Bulletin of Electrical Engineering and Informatics
Vol. 13, No. 5, October 2024, pp. 3301~3308
ISSN: 2302-9285, DOI: 10.11591/eei.v13i5.7274

A comparative analysis of activation functions in neural networks: unveiling categories

Sara Bouraya, Abdessamad Belangour
Laboratory of Information Technology and Modeling, Faculty of Sciences Ben M'sik, Hassan II University, Casablanca, Morocco

Article Info

Article history:

Received Jul 27, 2023
Revised Mar 20, 2024
Accepted Mar 29, 2024

Keywords:

Activation function
Artificial neural networks
Deep learning
Neural networks
Relu
Sigmoid
Tanh

ABSTRACT

Activation functions (AFs) play a critical role in artificial neural networks, allowing for the modeling of complex, non-linear relationships in data. In this review paper, we provide an overview of the most commonly used AFs in deep learning. In this comparative study, we survey and compare the different AFs in deep learning and artificial neural networks. Our aim is to provide insights into the strengths and weaknesses of each AF and to provide guidance on the appropriate selection of AFs for different types of problems. We evaluate the most commonly used AFs, including sigmoid, tanh, rectified linear units (ReLUs) and its variants, exponential linear unit (ELU), and SoftMax. For each activation category, we discuss its properties, mathematical formulation (MF), and the benefits and drawbacks in terms of its ability to model complex, non-linear relationships in data. In conclusion, this comparative study provides a comprehensive overview of the properties and performance of different AFs, and serves as a valuable resource for researchers and practitioners in deep learning and artificial neural networks.

Corresponding Author:

Sara Bouraya
Laboratory of Information Technology and Modeling, Faculty of Sciences Ben M'sik, Hassan II University Casablanca, Morocco
Email: sarabouraya95@gmail.com

1. INTRODUCTION

A single-layer perceptron, also known as a single-layer feedforward neural network, is the simplest form of a neural network architecture. It consists of a single layer of artificial neurons, also called perceptrons or units, arranged in a linear fashion [1]. Each perceptron takes in a set of input values, applies weights to those inputs, and passes the weighted sum through an activation function (AF) to produce an output (see Figure 1).

The AF plays a crucial role in determining the output of each perceptron [2]. It introduces non-linearity to the model, allowing the perceptron to learn complex patterns and make decisions based on the input data. The choice of AF greatly influences the model's capabilities and performance [3].

AFs play a crucial role in neural networks [4], acting as the critical "non-linearity" element that introduces complexity and enables the network to learn complicated patterns and make predictions [5]. Over the years, researchers have proposed various AFs [3], each with its unique characteristics and applications. In this paper, we delve into a comprehensive comparison of different AFs, aiming to classify them into distinct categories based on their properties and explore their practical implications in neural network architectures.

