\alpha = 0.2$.

$$P(\alpha, z) = \alpha z 0^{z+\sqrt{z^2}} + z 0^{-z+\sqrt{z^2}} \quad (12)$$

Fig.7. PReLU activation function

The derivative of PReLU is then determined to be as in (13):

$$P' = \frac{dP}{dz} = 0^{-z+\sqrt{z^2}} + \alpha z 0^{z+\sqrt{z^2}} = \begin{cases} \alpha, & z < 0 \\ 1, & z \geq 0 \end{cases} \quad (13)$$

D. Exponential Linear Unit (ELU)

The ELU [34] is defined as given in (14):

$$E(\alpha, z) = \begin{cases} \alpha(e^z - 1), & z < 0 \\ z, & z \geq 0 \end{cases} \quad (14)$$

The advantage of ELU over ReLU is in its ability to circumvent overfitting while training. In addition, it can decrease the bias shift effects observed in ReLU. The ELU exact closed-form analytical formulation is then derived as in (15).

$$E(\alpha, z) = \alpha(e^z - 1)0^{z+\sqrt{z^2}} + z 0^{-z+\sqrt{z^2}} \quad (15)$$

Equation (15) can be plotted as shown in Fig. 8 where α is picked to be equal to 1.1.

Fig.8. ELU activation function

The derivative of ELU is obtained as in (16):

$$E' = \frac{dE}{dz} = 0^{-z+\sqrt{z^2}} + \alpha e^z 0^{z+\sqrt{z^2}} = \begin{cases} E(\alpha, z) + \alpha, & z < 0 \\ 1, & z \geq 0 \end{cases} \quad (16)$$

And the second derivative of ELU is shown in (17).

$$E'' = \alpha e^z 0^{z+\sqrt{z^2}} \quad (17)$$

E. Other Activation Functions

At this point, other activation functions can be derived easily from previous activation functions. For instance, the Scaled Exponential Linear Unit (SELU) [35] activation function is known in its internal normalization and its cannt die compared to ReLU. It is mathematical formula beside to its closed analytical form are given in (18).

$$SELU(\alpha, z) = \tau[\alpha(e^z - 1)0^{z+\sqrt{z^2}} + z 0^{-z+\sqrt{z^2}}] = \begin{cases} \alpha(e^z - 1), & z < 0 \\ z, & z \geq 0 \end{cases} \quad (18)$$

Another interesting activation function is the Exponential linear Squashing (ELiSH) [36] activation function. In (19), the equation presents the mathematical definition of ELiSH along with its exact form.

$$ELiSH(\alpha, z) = \frac{e^z - 1}{e^{-z} + 1} 0^{z+\sqrt{z^2}} + \frac{z}{e^{-z} + 1} 0^{-z+\sqrt{z^2}} = \begin{cases} \frac{e^z - 1}{e^{-z} + 1}, & z < 0 \\ \frac{z}{e^{-z} + 1}, & z \geq 0 \end{cases} \quad (19)$$

IV. CONCLUSION

In this paper, we introduced a novel explicit analytic form for several activation functions (Heaviside, ReLU, PReLU, ELU, SELU, ELiSH). The derivative of Heaviside is proven to be zero for the entire domain of definition. The derivative of the ReLU has been also presented along with their Unilateral Laplace transform. Those proposed explicit formulas of activation function would simplify the algorithm within the trained neural networks and would help in achieving higher accuracies. Namely, the derived formulas will speed up the evaluation of neural network, scalable to diverse central processing units existed in the market and building quicker arithmetic

logic units. Besides, they can be used easily and efficiently for analog applications. The results of this paper will yield remarkable applications in several areas like in operational calculus, engineering practice, science, and more specifically in different types of neural networks including Convolutional Neural Network, Recursive Neural Networks, Recurrent Neural Network, and Modular Neural Network.

The proposed functions will be studied in detail in the future. Their derivations and limitations will be presented along with their applications into the various field of study.

