



# $\alpha$ -SechSig and $\alpha$ -TanhSig: two novel non-monotonic activation functions

Cemil Közkurt<sup>1</sup> · Serhat Kiliçarslan<sup>2</sup> · Selçuk Baş<sup>3</sup> · Abdullah Elen<sup>2</sup>

Accepted: 14 September 2023 / Published online: 6 October 2023

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

## Abstract

The deep learning architectures' activation functions play a significant role in processing the data entering the network to provide the most appropriate output. Activation functions (AF) are created by taking into consideration aspects like avoiding model local minima and improving training efficiency. Negative weights and vanishing gradients are frequently taken into account by the AF suggested in the literature. Recently, a number of non-monotonic AF have increasingly replaced previous methods for improving convolutional neural network (CNN) performance. In this study, two novel non-linear non-monotonic activation functions,  $\alpha$ -SechSig and  $\alpha$ -TanhSig are proposed that can overcome the existing problems. The negative part of  $\alpha$ -SechSig and  $\alpha$ -TanhSig is non-monotonic and approaches zero as the negative input decreases, allowing the negative part to retain its sparsity while introducing negative activation values and non-zero derivative values. In experimental evaluations,  $\alpha$ -SechSig and  $\alpha$ -TanhSig activation functions were tested on MNIST, KMNIST, Svhn\_Cropped, STL-10, and CIFAR-10 datasets. In addition, better results were obtained than the non-monotonic Swish, Logish, Mish, Smish, and monotonic ReLU, SinLU, and LReLU AF known in the literature. Moreover, the best accuracy score for the  $\alpha$ SechSig and  $\alpha$ TanhSig activation functions was obtained with MNIST at 0.9959 and 0.9956, respectively.

**Keywords** Activation function · Deep learning ·  $\alpha$ SechSig and  $\alpha$ TanhSig activation functions

---

✉ Serhat Kiliçarslan  
skilicarslan@bandirma.edu.tr

Cemil Közkurt  
ckozkurt@bandirma.edu.tr

Selçuk Baş  
sbas@bandirma.edu.tr

Abdullah Elen  
aelen@bandirma.edu.tr

<sup>1</sup> Faculty of Engineering and Natural Sciences, Department of Computer Engineering, Department of Transportation Engineering, Bandirma Onyedi Eylul University, BandirmaBalikesir, Turkey

<sup>2</sup> Faculty of Engineering and Natural Sciences, Department of Software Engineering, Bandirma Onyedi Eylul University, BandirmaBalikesir, Turkey

<sup>3</sup> Bandirma Vocational School, Department of Accounting and Tax, Bandirma Onyedi Eylul University, BandirmaBalikesir, Turkey

## 1 Introduction

Deep learning can be widely used in different problems such as signal processing, classification, and anomaly detection. CNN architecture, one of the types of deep learning, consists of pooling, activation, normalization, convolution, dropout, full connection, and output layers. Activation function is one of the most important criteria for better neural network performance.

Activation functions have been created by taking into consideration properties including vanishing gradients, enhancing training performance, and avoiding model local minima (Kiliçarslan and Celik 2021). The introduction of new AF for deep learning architectures has contributed to the increased interest in artificial neural networks (Apicella et al. 2021). In addition, due to not being selected suitable AF for deep learning architectures, for example, learning may not occur, a vanishing gradient may occur, or the training process may be slow. Among the important features of the developed AF are non-linearity and derivatives. Because, it is necessary to calculate how much

the curve will change during the training and back-propagation of the model. The back-propagation algorithm is a structure that is continuously derived, and it is one of the important features that can be derivatives in the developed AF (Kiliçarslan and Celik 2021, 2022).

Non-linear AF have been developed as monotonic and non-monotonic. The monotonic AF such as Sigmoid, Tanh, and ReLU AF can be expressed. It is seen that these AF are widely used in deep learning architectures. However, the vanishing gradient problem is encountered in Sigmoid and Tanh activation functions. To overcome the vanishing gradient problem, it can be provided to the ReLU activation function, which is not easy to reach the saturated level. Negative weights are ignored in the ReLU activation function. Therefore, some information on the neural network structure is lost and prevents the desired performance from being achieved. Therefore, many AF with fixed and trainable parameters have been developed in the literature to overcome these problems (Kiliçarslan and Celik 2021; Maas et al. 2013)–(Scardapane et al. 2019). Non-linear non-monotonic Swish, Mish, and Logish AF have been developed (Ramachandran et al. 2017)–(Zhu et al. 2021). Deep learning architectures allow networks to work efficiently using non-monotonic AF instead of non-linear monotonic.