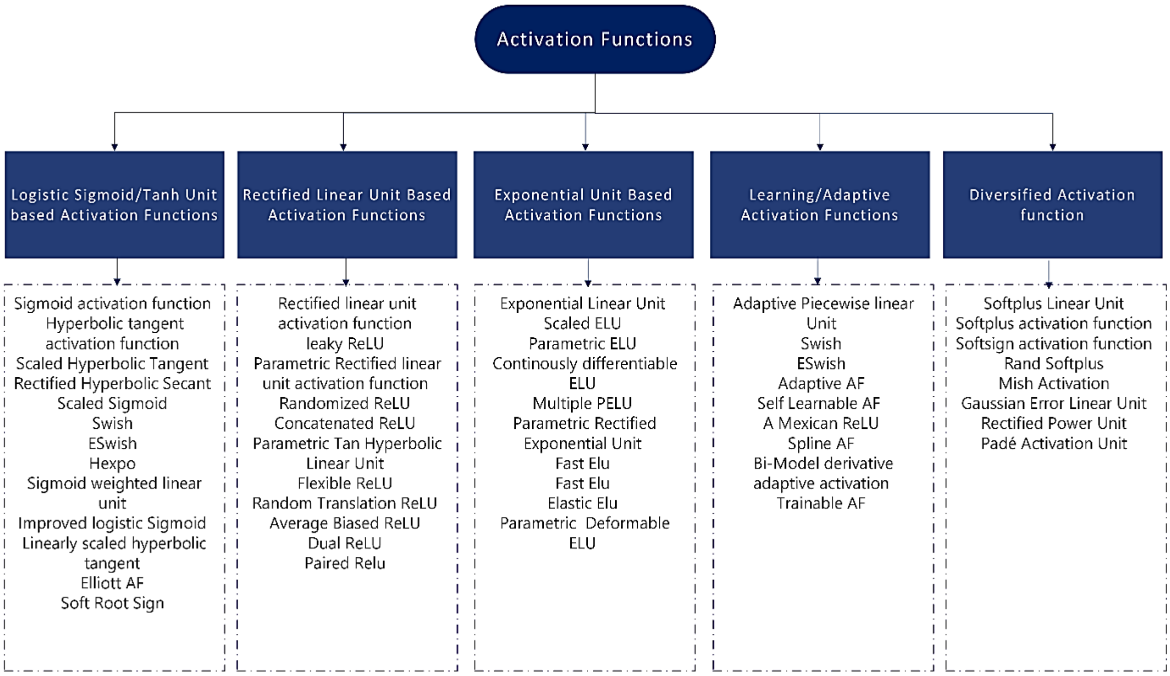

Figure 1. Taxonomy of deep learning AFs

2. BACKGROUND

The main role of an AF in a neural network is to introduce non-linearity to the network's computations [1]. Without AFs, a neural network would simply perform linear operations, resulting in a network that is equivalent to a single-layer perceptron. By applying non-linear AFs, neural networks become capable of modeling complex, and non-linear relationships between input and output.

Logistic sigmoid/tanh unit-based AFs: logistic sigmoid and hyperbolic tangent (tanh) functions are among the earliest and most widely used AFs in deep learning. These functions belong to the unit-based category and are characterized by their S-shaped curves. The logistic sigmoid function maps the input to a range between 0 and 1, while the tanh function maps it to a range between -1 and 1 [6]. These AFs are advantageous in binary classification problems, where the output is interpreted as probability. They enable the network to model non-linear relationships and provide a smooth (SM) transition between different statess.

Rectified linear unit (ReLU) based AFs: ReLU is a popular AF that has gained prominence in recent years. ReLU activation sets the output to zero for negative inputs (NI) and maintains a linear relationship for positive inputs. This characteristic allows ReLU to alleviate the vanishing gradient problem and accelerate training in deep neural networks. ReLU based AFs are computationally efficient and have been instrumental in the success of deep learning models across various domains [7].

Exponential unit based AFs: exponential unit-based AFs, such as exponential linear unit (ELU) and scaled exponential linear unit (SELU), have emerged as alternatives to traditional AFs. These functions introduce exponential behavior for NI, providing smoothness and enabling faster convergence. ELU and SELU have shown improvements in training stability and generalization performance, particularly in deep architectures. By incorporating adaptive components, these AFs facilitate more efficient gradient propagation and contribute to the model's robustness [8].

Learning/adaptive AFs: learning/adaptive AFs dynamically adjust their parameters based on the data, network architecture, or learning process. These AFs adapt to the characteristics of the input data, allowing the model to learn more complex representations. Examples of learning/adaptive AFs include spline AF [9] and adaptive piecewise linear unit (APL). These functions offer flexibility in capturing non-linear patterns and have been shown to enhance the model's expressive power [10].

Diversified AFs: in addition to the aforementioned categories, there exist diversified AFs that do not fit into the conventional groups but offer unique characteristics and advantages. These AFs, such as mish [11], and gaussian error linear units (GELUs) [12], have gained attention due to their empirical success in specific applications. They have demonstrated improved performance in terms of training speed, model accuracy, or robustness, and have sparked interest among researchers and practitioners for their potential to advance deep learning methodologies (Figure 1) [13].

3. COMPARISON

Our comparative study will be based on different criteria including the mathematical formulas as well as the monotonic behavior of each activation function without neglecting the abbreviation naming. Here os the list of the detailed criteria:

- AF mathematical formulas (MF): we will examine the MF actions in each category to understand how they transform the input into an output value.
- Parametric (PM): PM AFs are those that have additional parameters apart from the input. We will analyze if the AFs in each category fall into this category. PM AFs, such as PM rectified linear units (PReLU), allow the network to learn the optimal values for these additional parameters during training [14].
- Monotonic (MN): MN AFs preserve the order of inputs, meaning that if one input is greater than another, their outputs will also follow the same order. We will investigate if the AFs in each category exhibit MN behavior. Logistic sigmoid, tanh, and ReLU functions are MN, but this criterion may vary for other AFs [15].
- SM: SM AFs have continuous and differentiable derivatives. We will assess whether the AFs in each category possess this property [16].
- Bounded (BN): BN AFs are limited to a specific range of output values. We will determine if the AFs in each category are BN [17].
- Abbreviation (ABB): lastly, we will explore if there are commonly used ABBs for the AFs within each category.

