A Brief Review of the Most Recent Activation Functions for Neural Networks

Marina Adriana Mercioni
Department of Computer and Information Technology
Politehnica University Timisoara
Timisoara, Romania
marina.mercioni@cs.upt.ro

Stefan Holban
Department of Computer and Information Technology
Politehnica University Timisoara
Timisoara, Romania
stefan.holban@cs.upt.ro

2023 17th International Conference on Engineering of Modern Electric Systems (EMES) | 979-8-3503-1063-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/EMES58375.2023.10171705

***Abstract*— Even though the majority of the current functions have shortcomings, this study looks at a few activation function capabilities that might lead to performance enhancement. The research on neural networks still shows a lot of interest in the activation function since it can enhance performance. With or without trainable parameters, other adaptive activation functions have been put forth that have demonstrated to lead to better results than the benchmark. These studies outline the characteristics, benefits, constraints, and directions of such types of applications. Due to their shortcomings, several of those functions are now regarded as deprecated. The primary emphasis of such functions is on fundamental elements that are thought to be necessary for learning, such as monocity, derivatives, and finite of their range. The goal of this research article is to present and assess the most popular and recent activation functions. This will go through their characteristics, advantages and disadvantages, formulation, and usage.**

***Keywords*— activation; convergence; deep learning; function; neural networks; parameter; performance; trainable**

I. Introduction

Nowadays activation functions still present a high interest due to their role within a neural network. An activation function is used in an artificial neural network (ANN) to aid it in learning complex patterns in data. There are several reasons why a network should have non-linear activation functions. The power of an activation function to bring non-linearity to a neural network is its most important requirement [1]. Earlier studies focused on the appropriate definition and implementation of an activation function within a neural network architecture.

Activations in a deep neural network are either fixed before training or trainable. In recent years, researchers have suggested many activations by merging established functions. Several of these functions contain trainable parameters or hyperparameters. The parameters of trainable activation functions are adjusted during training.

The Sigmoid function was the most often used activation function. Continuous feedforward neural networks with a single internal, hidden layer and any continuous sigmoidal nonlinearity may arbitrarily well represent arbitrary decision regions [2].

Rumelhart-Hinton-Williams' multilayer neural networks with at least one hidden layer whose output functions are sigmoid functions may approximate any continuous mapping [3]. The nonlinear neatly defined with the sigmoidal nodes, as well as the linear combination parameters, are updated in the approximation [4].

However, when the Rectifier Linear Unit (ReLU) was developed, it quickly became a superior substitute for the Sigmoid function because of its strong influence on many artificial intelligence tasks [5].

Activation functions are critical to the overall network's effectiveness. As a result, picking the right activation function in neural network modeling is crucial [6].

The accuracy of a Neural Network's prediction is determined by the number of layers utilized and, more crucially, the type of activation function employed [7].

The article is organized as follows: Section 2 discusses the related work. Section 3 presents the used activation functions. The results are presented in Section 4. Section 5 concludes with final remarks.

II. State of The Art

ReLU became the most popular function used in research literature. For example, MobileNets use both batch normalization [8] and ReLU nonlinearities for both layers.

In [9-11] Network Pruning (NP) is used and its goal is to create a light network by deleting irrelevant elements of a well-trained but large-scale network.

Current NP research has mostly focused on structured pruning, which filters out unnecessary channels by applying sparsity functions to the convolutional layers of well-trained massive models.

PyramidNet was introduced in [12], and they used the pyramidal bottleneck residual unit (PBRU) [13]. The PBRU is made up of three convolution layers, four batch normalization (BN) levels, and two rectified linear unit (ReLU) layers. Swim is proposed in [14] as a flexible, effective, and outstanding variant of Swish [15].

[16] studied neural network verification using piecewise affine (pwa) activation functions, allowing theorem provers to encode the verification issue. They offer the first formalization of pwa activation functions for an interactive theorem prover specialized to validating neural networks within Coq using the real analysis package Coquelicot. They built the popular pwa activation function ReLU as a proof-of-concept.