In this study, two novel non-linear non-monotonic parametric AF are proposed. The proposed functions are expressed as  $\alpha$ -*SechSig* and  $\alpha$ -*TanhSig*. The proposed AF can overcome existing problems by taking advantage of smooth AF such as sigmoid and tanh and piecewise AF such as ReLU and its derivatives. The proposed AF in the literature commonly overcome the vanishing gradient and negative weight problems. Proposed  $\alpha$ -*SechSig* and  $\alpha$ -*TanhSig* AF have a continuously differentiable structure like Swish, Mish, and Logish in the literature. In experimental evaluations, the proposed  $\alpha$ -*SechSig* and  $\alpha$ -*TanhSig* AF were tested on MNIST, KMNIST, Svhn\_Cropped, STL-10 and CIFAR-10 datasets. Thus, it has been seen that it gives better results than the non-monotonic Swish, Logish, Mish, and Smish and monotonic ReLU, SinLU, and LReLU AF in the literature. In addition, the proposed AF can achieve more efficient results on big data than other functions. Following is a summary of the article's main contributions:

- Two new non-monotonic AF  $\alpha$ -*SechSig* and  $\alpha$ -*TanhSig* are proposed.
- The proposed AFs outperform the ReLU, SinLU, Swish, Logish, Mish, and LReLU.
- Proposed functions overcome vanishing gradient, conservation of negative weights.
- Convergence speed of the proposed AF is better than the others.

The second section of this study discusses the examination of the literature, the methodology and materials employed in the third section, the proposed activation function in the fourth section, the experimental findings in the fifth section, and the conclusions drawn in the sixth section.

## 2 Literature review

One of the most significant areas of research for scientists is activation function since they increase the success rates of deep learning systems. It is clear from the literature that non-linear AF have been widely developed. In this section, monotonic and non-monotonic AF are presented.

The ReLU monotonic AF has been proposed to deal with vanishing gradient occurring in sigmoid and tanh AF (Nair and Hinton 2010). The ReLU activation function (Eq. 1) overcomes the problem of vanishing gradient and causes the problem of ignoring negative weights during training. Therefore, the LReLU (Eq. 2) is proposed to participate in training at negative weights. (Maas et al. 2013). Moreover, the PReLU (Eq. 3) is proposed using the trainable slope parameter instead of the fixed slope parameter (He et al. 2015). As an alternative to the PReLU, the ELU (Eq. 4) was proposed to solve the problems of vanishing gradient and ignoring negative weights (Clevert et al. 2016). The SELU (Eq. 5) has been proposed to improve the training performance of the ELU activation function (Klambauer et al. 2017). In addition, the PELU (Eq. 6) activation function is developed by adding the trainable parameter for the positive input of the ELU activation function (Trottier et al. 2018). The RSigELU (Eq. 7) was developed to solve the problems of vanishing gradient and ignoring negative weights (Kiliçarslan and Celik2021). They reported that the proposed RSigELU can work actively in three regions positive, negative, and linear. The Sinu-sigmoidal Linear Unit (SinLU) (Eq. 8) was inspired by the sine wave and developed to overcome the problem found in the sigmoid activation function and to achieve better performance (Paul et al. 2022). The  $\alpha$  and  $\beta$  given in the equations are expressed as slope parameters in linear activation functions.  $x$  given in the equations is expressed as input parameter.

$$f_{ReLU}(x) = \max(x, 0) = \begin{cases} 0, & x < 0 \\ x, & \text{otherwise,} \end{cases} \quad (1)$$

$$f_{LReLU}(x) = \max(x, 0.01x) = \begin{cases} 0.01x, & x < 0 \\ x, & \text{otherwise,} \end{cases} \quad (2)$$

$$f_{PReLU}(x, \alpha) = \max(x, \alpha x) = \begin{cases} \alpha x, & x < 0 \\ x, & \text{otherwise,} \end{cases} \quad (3) \quad f_{Smish}(x) = x \tanh \left[ \ln \left( 1 + \frac{1}{1 + e^{-x}} \right) \right], \quad (12)$$

$$f_{ELU}(x, \alpha) = \begin{cases} \alpha(e^x - 1), & x < 0 \\ x, & \text{otherwise,} \end{cases} \quad (4) \quad f_{SAAF}(x, \alpha, \beta) = x \left( \frac{x}{\alpha} + e^{-x/\beta} \right)^{-1}. \quad (13)$$