By comparing these AF categories based on the above criteria, we can gain insights into their characteristics, strengths, and limitations, enabling us to make informed decisions when selecting the most appropriate AF for specific deep learning applications.

3.1. Logistic sigmoid/tanh unit based AFs

Table 1 displays when comparing the logistic sigmoid and tanh unit-based AFs based on various characteristics including PM, MN, SM, BN, ABB, and NI [18].

Table 1. List of logistic sigmoid/tanh unit based AFs

Logistic sigmoid/tanh unit based AFs						
AF	ABB	MF	PM	MN	SM	BN
Sigmoid	Sigmoid AF	$f(x) = \frac{1}{1 + e^{-x}}$	X	✓	✓	✓
Tanh	Hyperbolic tangent AF [19]	$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$	X	✓	✓	✓
sTanh	Scaled hyperbolic tangent [20]	$f(x) A \times \text{Tanh}(\text{B} \times \text{x})$	✓	✓	✓	✓
ReSech	Rectified hyperbolic secant [21]	$f(x) = x \times Sech(x)$	X	X	✓	✓
sSigmoid	Scaled sigmoid [22]	$f(x) = (4 \times Sigmoid(x) - 2)$	X	✓	✓	✓
Swish	Swish [23]	$f(x) = x \times Sigmoid(\beta \times x)$	✓	X	✓	X
ESwish	ESwish [24]	$f(x) = \beta \times x \times Sigmoid(x)$	✓	X	✓	X
Hexpo	Hexpo [25]	$f(x) = \begin{cases} -a\left(e^{-x/b} - 1\right), & x \geq 0 \\ c\left(e^{x/d} - 1\right), & x < 0 \end{cases}$	X	✓	✓	✓
SiLU	Sigmoid weighted linear unit [26]	$f(x) = x \times Sigmoid(x)$	X	X	✓	NI
ISigmoid	Improved logistic sigmoid [27]	$f(x) = \begin{cases} \alpha \times (x - \alpha) + Sigmoid(\alpha), & x \geq \alpha \\ Sigmoid(x), & -\alpha < x < \alpha \\ \alpha \times (x + \alpha) + Sigmoid(\alpha), & x \leq -\alpha \end{cases}$	X	✓	✓	X
LiSHT	Linearly scaled hyperbolic tangent [28]	$f(x) = x \times Tanh(x)$	X	X	✓	X
ELLiott	Elliott [29]	$f(x) = \frac{0.5 \times x}{1 + \lvert x \rvert} + 0.5$	X	✓	✓	✓
SRS	Soft root sign [30]	$f(x) = \frac{x}{\frac{x}{\alpha} + e^{-x/\beta}}$	X	✓	✓	✓

3.2. ReLU based AFs

Table 2 displays when comparing the ReLU based AFs based on various characteristics including PM, MN, SM, BN, ABB, and NI [31].

Table 2. ReLU based AFs

ReLU based AFs						
AF	ABB	MF	PM	MN	SM	BN
ReLU	ReLU AF [32]	$f(x) = \max(0, x)$	X	✓	X	NI
LReLU	Leaky ReLU [7]	$f(x) = \begin{cases} x, & x \geq 0 \\ 0.01 \times x, & x < 0 \end{cases}$	X	✓	X	X
PRelu	PM ReLU AF [33]	$f(x) = \begin{cases} \alpha x, & x < 0 \\ x, & x \geq 0 \end{cases}$	✓	✓	X	X
RReLU	Randomized ReLU [7]	$f(x) = \begin{cases} x, & x \geq 0 \\ R \times x, & x < 0 \end{cases}$	X	✓	X	X
CReLU	Concatenated ReLU [34]	$f(x) = \lvert ReLU(x), ReLU(-x) \rvert$	X	✓	X	NI
PTELU	PM tan hyperbolic linear unit [35]	$f(x) = \begin{cases} x, & x \geq 0 \\ \alpha \times Tanh(\beta \times x), & x < 0 \end{cases}$	✓	✓	✓	NI
FReLU	Flexible ReLU [36]	$f(x) = ReLU(x) + b$	✓	✓	X	NI
RTReLU	Random translation ReLU [37]	$f(x) = \begin{cases} x + \alpha, & x + \alpha > 0 \\ 0, & x + \alpha \leq 0 \end{cases}$	X	✓	X	NI
ABReLU	Average biased ReLU [38]	$f(x) = \begin{cases} x - \beta, & x - \beta \geq 0 \\ 0, & x - \beta < 0 \end{cases}$	X	✓	X	NI
DualReLU	Dual ReLU [39]	$f(a, b) = \max(0, a) - max(0, b)$	X	✓	X	X
PairedReLU	Paired ReLU [40]	$f(x) = [\max(s \times x - \theta, 0) - max(s_p \times x - \theta_p, 0)]$	✓	✓	X	X

3.3. Exponential unit based AFs

Table 3 displays when comparing the exponential unit based AFs based on various characteristics including PM, MN, SM, BN, ABB, and NI [41].