In [17], a theoretical explanation for the spectral bias of ReLU neural networks is presented by exploiting linkages with finite element method theory. They forecast that changing the activation function to a piecewise linear B-spline, such as the Hat function, will remove the spectral bias, which they empirically validate in a variety of scenarios. Their empirical investigations further suggest that employing stochastic gradient descent (SGD) [18] and Adam [19], neural networks with the Hat activation function can be trained substantially quicker.

As an alternative, [20] developed a cutting-edge non-monotonic activation function known as the Negative Stimulated Hybrid activation function (Nish). It is a Rectified Linear Unit (ReLU) function in the positive region and a sinus-sigmoidal function in the negative region. It integrates a sigmoid and a sine function, acquiring new dynamics above regular ReLU. In [21], they present a new activation function called Smooth Maximum Unit (SMU) that is based on approximation of previous activation functions like Leaky ReLU [22].

The activation function Phish [23] is proposed. It is a composite function with no discontinuities in the differentiated graph on the given domain. Phish, Swish, Sigmoid, and TanH (hyperbolic tangent) were used to build four generalized networks. The output function was SoftMax [24].

Different functions were developed, however due to their drawbacks, other new activation functions appeared. These most powerful functions are also among the most well-known artificial intelligence activation functions in machine learning and deep learning research [25-28].

Current study has revealed biological neurons in layers two and three of the human brain that have oscillating activation functions and can learn the XOR function independently.The presence of oscillating activation functions in biological brain neurons may explain a portion of the performance difference between biological and artificial neural networks. The authors offer four novel oscillating activation functions that allow individual neurons to learn the XOR function without the need for explicit feature engineering. The research investigates the use of oscillating activation functions to solve classification problems with fewer neurons and shorten training time [29].

The impact of increasing capacity by using learnable parametric activation functions (PAFs) has not been studied, even though prior studies looked at the impact of changing model breadth and depth on robustness. They look into how using adversarial training together with learnable PAFs could improve robustness [30].

[31] presents novel smooth approximations of a non-differentiable activation function by convolving it using approximate identities. They provide smooth approximations of Leaky ReLU in particular, and show that they outperform numerous well-known activation functions in different datasets and models. [32] introduces ErfAct and Pserf, two novel non-monotonic smooth trainable activation functions. Studies show that the suggested functions greatly increase network performance when compared to commonly utilized activations such as ReLU, Swish, and Mish [33].

Although being simple and effective, the frequently used activation function ReLU has a few drawbacks, one of which is the Dying ReLU issue. [34] presents a unique activation function named Serf, which is self-regularized and nonmonotonic in design, to address such issues. Serf, like Mish, is a function in the Swish category.

Numerous experiments with different state-of-the-art architectures on computer vision (CV) and natural language processing (NLP) tasks show that Serf vastly outperforms ReLU (baseline) and other activation functions, including both Swish and Mish, with a significantly larger margin on deeper architectures. Gated Linear Units (GLU) are the result of two linear projections, one of which is first put through a sigmoid function. GLU can be modified by substituting other nonlinear or even linear functions for the sigmoid [35].

The use of periodic activation functions for neural models is suggested in [36], which demonstrates that these networks—also known as Sirens or "sinusoidal representation networks"—are well adapted for describing complex signals. The *'m-arcsinh*,' a customized version of the inverse hyperbolic sine function 'arcsinh,' is shown in [37].

Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP) are examples of machine learning (ML)-based supervised algorithms that have the potential to extract information from data due to kernel and activation functions. Tanh Exponential activation function (TanhExp) is presented in [38] as a new activation function that can considerably improve the network's performance on image classification tasks. In [39], a novel activation function called CoLU is proposed, giving outstanding results.

III. Activation Functions

In this section we will present the functions we have used in this study to describe and evaluate the most popular and recent activation functions. This will go through their characteristics, benefits and drawbacks, formulation, and application.

