REVIEW OF ACTIVATION FUNCTIONS IN NEURAL NETWORKS

Oshank Kashyap, Kriti Kumari, Kartikay Yadav, Jatin Baweja, Vintee Vats, Mohit Sharma,

Dronacharya College of Engineering, oshankashyap@protonmail.com, itzkriti13@gmail.com, manojkartikay@gmail.com, jbaweja24@gmail.com, vatsvinti5@gmail.com, mohitsharma632004@gmail.com

Abstract: A neural network is a machine-learning solid technique that mimics human brain activity. It consists of various layers, including an input, an output, and sometimes multiple hidden layers. In a human brain, all of the information in the cell is summed up, with the frequency of incoming impulses fluctuating. The activation function determines the transmission of a signal to the next layer or the activation of a neuron. A threshold function processes the input sum in the neural network and provides an output signal. The role and types of activation functions in neural networks will be discussed in this paper.

Number: 10.14704/nq.2022.20.7.NQ33217 **Neuro Quantology 2022; 20(7):1736-1740**

INTRODUCTION

With the development in the computer science field, the natural thinking process of the human brain inspired the designing of an artificial intelligence system called the artificial neural networks (ANNs). This system, designed to solve more complex problems, is not only up for machine learning but also plays a vital role in pattern recognition. Like a website network, Neural Network consists of millions of interconnected neurons present in different layers. Each neuron in an artificial neural network has an activation function that governs the neuron's activity, provides non-linearity to the ANN, and limits the input data to a finite value [1].

The three types of activation functions are saturated, unsatbrain inspired rated, and adaptive activation functions. The sigmoid function is renowned for its differentiable property, which suits the backpropagation training algorithm in the best manner. The multistate activation function (MSAF) [2] is a novel activation function that aids in categorizing saturated activation functions. Xu et al. [3] proposed the tanh activation function that penalizes its negative part gradient, avoiding the saturation problem.

Unsaturated activation functions like ReLU [4] help to overcome the problem of saturation by the elimination of exponential terms on its backpropagation. The LReLU activation function addresses ReLU's issue by providing a constant variable in the negative portion of the activation function [5].

Furthermore, adaptive activation functions such as PReLU [6] assigned a trainable coefficient to the harmful component of the LReLU activation function, replacing the constant coefficient in the LReLU activation function. Max out activation function [7] left ReLU and LReLU activation functions behind with the help of additional adaptive coefficients.

Further, in this paper, all the activation functions will be reviewed in detail, and by the end, a strong comparison will be formulated.

TYPES OF ACTIVATION FUNCTIONS [27]

The most basic units in constructing a neural network are net input. They are processed and turned into an output result known as unit activation using a scalar-to-scalar transformation termed the activation function, threshold function, or transfer function.

• Sigmoid Activation Function

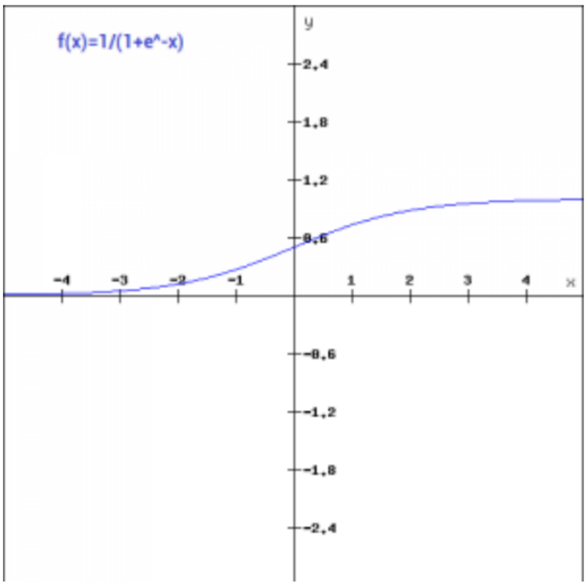
The (-infinity; +infinity) input range is converted to [0; 1] using the sigmoid activation function. It has a smooth derivative and is nonlinear by nature [8]. It is the most widely used activation function since it is a nonlinear function [9]. It's possible to write it as:

$$g(a) = 1/e^{-a}$$

The sigmoid function is a continuously differentiable smooth S-shaped function. The derivative of the function is:

$$ft(b) = 1\text{-sigmoid}(b)$$

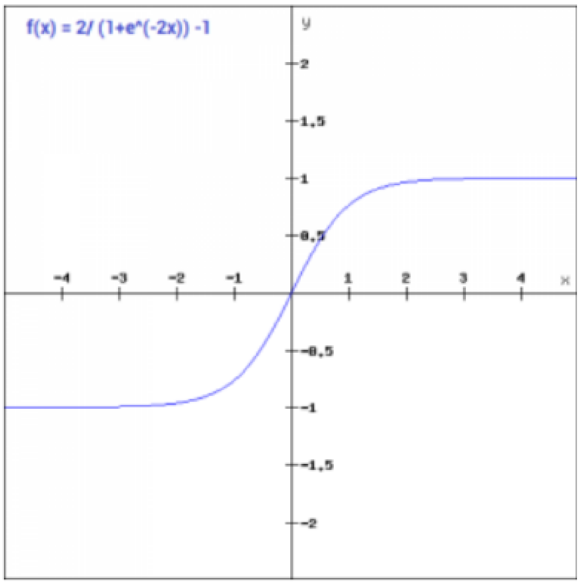
Furthermore, the sigmoid function is not symmetric around zero, implying that the signs of every neuron's output value will be the same.



• Hyperbolic Tangent Activation Function

The Hyperbolic Tangent function has a pretty similar structure to the Sigmoid function. On the other hand, the process squashes the input value to a range between [-1, 1]. The [8] exponential function has one significant advantage over the sigmoid function: its derivative is steeper, as seen in the image, meaning it can gain more value. This function tends to vanish despite having a more significant gradient due to its restricted output value. The following is a definition for this function:

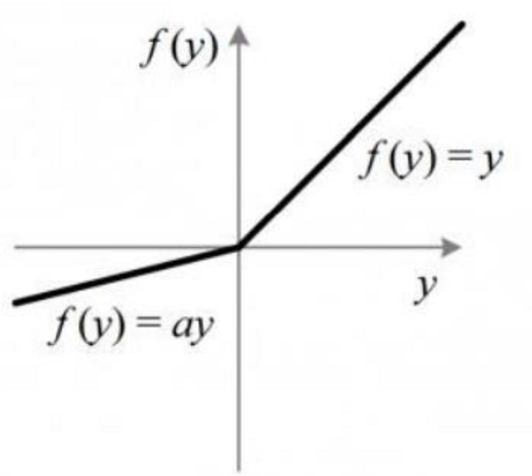
$$\tanh(k) = (2/1+e^{-2k}) -1$$



• Swish Activation Function

Swish is a self-gated activation function found through reinforcement learning. There is no upper bound for the [8] Swish function. Swish is a non-monotonic activation function that is smooth and bounded below and unbounded above, comparable to ReLU.

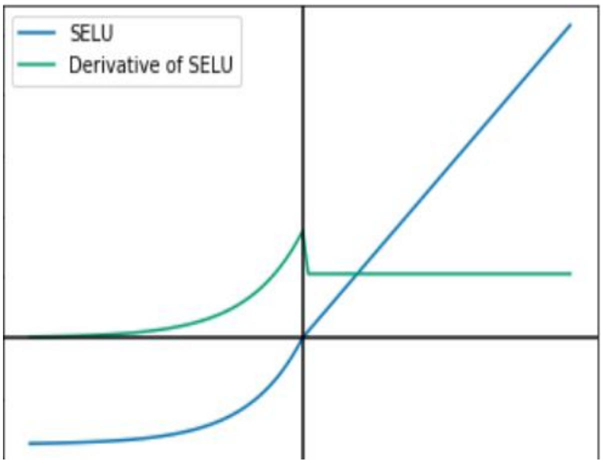
$$\text{Swish}(z) = (z/1 + e^{-z})$$



• Mish Activation Function

Mish is a brand-new activation function that looks and behaves like the Swish activation function. The concave region of the function is where the primary difference lies. Surprisingly, it consistently outperforms ReLU [8] and Swish despite requiring more processing. The activation function of Mish can be written as follows:

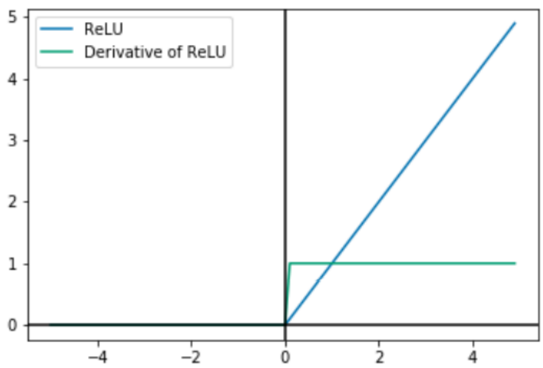
$$\text{mish}(y) = \tanh(\log(1 + e^{y}))$$



• **ReLU**

Rectified linear unit is a nonlinear activation function denoted by the acronym ReLU [9]. Nair and Hinton proposed it for Restricted Boltzmann Machines [10]. It is the most utilized activation function in a neural network by deep learning researchers [8]. The function's output value ranges from zero to infinity. It is more efficient than other activities because it does not engage all neurons simultaneously; instead, just a small number of neurons are stimulated simultaneously [9]. It can emit an actual zero value, allowing hidden layers in neural networks to be activated with one or more true zero values,[26] increasing its representational sparsity. Negative values are converted to zero, resulting in a zero output from every opposing unit. This is ReLU's most serious flaw. This is referred to as "dying ReLU" [8]. The following is the definition of the ReLU function:

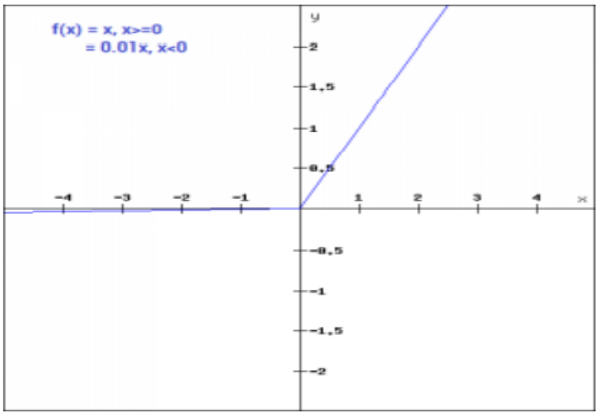
$$f(a) = \max(0, a) = \begin{cases} a & if\ a \geq 0 \\ 0 & if\ a < 0 \end{cases} \quad [25]$$



• **LReLU**

Leaky Rectifier Activation Functions is the abbreviation for LReLU. It is one of the early ReLU-based rectified activation functions. An LReLU permits the unit to produce a slight gradient when it is saturated. It solves the issue of the rise of ReLU being zero for negative x values by providing a new parameter for the negative portion of the function, namely Slope [10].

$$f(y) = \max(0, y) = \begin{cases} y & if\ y \geq 0 \\ 0.01y & if\ y < 0 \end{cases} \quad [26]$$



COMPARISON

Functions	Pros	Cons
Sigmoid	is Preferred for classification problems.	Gradient vanishes to zero.
Tanh	has a steeper derivative.	Gradient tends to vanish.
ReLU	All neurons are not activated at the same time.	There may be dead neurons.
LLC	It can be used in the case of dead neurons.	The x coefficient is predetermined, and the Neural Network is not responsible for determining it.
Swish	It even surpasses the ReLU function.	It is slower to compute.