Table 3. Exponential unit based AFs

Exponential unit based AFs						
AF	ABB	MF	PM	MN	SM	BN
ELU	ELU [8]	$f(x) = \begin{cases} x, & x > 0 \\ \alpha \times (e^x - 1), & x \leq 0 \end{cases}$	✓	✓	✓	NI
SELU	Scaled ELU [42]	$f(x) = \lambda \times \begin{cases} x, & x > 0 \\ \alpha \times (e^x - 1), & x \leq 0 \end{cases}$	✓	✓	✓	NI
PELU	PM ELU [33]	$f(x) = \lambda \times \begin{cases} \frac{a}{b} \times x, & x \geq 0 \\ \alpha \times (e^{x/\alpha} - 1), & x < 0 \end{cases}$	✓	✓	X	NI
CELU	Continously differentiable ELU [43]	$f(x) = \begin{cases} x, & x \geq 0 \\ \alpha \times (e^{x/\alpha} - 1), & x < 0 \end{cases}$	✓	✓	X	NI
MPELU	Multiple PELU [44]	$f(x) = \begin{cases} x, & x > 0 \\ \alpha_c \times (e^{\beta_c \times x} - 1), & x \leq 0 \end{cases}$	✓	✓	X	NI
PREU	PM rectified exponential unit [45]	$f(x) = \begin{cases} \alpha \times x, & x > 0 \\ \alpha \times x \times e^{\beta \times x}, & x \leq 0 \end{cases}$	✓	X	✓	NI
FELU	Fast Elu [46]	$f(x) = \begin{cases} x, & x > 0 \\ \alpha \times (e^{x/\ln(2)} - 1), & x \leq 0 \end{cases}$	✓	✓	X	NI
EELU	Elastic Elu [47]	$f(x) = \begin{cases} k \times x, & x > 0 \\ \alpha \times (e^{\beta \times x} - 1), & x \leq 0 \end{cases}$	✓	✓	X	NI
PDELU	PM deformable ELU [48]	$f(x) = \begin{cases} x, & x > 0 \\ \alpha \times (1 + (1 - t) \times x)^{1/1-t} - 1, & x \leq 0 \end{cases}$	✓	✓	✓	NI

3.4. Learning/adaptive AFs

Table 4 displays when comparing the learning/adaptive AFs based on various characteristics including PM, MN, SM, BN, ABB, and NI [49].

Table 4. Learning/adaptive AFs

Learning/adaptive AFs						
AF	ABB	MF	PM	MN	SM	BN
APL	APL [50]	$f(x) = \max(0, x) + \sum_{s=1}^{s} \alpha_s \times \max(0, b_s - x)$	✓	X	X	X
Swish	Swish [23]	$f(x) = x \times Sigmoid(\beta \times x)$	✓	X	✓	X
ESwish	ESwish [24]	$f(x) = \beta \times x \times Sigmoid(x)$	✓	X	✓	X
AAF	Adaptive AF [51]	$f(x) = \sigma(\omega \times x) \times PRELU(x) + (1 - \sigma(\omega \times x)) \times PELU(x)$	✓	✓	X	X
SLAF	Self-learnable AF [52]	$f(x) = \sum_{i=0}^{N-1} \alpha_i \times x^i$	✓	X	✓	X
MeLU	Mexican ReLU [53]	$f(x) = PReLU(x) + \sum_{j=1}^{k} c_j \times \max(\lambda_j - \lvert x - \alpha_j \rvert, 0)$	✓	X	X	X
SAF	Spline AF [54]	$f(x) = \emptyset(s; q)$	✓	✓	✓	X
BDAA	Bi-model derivative adaptive activation [55]	$f(x) = {}^1/_2 \times ({}^1/_{1+e^{-x}} - {}^1/_{1+e^{-x-a}})$	✓	✓	✓	✓
TAF	Trainable AF [56]	$f(x) = \sqrt{(x-a)^2 + b^2}$	✓	X	✓	X

3.5. Diversified activation function

Table 5 displays when comparing the diversified AF on various characteristics including parametric, monotonic, smooth, bounded, abbreviation, and negative inputs.