A. ReLU

The Rectified Linear Unit (ReLU) [40] (as seen in 1) function has a ramp shape; it is continuous, monotonic, and computationally efficient. Its derivative is monotonic. The ReLU function has a positive codomain [0, ∞).

$$f(x) = \begin{cases} x \ \ if \ x \geq 0 \\ 0 \ \ if \ x < 0 \end{cases} \tag{1}$$

B. Swish

A smooth, non-monotonic activation function called Swish (as seen in 2) regularly equals or surpasses ReLU on deep networks used in many complex fields [15].

$$f(x) = \frac{x}{1+e^{-x}} \tag{2}$$

C. Mish

Compared to Swish and ReLU activation functions, *Mish (*as seen in 3*)* is a new self-regularized non-monotonic activation function that attempts to equal or outperform the effectiveness of neural network designs [41].

$$f(x) = \frac{x((1+e^{x})^{2}-1)}{((1+e^{x})^{2}+1)} \tag{3}$$

D. CoLU

Collapsing Linear Unit (CoLU) (as seen in 4) is similar to Mish and Swish and has similar characteristics to them. CoLU has a range of [≈-0.3762, ∞). Mish has a range of [≈-0.3087, ∞), while Swish has a range of [≈-0.2784, ∞) [39].

$$f(x) = \frac{x}{1-xe^{-(x+e^{x})}} \tag{4}$$

E. Serf

Serf is an activation function inspired by the development of Mish. Serf [34] is defined as:

$$f(x) = x(ln(1 + e^x)) \quad (5)$$

F. Nipuna

As it has a tiny positive slope in the negative part of the input, the Nipuna activation function solves the dying ReLU issue to estimate the appropriate slope of the negative region of the input data. Combines the benefits of the ReLU and Swish activations to enhance gradient descent convergence towards global minima and optimize time-intensive computing for deeper layers with large parameter dimensions [42].

$$f(x) = max(\frac{x}{1+e^{-\beta x}}, x) \quad (6)$$

As seen in (6), the parameter β value is either a value is either a constant or trainable parameter; for this study, the authors used β = 1 a trainable parameter.

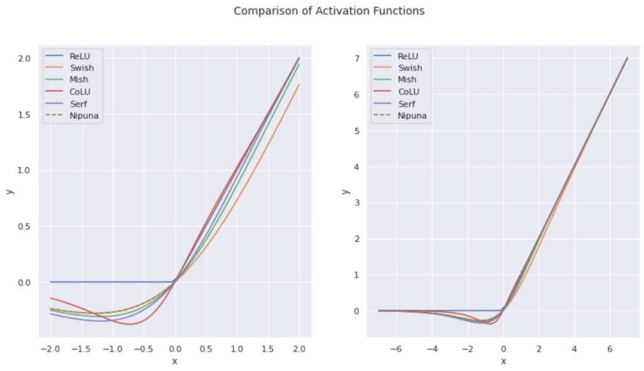

Fig. 1. Activation functions.

Fig. 1 highlights the activation functions used in the current study to emphasize the novel functions and their results.

IV. Results

The experimental results based on the Jena Climate dataset recorded by Max Planck Institute for Biogeochemistry will be presented in this section. The data base consists of fourteen parameters, including temperature, pressure and humidity measured once per 10 minutes [43].

In terms of the model, we considered a lightweight architecture (Fig. 2) of 16 units of dense layer with "*rmsprop*" optimizer, "*mse*" (mean squared error) loss, and trained it for 20 epochs.

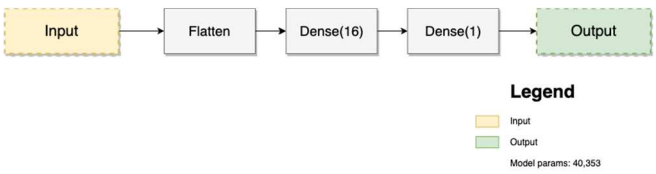

Fig. 2. The architecture.

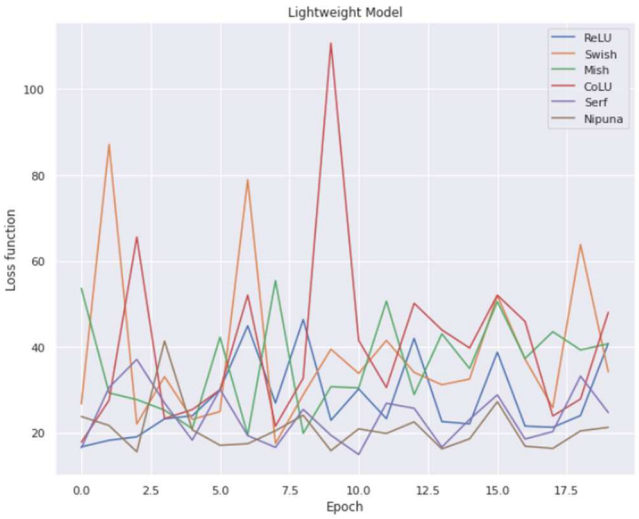

Fig. 3. Experimental results.