PReLU	It solves the problem of zero gradients, and it has better performance.	It solves the problem of zero gradients, and it has better performance.
Mish	It is better than swish and ReLU functions in accuracy and consistency.	It is computationally costly.

CONCLUSION

This paper comes up with a directive of various activation functions involved in the field of deep learning and the significance of activation functions in evolving an efficient deep learning model. This paper spotlight the [9] need for non-linearity in neural networks here, different types of activation function have been explained with the help of their graphs and their python code. These activation functions came into existence through numerous experiments and have evolved. A brief comparison of the activation functions based on their pros and cons has been reviewed in the paper. This writing does not cover numerous activation functions as they aren't extensively used in deep learning. Instead, we have put the stress on the majorly used activation functions

.

ACKNOWLEDGMENTS

First and Foremost, We would like to extend a heartfelt gratitude to our professor and supervisor, Dr. Manoj Kumar for providing us an opportunity to do the review paper work in Dronacharya College Of Engineering and giving us all incomparable support, which made us complete the outline duly. We are ineffably indebted to our English faculty, Ms. Anchal Bhutani who took a keen interest in our task and guided us all along, till the completion of it by providing all the necessary information developing a crystal-clear understanding of the same. Moreover, we are tremendously grateful to our beloved parents and siblings for their elevating motivation and stabilizing supervision.

REFERENCES

[1] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs).” arXiv, Feb. 22, 2016. Accessed: May 30, 2022. [Online]. Available: http://arxiv.org/abs/1511.07289

[2] C. Cai, Y. Xu, D. Ke, and K. Su, “Deep Neural Networks with Multistate Activation Functions,” Comput. Intell. Neurosci., vol. 2015, pp. 1–10, 2015, DOI: 10.1155/2015/721367.

[3] B. Xu, R. Huang, and M. Li, “Revise Saturated Activation Functions.” arXiv, May 02, 2016. Accessed: May 30, 2022. [Online]. Available: http://arxiv.org/abs/1602.05980

[4] X. Glorot, A. Bordes, and Y. Bengio, “Deep Sparse Rectifier Neural Networks,” p. 9.

[5] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier Nonlinearities Improve Neural Network Acoustic Models,” p. 6.

[6] X. Jin, C. Xu, J. Feng, Y. Wei, J. Xiong, and S. Yan, “Deep Learning with S-shaped Rectified Linear Activation Units.” arXiv, Dec. 22, 2015. Accessed: May 30, 2022. [Online]. Available: http://arxiv.org/abs/1512.07030

[7] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, “Maxout Networks.” arXiv, Sep. 20, 2013. Accessed: May 30, 2022. [Online]. Available: http://arxiv.org/abs/1302.4389

[8] A. D. Rasamoelina, F. Adjailia, and P. Sincak, “A Review of Activation Function for Artificial Neural Network,” in 2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI), Germany, Slovakia, Jan. 2020, pp. 281–286. DOI: 10.1109/SAMI48414.2020.9108717.

[9] S. Sharma, S. Sharma, U. Scholar, and A. Athaiya, “ACTIVATIONFUNCTIONSINNEURALNETWORKS,”vol.4,no.12,p.7,2020.

[10] B. Ding, H. Qian, and J. Zhou, “Activation functions and their characteristics in deep neural networks,” in 2018 Chinese Control And Decision Conference (CCDC), Shenyang, Jun. 2018, pp. 1836–1841. DOI: 10.1109/CCDC.2018.8407425.

[11.] STINCHCOMBE, M., & WHITE, H. (1989, DECEMBER). UNIVERSAL APPROXIMATION USING FEEDFORWARD NETWORKS WITH NON-SIGMOID HIDDEN LAYER ACTIVATION FUNCTIONS. IN IJCNN INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS.