Table 5. Diversified AF

Diversified AF		
AF	ABB	MF
SLU	Softplus linear unit [57]	$f(x) = \begin{cases} \alpha \times x, & x \leq 0 \\ \beta \times \log(e^x + 1) - \delta, & x > 0 \end{cases}$
Softplus	Softplus activation function [58]	$f(x) = \ln(1 + e^x)$
Softsign	Softsign activation function [59]	$f(x) = \frac{x}{1 + \lvert x \rvert}$
RSP	Rand softplus [60]	$f(x) = (1 - \rho) \times max(0, x) + \rho \times \log(1 + e^x)$
Mish	Mish activation [11]	$f(x) = x \times Tanh(Softplus(x))$
GELU	GELU [12]	$f(x) = x \times P(X \leq x)$
RePU	Rectified power unit [61]	$f(x) = \begin{cases} x^s, & x \geq 0 \\ 0, & x < 0 \end{cases}$
PAU	Pad activation unit [62]	$f(x) = \frac{P(x)}{Q(x)}$

4. RESULTS AND DISCUSSION

The logistic sigmoid and hyperbolic tangent functions, falling under the unit-based category, have been extensively employed in neural networks. These functions introduce non-linearity and transform the output values into a specific range, making them suitable for binary classification tasks and enabling probabilistic interpretations. ReLU based AFs have gained significant popularity due to their simplicity and ability to address the vanishing gradient problem. These functions set the output to zero for and NI, allowing the network to learn sparse representations and facilitating faster training. Exponential unit-based AFs, such as ELU and SELU [42], exhibit improved gradient propagation properties and hold the potential to enhance model performance. By introducing exponential behavior for NI, these functions offer smoothness and expedited convergence.

Learning/adaptive AFs dynamically adjust their parameters based on the data, network architecture, or learning process. Examples include PReLU [63] and APL [50]. These functions provide adaptability and flexibility, enabling the model to learn more intricate representations. Furthermore, we explored diversified AFs that do not fit into the conventional categories but offer unique properties and capabilities.

In conclusion, understanding the different categories of deep learning AFs is crucial for effectively designing and training neural network models. Each category offers distinct properties and benefits, and the selection of an appropriate AF depends on the specific task, model architecture, and performance requirements.

5. CONCLUSION

In conclusion, this paper aimed to provide a comprehensive survey of various categories of AFs used in deep learning. Five major categories were explored: logistic sigmoid/tanh unit-based AFs, ReLU based AFs, exponential unit-based AFs, learning/adaptive AFs, and diversified AFs. In conclusion, the selection of an appropriate AF is crucial in deep learning models, as it greatly influences the model's expressive power, convergence speed, and generalization performance. Researchers and practitioners need to understand the characteristics and properties of various AFs to effectively leverage them in deep learning applications. The presented survey provides valuable insights into the different categories of AFs, allowing practitioners to make informed decisions and advance the field of deep learning.

REFERENCES

[1] S. Sharma, S. Sharma, and A. Anidhya, "Understanding Activation Functions in Neural Networks," *International Journal of Engineering, Applied Sciences and Technology*, vol. 4, no. 12, pp. 310–316, 2020.

[2] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun, "Object detection networks on convolutional feature maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1476–1481, 2017, doi: 10.1109/TPAMI.2016.2601099.

[3] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92–108, 2022, doi: 10.1016/j.neucom.2022.06.111.

[4] I. Zafar, G. Tzanidou, R. Burton, N. Patel, and L. Araujo, *Hands-On Convolutional Neural Networks with TensorFlow*. Packt Publishing, 2018.

[5] A. D. Rasamoelina, F. Adjailia, and P. Sincak, "A Review of Activation Function for Artificial Neural Network," *Proceedings of the IEEE 18th World Symposium on Applied Machine Intelligence and Informatics*, pp. 281–286, 2020, doi: 10.1109/SAMI48414.2020.9108717.

[6] N. G. Timmons and A. Rice, "Approximating Activation Functions," *arXiv*, 2020, doi: 10.48550/arXiv.2001.06370.

[7] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network," *arXiv*, 2015, doi: arXiv.1505.00853.

[8] D. A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," *Proceedings of the 4th International Conference on Learning Representations Conference Track*, pp. 1–14, 2016.