REFERENCES

- [1] N. Zhang, W. Wu and G. Zheng, "Convergence of gradient method with momentum for two-Layer feedforward neural networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 2, pp. 522 - 525, 2006.
- [2] S. T. Ahmad and K. P. Kumar, "Radial Basis Function Neural Network Nonlinear Equalizer for 16-QAM Coherent Optical OFDM," *IEEE Photonics Technology Letters*, vol. 28, no. 22, pp. 2507 - 2510, 2016.
- [3] D. Shnayder, T. Barbasova and Y. Lapteva, "Enhancing Blast Furnace Control Efficiency Based on Self-Organizing Kohonen Neural Networks," in 2018 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), Moscow, Russia, Russia, 2018.
- [4] Q. Song, Y. Wu and Y. Chai Soh, "Robust Adaptive Gradient-Descent Training Algorithm for Recurrent Neural Networks in Discrete Time Domain," *IEEE Transactions on Neural Networks*, vol. 19, no. 11, pp. 1841 - 1853, 2008.
- [5] Y. Xiao and J. Keung, "Improving Bug Localization with Character-Level Convolutional Neural Network and Recurrent Neural Network," in 2018 25th Asia-Pacific Software Engineering Conference (APSEC), Nara, Japan, 2018.
- [6] N. P. T. Kishna and S. Francis, "Intelligent tool for Malayalam cursive handwritten character recognition using artificial neural network and Hidden Markov Model," in 2017 International Conference on Inventive Computing and Informatics (ICICI), Coimbatore, India, 2018.
- [7] "IEE Colloquium on 'Neural Networks for Image Processing Applications' (Digest No.186)," in IEE Colloquium on Neural Networks for Image Processing Applications, London, UK, 1992.
- [8] M. Firdaus, S. Endah Pratiwi, D. Kowanda and A. Kowanda, "Literature review on Artificial Neural Networks Techniques Application for Stock Market Prediction and as Decision Support Tools," in 2018 Third International Conference on Informatics and Computing (ICIC), Palembang, Indonesia, 2018.
- [9] S. Chandra and S. Maheskar, "Offline signature verification based on geometric feature extraction using artificial neural network," in 2016 3rd International Conference on Recent Advances in Information Technology (RAIT), Dhanbad, India, 2016.
- [10] D. A. Woten, J. Luth and M. El-Shenawee, "Interpreting Artificial Neural Networks for Microwave Detection of Breast Cancer," *IEEE Microwave and Wireless Components Letters*, vol. 17, no. 12, pp. 825 - 827, 2007.
- [11] Y. Jiang, J. Yao and Z. Qian, "A Method of Forecasting Thunderstorms and Gale Weather Based on Multisource Convolution Neural Network," *IEEE Access*, vol. 7, pp. 107695 - 107698, 2019.
- [12] Y.-T. Liu, Y.-Y. Lin, S.-L. Wu, C.-H. Chuang and C.-T. Lin, "Brain Dynamics in Predicting Driving Fatigue Using a Recurrent Self-Evolving Fuzzy Neural Network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 347 - 360, 2016.
- [13] M. Smolyakov, A. Frolov, V. Volkov and I. Stelmashchuk, "Self-Driving Car Steering Angle Prediction Based On Deep Neural Network An Example Of CarND Udacity Simulator," in 2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT), Almaty, Kazakhstan, 2018.
- [14] M.-T. Duong, T.-D. Do and M.-H. Le, "Navigating Self-Driving Vehicles Using Convolutional Neural Network," in 2018 4th International Conference on Green Technology and Sustainable Development (GTSD), Ho Chi Minh City, Vietnam, 2018.
- [15] G. Ayhan, Ç. Şenel, Z. Eda Uran and B. U. Töreyn, "Landing page component classification with convolutional neural networks for online advertising," in 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2018.
- [16] S. Prasomphan, "Improvement of Chatbot in Trading System for SMEs by Using Deep Neural Network," in 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Chengdu, China, 2019.
- [17] T. Guan Tan, J. Teo and P. Anthony, "A comparative investigation of non-linear activation functions in neural controllers for search-based game AI engineering," *Artificial Intelligence Review*, vol. 41, no. 1, pp. 1-25, 2011.
- [18] T. Chen and H. Chen, "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems," *IEEE Transactions on Neural Networks*, vol. 6, no. 4, pp. 911 - 917, 1995.
- [19] S. M. I. L. C. a. C.-K. S. Guang-Bin Huang, "Universal Approximation Using Incremental Constructive Feedforward Networks With Random Hidden Nodes," *IEEE TRANSACTIONS ON NEURAL NETWORKS*, vol. 17, no. 4, pp. 879-891, 2006.
- [20] M. Mostafa, W. G. Teich and J. Lindner, "Approximation of activation functions for vector equalization based on recurrent neural networks," in 2014 8th International Symposium on Turbo Codes and Iterative Information Processing (ISTC), Bremen, Germany, 2014.
- [21] N. G. Timmons and A. Rice, "Approximating Activation Functions," *arXiv*, 2020.
- [22] B. DasGupta and G. Schnitger, "The Power of Approximating: a Comparison of Activation Functions," in NIPS'92: Proceedings of the 5th International Conference on Neural Information Processing Systems November, 1992.
- [23] S. Scardapane, S. Van Vaerenbergh, A. Hussain and A. Uncini, "Complex-valued Neural Networks with Non-parametric Activation Functions," *IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE*, pp. 1-12, 2018.
- [24] I. Ohn and Y. Kim, "Smooth Function Approximation by Deep Neural Networks with General Activation Functions," *MDPI- Entropy*, vol. 21, no. 627, 2019.
- [25] J. Venetis, "An Analytic Exact Form of the Unit Step Function," *Mathematics and Statistics*, vol. 2, no. 7, pp. 235-237, 2014.
- [26] M. H.N., "Function approximation with zonal function networks with activation functions analogous to the rectified linear unit functions," *Journal of Complexity*, vol. 51, pp. 1-19, 2019.
- [27] J. Han and C. Morag, "The influence of the sigmoid function parameters on the speed of backpropagation learning," *Lecture Notes in Computer Science*, vol. 930, pp. 195-201, 1995.
- [28] A. Wibisono, M. Roby Alhamidi, A. Nurhadiyatna and W. Jatmiko, "Hyperbolic tangent activation function on FIMT-DD algorithm analysis for airline big data," in 2017 International Workshop on Big Data and Information Security (IWBIIS), Jakarta, Indonesia, 2017.
- [29] R. N. Bracewell, *The Fourier transform and its applications* (3rd ed.), New York: McGraw-Hill, 2000.
- [30] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in 27th International Conference on International Conference on Machine Learning, USA: Omnipress, 2010.
- [31] L. Johnson and C. Corcoran, "Switched linear encoding with rectified linear autoencoders," *arXiv*, 2013.
- [32] K. He, X. Zhang, S. Ren and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *arXiv:1502.01852*, 2015.
- [33] A. L. Maas, A. Y. Hannun and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in International Conference on Machine Learning, 2013.
- [34] D.-A. Clevert, T. Unterthiner and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," *arXiv:1511.07289*, 2015.
- [35] G. Klambauer, T. Unterthiner, A. Mayr and S. Hochreiter, "Self-Normalizing Neural Networks, Advances in Neural Information Processing Systems," *arXiv:1706.02515*, 2017.
- [36] C. Nwankpa, W. Ijomah, A. Gachagan and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," *arXiv. 1811.03378*, 2018.