[12] J. Pennington, S. S. Schoenholz, and S. Ganguli, "Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice," arXiv:1711.04735 [cs, stat], Nov. 2017. arXiv: 1711.04735. [15] M. Tanaka, "Weighted Sigmoid Gate Unit for an Activation Function of Deep Neural Network," arXiv:1810.01829 [cs], Oct. 2018. arXiv: 1810.01829

[13] D. Misra, "Mish: A Self Regularized Non-Monotonic Neural Activation Function," arXiv:1908.08681 [cs, stat], Oct. 2019. arXiv: 1908.08681.

[14] S. K. Roy, S. Manna, S. R. Dubey, and B. B. Chaudhuri, "LiSHT: NonParametric Linearly Scaled Hyperbolic Tangent Activation Function for Neural Networks," arXiv:1901.05894 [cs], Dec. 2018. arXiv: 1901.05894.

[15] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," arXiv:1803.08375 [cs, stat], Feb. 2019. arXiv: 1803.08375.

[16] L. Lu, Y. Shin, Y. Su, and G. E. Karniadakis, "Dying ReLU and Initialization: Theory and Numerical Examples," arXiv:1903.06733 [cs, math, stat], Nov. 2019. arXiv: 1903.06733.

[17] D. Arpit and Y. Bengio, "The Benefits of Over-parameterization at Initialization in Deep ReLU Networks," arXiv:1901.03611 [cs, stat], Oct. 2019. arXiv: 1901.03611.

[18] J. Berner, D. Elbrachter, and P. Grohs, "How degenerate is the ¨ parametrization of neural networks with the ReLU activation function?," arXiv:1905.09803 [cs, math, stat], Oct. 2019. arXiv: 1905.09803.

[19] A. Dereventsov, A. Petrosyan, and C. Webster, "Neural network integral representations with the ReLU activation function," arXiv:1910.02743 [cs, stat], Oct. 2019. arXiv: 1910.02743.

[20] S. C. Douglas and J. Yu, "Why ReLU Units Sometimes Die: Analysis of Single-Unit Error Backpropagation in Neural Networks," arXiv:1812.05981 [cs, eess, stat], Dec. 2018. arXiv: 1812.05981.

[21] J. Ferlez and Y. Shoukry, "AReN: Assured ReLU NN Architecture for Model Predictive Control of LTI Systems," arXiv:1911.01608 [cs, eess, math], Nov. 2019. arXiv: 1911.01608.

[22] S. Hayou, A. Doucet, and J. Rousseau, "On the Impact of the Activation Function on Deep Neural Networks Training," arXiv:1902.06853 [cs, stat], May 2019. arXiv: 1902.06853.

[23] S. Goel, S. Karmalkar, and A. Klivans, "Time/Accuracy Tradeoffs for Learning a ReLU with respect to Gaussian Marginals," arXiv:1911.01462 [cs, stat], Nov. 2019. arXiv: 1911.01462.

[24] M. Milletar´ı, T. Chotibut, and P. E. Trevisanutto, "Mean Field Theory of Activation Functions in Deep Neural Networks," arXiv:1805.08786 [cs, stat], June 2019. arXiv: 1805.08786.

[25] res.mdpi.com

[26] Philip Naveen. "Phish: A Novel Hyper-Optimizable Activation Function", Institute of Electrical and Electronics Engineers (IEEE), 2022

[27] www.ijeast.com

1740

Author Information

Oshank Kashyap, Undergraduate student, Department of Computer Science Engineering, Dronacharya College of Engineering.

Kriti Kumari, Undergraduate student, Department of Computer Science Engineering, Dronacharya College of Engineering.

Kartikay Yadav, Undergraduate student, Department of Computer Science Engineering, Dronacharya College of Engineering.

Jatin Baweja, Undergraduate student, Department of Computer Science Engineering, Dronacharya College of Engineering.

Vintee Vats, Undergraduate student, Department of Computer Science Engineering, Dronacharya College of Engineering.

Mohit Sharma, Undergraduate student, Department of Computer Science Engineering, Dronacharya College of Engineering.