[9] P. Bohra, J. Campos, H. Gupta, S. Aziznejad, and M. Unser, "Learning Activation Functions in Deep (Spline) Neural Networks," *IEEE Open Journal of Signal Processing*, vol. 1, pp. 295–309, 2020, doi: 10.1109/OJSP.2020.3039379.

[10] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, "Adaptive activation functions accelerate convergence in deep and physics-informed neural networks," *Journal of Computational Physics*, vol. 404, 2020, doi: 10.1016/j.jcp.2019.109136.

[11] D. Misra, "Mish: A Self Regularized Non-Monotonic Activation Function," *Proceedings of the 31st British Machine Vision Conference*, no. 1, 2020.

[12] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *arXiv*, 2016, doi: 10.48550/arXiv.1606.08415.

[13] S. Zhang, J. Lu, and H. Zhao, "Deep Network Approximation: Beyond ReLU to Diverse Activation Functions," *Neural Networks*, vol. 25, pp. 1–39, 2023.

[14] G. Bingham and R. Miikkulainen, "Discovering Parametric Activation Functions," *Neural Networks*, vol. 148, pp. 48–65, 2022, doi: 10.1016/j.neunet.2022.01.001.

[15] D. Runje and S. M. Shankaranarayana, "Constrained Monotonic Neural Networks," *International Conference on Machine Learning*, 2023, pp. 1-16.

[16] K. Biswas, S. Kumar, S. Banerjee, and A. K. Pandey, "Smooth Maximum Unit: Smooth Activation Function for Deep Networks using Smoothing Maximum Technique," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2022-June, pp. 784–793, 2022, doi: 10.1109/CVPR52688.2022.00087.

[17] S. S. Liew, M. Khalil-hani, and R. Bakhteri, "Bounded Activation Functions for Enhanced Training Stability of Deep Neural Networks on Visual Pattern Recognition Problems," *Neurocomputing*, 2016, doi: 10.1016/j.neucom.2016.08.037.

[18] L. Datta, "A Survey on Activation Functions and their relation with Xavier and He Normal Initialization," *arXiv*, 2020, doi: 10.48550/arXiv.2004.06632.

[19] M. Chandra, "Hardware Implementation of Hyperbolic Tangent Function using Catmull-Rom Spline Interpolation," *arXiv*, 2020, doi: 10.48550/arXiv.2007.13516.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.

[21] A. Nasser, S. Njikam, and H. Zhao, "A novel activation function for multilayer feed-forward neural networks," *Applied Intelligence*, 2016, doi: 10.1007/s10489-015-0744-0.

[22] B. Xu, R. Huang, and M. Li, "REVISE SATURATED ACTIVATION FUNCTIONS," in *Proceedings of the International Conference on Neural Information Processing*, Guangzhou, China, Nov. 2016, pp. 1–7.

[23] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi, "SWISH: A SELF-GATED ACTIVATION FUNCTION," *in Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015 - Workshop Track),* San Diego, CA, USA, May 2015, pp. 1–9.

[24] E. Alcaide, "E-swish : Adjusting Activations to Different Network Depths," *arXiv*, 2018, doi: 10.48550/arXiv.1801.07145.

[25] S. Kong and M. Takatsuka, "Hexpo: A vanishing-proof activation function," *2017 International Joint Conference on Neural Networks (IJCNN),* Anchorage, AK, USA, 2017, pp. 2562-2567, doi: 10.1109/IJCNN.2017.7966168.

[26] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, vol. 107, no. 2015, pp. 3–11, 2018, doi: 10.1016/j.neunet.2017.12.012.

[27] Y. Qin, X. Wang, and J. Zou, "The Optimized Deep Belief Networks with Improved Logistic Sigmoid Units and Their Application in Fault Diagnosis for Planetary Gearboxes of Wind Turbines," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 5, pp. 3814–3824, May 2019, doi: 10.1109/TIE.2018.2856205.

[28] S. K. Roy, S. Manna, S. R. Dubey, and B. B. Chaudhuri, "LiSHT: Non-parametric Linearly Scaled Hyperbolic Tangent Activation Function for Neural Networks," *Communications in Computer and Information Science*, vol. 1776 CCIS, pp. 462–476, 2023, doi: 10.1007/978-3-031-31407-0_35.

[29] D. L. Elliott, "A Better Activation Function for Artificial Neural Networks," *Institute for Systems Research Technical Reports*, University of Maryland, College Park, p. 4, Oct. 1993.

[30] Y. Zhou, D. Li, S. Huo, and S.-Y. Kung, "Soft-Root-Sign Activation Function," *arXiv,* 2020, doi: 10.48550/arXiv.2003.00547

[31] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," *arXiv*, 2018, doi: 10.48550/arXiv.1803.08375.