TABLE I. Experimental Results

Activation function	**Validation MAE (Mean Absolute Error)**
ReLU	3.9390
Swish	4.7286
Mish	4.6560
CoLU	4.7591
Serf	3.7074
Nipuna	**3.3954**

[a.] Epochs=20

According to Table I, we can see that the latest function, Nipuna, obtained the smallest mean absolute error (Fig 3).

V. Conclusion

More activation functions were analyzed throughout the study. The benefits of ReLU and Swish are carried over into the Nipuna activation function, giving the smallest error. Additionally, the Serf activation function performed well, ranking in the top 3 best functions.

As future research, we would like to extend the experiments to larger architectures and datasets to evaluate novel functions' behavior.

References

[1] Dabal Pedamonti, "Comparison of non-linear activation functions for deep neural networks on MNIST classification task", doi.org/10.48550/arXiv.1804.02763, 2018.

[2] George, Cybenko. (1989). "Approximation by superpositions of a sigmoidal function". Mathematics of Control, Signals, and Systems, 2(4):303-314. doi: 10.1007/BF02551274.

[3] K., Funahashi. (1989). "On the approximate realization of continuous mappings by neural networks". Neural Networks, 2(3):183-192. doi: 10.1016/0893-6080(89)90003-8.

[4] Andrew, R., Barron. (1993). "Universal approximation bounds for superpositions of a sigmoidal function". IEEE Transactions on Information Theory, 39(3):930-945. doi: 10.1109/18.256500.

[5] Nair, Vinod and Hinton, Geoffrey E. “Rectified linear units improve restricted boltzmann machines”. pp. 807–A ¸S814. ˇ In Proc. ICML, volume 30, 2010.

[6] Szandała, Tomasz. “Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks.” ArXiv abs/2010.09458 (2020).

[7] Siddharth Sharma et al., “ACTIVATION FUNCTIONS IN NEURAL NETWORKS”, International Journal of Engineering Applied Sciences and Technology, Vol. 4, Issue 12, ISSN No. 2455-2143, Pages 310-316, 2020.

[8] Ioffe, Sergey and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” International Conference on Machine Learning (2015).

[9] Xiaohan Ding, Guiguang Ding, Jungong Han, and Sheng Tang. “Auto-balanced filter pruning for efficient convolutional neural networks”. In AAAI, volume 32, 2018.

[10] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. “Learning efficient convolutional networks through network slimming”. In ICCV, pages 2755–2763, 2017.

[11] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. “Learning structured sparsity in deep neural networks”. In NIPS, pages 2074–2082, 2016.

[12] Lee, Kyu-Yul and Nam Yul Yu. “End-to-End Learning-Based Wireless Image Recognition Using the PyramidNet in Edge Intelligence.” (2023).

[13] D. Han, J. Kim, and J. Kim, “Deep pyramidal residual networks,” 2020.

[14] Maryam Abdool et al., “Swim: A General-Purpose, High-Performing, and Efficient Activation Function for Locomotion Control Tasks”, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2023.

[15] Prajit Ramachandran et al, “Searching for Activation Functions”, doi.org/10.48550/arXiv.1710.05941 2017.

[16] Andrei Aleksandrov et al, “Formalizing Piecewise Affine Activation Functions of Neural Networks in Coq”, doi.org/10.48550/arXiv.2301.128932023, 2023.

[17] Qingguo Hong et al., “On the Activation Function Dependence of the Spectral Bias of Neural Networks”, doi.org/10.48550/arXiv.2208.049242022, 2022.

[18] Ruder, S. “An overview of gradient descent optimization algorithms”. ArXiv Preprint ArXiv:1609.04747, doi.org/10.48550/arXiv.1609.04747, 2016.