[32] G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," *Neural Networks*, vol. 24, no. 3, pp. 19–27, Jan. 2011.

[33] L. Trottier, P. Gigure, and B. Chaib-Draa, "Parametric exponential linear unit for deep convolutional neural networks," in *Proceedings of the 16th IEEE International Conference on Machine Learning and Applications (ICMLA),* Cancun, Mexico, 2017, vol. 2017, pp. 207–214, doi: 10.1109/ICMLA.2017.00038.

[34] W. Shang, K. Sohn, D. Almeida, and H. Lee, "Understanding and improving convolutional neural networks via concatenated rectified linear units," in *Proceedings of the 33rd International Conference on Machine Learning (ICML),* New York, NY, USA, Jun. 2016, vol. 5, pp. 3276–3284.

[35] A. Gupta and R. Duggal, "P-TELU: Parametric Tan Hyperbolic Linear Unit Activation for Deep Neural Networks," in *Proceedings-2017 IEEE International Conference on Computer Vision Workshops*, Oct. 2017, vol. 2018-Janua, pp. 974–978, doi: 10.1109/ICCVW.2017.119.

[36] S. Qiu, X. Xu and B. Cai, "FReLU: Flexible Rectified Linear Units for Improving Convolutional Neural Networks," *2018 24th International Conference on Pattern Recognition (ICPR),* Beijing, China, 2018, pp. 1223-1228, doi: 10.1109/ICPR.2018.8546022.

[37] J. Cao, Y. Pang, X. Li, and J. Liang, "Randomly translational activation inspired by the input distributions of ReLU," *Neurocomputing*, vol. 275, pp. 859–868, Jan. 2018, doi: 10.1016/j.neucom.2017.09.031.

[38] S. R. Dubey and S. Chakraborty, "Average biased ReLU based CNN descriptor for improved face retrieval," *Multimedia Tools and Applications,* vol. 80, pp. 23181–23206, 2021, doi: 10.1007/s11042-020-10269-x.

[39] F. Godin, J. Degrave, J. Dambre, and W. De Neve, "Dual Rectified Linear Units (DReLUs): A replacement for tanh activation functions in Quasi-Recurrent Neural Networks," *Pattern Recognition Letters,* vol. 116, pp. 8–14, 2018, doi: 10.1016/j.patrec.2018.09.006.

[40] Z. Tang, L. Luo, H. Peng, and S. Li, "A Joint Residual Network with Paired ReLUs activation for Image Super-Resolution," *Neurocomputing*, 2017, doi: 10.1016/j.neucom.2017.07.061.

[41] V. Pandey, "Overcoming Overfitting and Large Weight Update Problem in Linear Rectifiers: Thresholded Exponential Rectified Linear Units," *Neural Networks*, vol. 142, pp. 178–188, Jul. 2021, doi: 10.1016/j.neunet.2021.03.002.

[42] G. Klambauer, T. Unterthiner, and A. Mayr, "Self-Normalizing Neural Networks," *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA. 2017, pp. 1-10.

[43] J. T. Barron, "Continuously Differentiable Exponential Linear Units," *arXiv,* 2017, doi: 10.48550/arXiv.1704.07483.

[44] Y. Li, C. Fan, Y. Li, Q. Wu, and Y. Ming, "Improving Deep Neural Network with Multiple Parametric Exponential Linear Units," *Neural Networks*, vol. 133, pp. 263–270, Nov. 2020, doi: 10.1016/j.neunet.2020.08.012.

[45] Y. Ying, J. Su, P. Shan, L. Miao, X. Wang, and S. Peng, "Rectified Exponential Units for Convolutional Neural Networks," *IEEE Access*, vol. 7, pp. 101633–101640, 2019, doi: 10.1109/ACCESS.2019.2928442.

[46] Z. Qiumei, T. Dan, and W. Fenghua, "Improved Convolutional Neural Network Based on Fast Exponentially Linear Unit Activation Function," *IEEE Access*, vol. PP, no. 61305008, p. 1, 2019, doi: 10.1109/ACCESS.2019.2948112.

[47] D. Kim, J. Kim, and J. Kim, "Elastic exponential linear units for convolutional neural networks," *Neurocomputing*, vol. 406, pp. 253–266, 2020, doi: 10.1016/j.neucom.2020.03.051.

[48] Q. Cheng, H. L. Li, Q. Wu, L. Ma, and K. N. Ngan, "Parametric Deformable Exponential Linear Units for deep neural networks," *Neural Networks*, vol. 125, pp. 281–289, 2020, doi: 10.1016/j.neunet.2020.02.012.

[49] A. Rajanand and P. Singh, "ErfReLU : Adaptive Activation Function for Deep Neural Network," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 3, pp. 643–662, Mar. 2021, doi: 10.1007/s13042-021-01276-x.

[50] J. Inturrisi, S. Y. Khoo, A. Kouzani, and R. Pagliarella, "Piecewise Linear Units Improve Deep Neural Networks," *Neurocomputing*, vol. 329, pp. 317–325, Dec. 2018, doi: 10.1016/j.neucom.2018.12.010.

[51] S. Qian, H. Liu, C. Liu, S. Wu, and H. S. Wong, "Adaptive activation functions in convolutional neural networks," *Neurocomputing*, vol. 272, pp. 204–212, Jan. 2018, doi: 10.1016/j.neucom.2017.06.070.

[52] M. Goyal, R. Goyal, and B. Lall, "Learning Activation Functions: A new paradigm for understanding Neural Networks," *Neurocomputing*, vol. 335, pp. 195–214, Apr. 2019, doi: 10.1016/j.neucom.2018.10.084.

[53] G. Maguolo, L. Nanni, and S. Ghidoni, "Ensemble of Convolutional Neural Networks Trained with Different Activation Functions," in *Proceedings of the 17th International Conference on Image Analysis and Processing*, Catania, Italy, Sep. 2019, pp. 471–481.

[54] S. Scardapane, M. Scarpiniti, D. Comminiello, and A. Uncini, "Learning activation functions from data using cubic spline interpolation," *Neural Networks,* vol. 129, pp. 194–202, Jul. 2020, doi: 10.1016/j.neunet.2020.04.020.

[55] A. Mishra, P. Chandra, U. Ghose, and S. S. Sodhi, "Bi-Modal Derivative Adaptive Activation Function Sigmoidal Feedforward Artificial Neural Networks," *Applied Soft Computing Journal*, vol. 62, pp. 460–476, Sep. 2017, doi: 10.1016/j.asoc.2017.09.002.

[56] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete, "A survey on modern trainable activation functions," *Expert Systems with Applications*, vol. 190, p. 115507, Feb. 2021, doi: 10.1016/j.eswa.2021.115507.

[57] H. Zhao, F. Liu, L. Li, and C. Luo, "A novel softplus linear unit for deep convolutional neural networks," *Applied Intelligence*, vol. 48, no. 7, pp. 1707–1720, 2018, doi: 10.1007/s10489-017-1028-7.

[58] H. Zheng, Z. Yang, W. Liu, J. Liang, and Y. Li, "Improving Deep Neural Networks Using Softplus Units," *2015 International Joint Conference on Neural Networks (IJCNN)*, Killarney, 2015, pp. 1-4, doi: 10.1109/IJCNN.2015.7280459.

[59] E. Pishchik, "Trainable Activations for Image Classification," *Preprints.Org*, ol. 2023, Jan. 2023, pp. 1–8, doi: 10.20944/preprints202301.0026.v1.

[60] Y. Chen, Y. Mai, J. Xiao, and L. Zhang, "Improving the antinoise ability of DNNs via a bio-inspired noise adaptive activation function rand softplus," *Neural Computation*, vol. 31, no. 6, pp. 1215–1233, 2019, doi: 10.1162/neco_a_01192.

[61] A. Abdeljawad and P. Grohs, "Integral representations of shallow neural network with Rectified Power Unit activation function," *Neural Networks*, vol. 155, pp. 536-550, 2022.

[62] A. Molina, P. Schramowski, and K. Kersting, "Padé Activation Units: End-To-End Learning of Flexible Activation Functions in Deep Networks," in *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, vol. 2, no. 2019, pp. 1–17, 2020.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, vol. 2015 Inter, pp. 1026–1034, 2015, doi: 10.1109/ICCV.2015.123.

BIOGRAPHIES OF AUTHORS

Sara Bouraya Ph.D. student at the Faculty of Science, Ben Msik, at the Hassan II University. She specializes in the field of computer vision, which involves developing algorithms and systems that enable computers to perceive and understand visual information. She can be contacted at email: sarabouraya95@gmail.com.

Abdessamad Belangour academic researcher faculty member at the Hassan II University Faculty of Science in Ben Msik, Morocco. He specializes in the fields of data science, big data engineering, and meta-modeling. He can be contacted at email: belangour@gmail.com.