[19] Diederik P. Kingma, Jimmy Ba, “Adam: A Method for Stochastic Optimization”, doi.org/10.48550/arXiv.1412.6980, 2014.

[20] Anagün, Yildiray and Şahin Işık. “Nish: A Novel Negative Stimulated Hybrid Activation Function.” ArXiv abs/2210.09083 (2022).

[21] Koushik Biswas et al., “SMU: SMOOTH ACTIVATION FUNCTION FOR DEEP NETWORKS USING SMOOTHING MAXIMUM TECHNIQUE”, 2022.

[22] Bing Xu et al., “Empirical Evaluation of Rectified Activations in Convolutional Network”, doi.org/10.48550/arXiv.1505.00853, 2015.

[23] Naveen, P R. “Phish: A Novel Hyper-Optimizable Activation Function”, 2021.

[24] Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron. "Softmax Units for Multinoulli Output Distributions". Deep Learning. MIT Press. pp. 180–184. ISBN 978-0-26203561-3, 2016.

[25] Mercioni, Marina Adriana, and Stefan Holban. "The most used activation functions: Classic versus current." 2020 International Conference on Development and Application Systems (DAS). IEEE, 2020.

[26] Mercioni, Marina Adriana, and Stefan Holban. "P-swish: Activation function with learnable parameters based on swish activation function in deep learning." 2020 International Symposium on Electronics and Telecommunications (ISETC). IEEE, 2020.

[27] Mercioni, Marina Adriana, and Stefan Holban. "Developing Novel Activation Functions in Time Series Anomaly Detection with LSTM Autoencoder." 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI). IEEE, 2021.

[28] Mercioni, Marina Adriana, and Stefan Holban. "Soft-clipping swish: A novel activation function for deep learning." 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI). IEEE, 2021.

[29] Matthew Mithra Noel et al., “Biologically Inspired Oscillating Activation Functions Can Bridge the Performance Gap between Biological and Artificial Neurons”, 2022.

[30] Dai, Sihui et al. “Parameterizing Activation Functions for Adversarial Robustness.” 2022 IEEE Security and Privacy Workshops (SPW) (2021): 80-87.

[31] Biswas, Koushik et al. “SAU: Smooth activation function using convolution with approximate identities.” European Conference on Computer Vision (2021).

[32] Koushik Biswas1 et al., “ErfAct and Pserf: Non-monotonic Smooth Trainable Activation Functions”, doi.org/10.1609/aaai.v36i6.20557, 2022.

[33] Misra, Diganta. “Mish: A Self Regularized Non-Monotonic Activation Function.” British Machine Vision Conference (2020).

[34] Nag, Sayan and Mayukh Bhattacharyya. “Serf: Towards better training of deep neural networks using log-Softplus ERror activation Function.” 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2021): 5313-5322.

[35] Shazeer, Noam M.. “GLU Variants Improve Transformer.” ArXiv abs/2002.05202 (2020).

[36] Vincent Sitzmann et al., “Implicit Neural Representations with Periodic Activation Functions”, doi.org/10.48550/arXiv.2006.09661,2020.

[37] Parisi, Luca. “m-arcsinh: An Efficient and Reliable Function for SVM and MLP in scikit-learn.” ArXiv abs/2009.07530 (2020).

[38] Xinyu Liu et al., “TanhExp: A Smooth Activation Function with High Convergence Speed for Lightweight Neural Networks”, doi.org/10.48550/arXiv.2003.09855, 2020.

[39] Advait Vagerwal, “Deeper Learning with CoLU Activation”, doi.org/10.48550/arXiv.2112.12078, 2021.

[40] ABIEN FRED AGARAP, “DEEP LEARNING USING RECTIFIED LINEAR UNITS (RELU)”, DOI.ORG/10.48550/ARXIV.1803.08375, 2018.

[41] Diganta Misra, “Mish: A Self Regularized Non-Monotonic Activation Function”, doi.org/10.48550/arXiv.1908.08681, 2019.

[42] Madhu G et al., “NIPUNA: A Novel Optimizer Activation Function for Deep Neural Networks”. Axioms; 12(3):246, doi.org/10.3390/axioms12030246, 2023.

[43] Jena Dataset 2009-2020, https://www.bgc-jena.mpg.de/wetter/.