$$f_{SELU}(x, \alpha, \beta) = \beta \begin{cases} \alpha(e^x - \alpha), & x < 0 \\ x, & \text{otherwise,} \end{cases} \quad (5)$$

$$f_{PELU}(x, \alpha, \beta) = \begin{cases} \alpha(e^{x/\beta} - 1), & x < 0 \\ \alpha x / \beta, & \text{otherwise,} \end{cases} \quad (6)$$

$$f_{RSigELU}(x, \alpha) = \begin{cases} x + \alpha x(1 + e^{-x})^{-1}, & 1 < x < \infty \\ x, & 0 \leq x \leq 1 \\ \alpha(e^x - 1), & \text{otherwise,} \end{cases} \quad (7)$$

$$f_{SinLU}(x, \alpha, \beta) = (x + \alpha \sin(\beta x)) \frac{1}{1 + e^{-x}}. \quad (8)$$

Second, a number of AF such as non-monotonic GELU (Hendrycks and Gimpel 2020), Swish (Ramachandran et al. 2017), Mish (Misra 2020), Logish (Zhu et al. 2021), Smish (Wang et al. 2022), Rectified Exponential Unit (REU) (Ying et al. 2019), SupEx (Kiliçarslan et al. 2023) have been proposed in the literature. Thanks to these proposed activation functions, it is seen that deep learning architectures perform better than monotonic activation functions. Swish (Eq. 9), one of the AF suggested in the literature, has been developed to achieve good performance in big data and multidimensional deep neural networks (Ramachandran et al. 2017). The Logish (Eq. 10) is proposed to achieve better performance than swish (Zhu et al. 2021). The Mish (Eq. 11), which works similarly to the Swish activation function, was developed by Misra et al. (2021) (Misra 2020). The Mish activation function may work better than the Swish function in terms of accuracy and generalization performance. The Swish, Mish, and Logish AF allow negative weights to participate in the training phase, thanks to a slight negative margin for their negative values, rather than a strict zero limit in the ReLU. In addition, since the Swish, Mish, Logish AF guarantee a better gradient optimization, they allow obtaining high-performance results in experimental evaluations. The Smish (Eq. 12) was proposed by Wang et al. (2022). The Smish is inspired by the Logish function. The SAAF (Eq. 13) activation function was developed to overcome the existing problems found in the sigmoid and ReLU (Zhou et al. 2021).

$$f_{Swish}(x) = x \frac{1}{1 + e^{-x}}, \quad (9)$$

$$f_{Logish}(x) = x \ln \left( 1 + \frac{1}{1 + e^{-x}} \right), \quad (10)$$

$$f_{Mish}(x) = x \tanh[\ln(1 + e^x)], \quad (11)$$

When the AF in the literature are examined, the death of negative weights and vanishing gradient are commonly concerned. While the above-mentioned AF can produce successful results in some deep learning architectures, it is observed that they cannot achieve good results in the architecture we used in our study. Therefore, two novel AF are proposed in our study and existing problems can be overcome. In addition, it has been observed that it gives better results in classification accuracy thanks to the proposed activation functions. Behaviors of proposed and other AF in the literature are shown in Fig. 1.

It was also a non-monotonically derivable function with bounded bottom and unbounded top that was proposed as an activation function for *SechSig* and *TanhSig*. Since the curve was smooth almost everywhere, more information could enter the neural network, improving accuracy and generalization performance. Among the features that should be in non-monotonic activation functions, non-linearity, almost ubiquitous differentiability, nearly linear matching, few parameters, and not very complex computation. The proposed  $\alpha$ -*SechSig* and  $\alpha$ -*TanhSig* AF incorporate the relevant features. As seen in Fig. 1b, the proposed functions do not pass through the origin with the change of alpha and so generate values in the negative region. Functions that generate values in the negative region overcome the problem of bias of network layers due to the above-zero average value known in ReLU (Gironés et al. 2005). In this context, all non-piecewise functions in Fig. 1b except Sigmoid and Softplus pass through the origin.

### 3 Proposed activation function

#### 3.1 Creating functions

Activation functions are designed as a piecewise or single equation to produce different outputs for the positive and negative signs of the input value. Piecewise AF are generally based on RELU, and those given between Eqs. (1) and (7) are examples of piecewise functions. Other function designs may consist of a single function such as Sigmoid, Tanh, or may be in the form of an input value or an algebraic combination or composition of more than one function. The combinations of the input value and the function are  $f(x) = x \cdot g(x)$  and have the Swish activation function structure. There is also the Logish function  $f(x) = x * g(h(1 + r(x)))$  which uses both compositional and

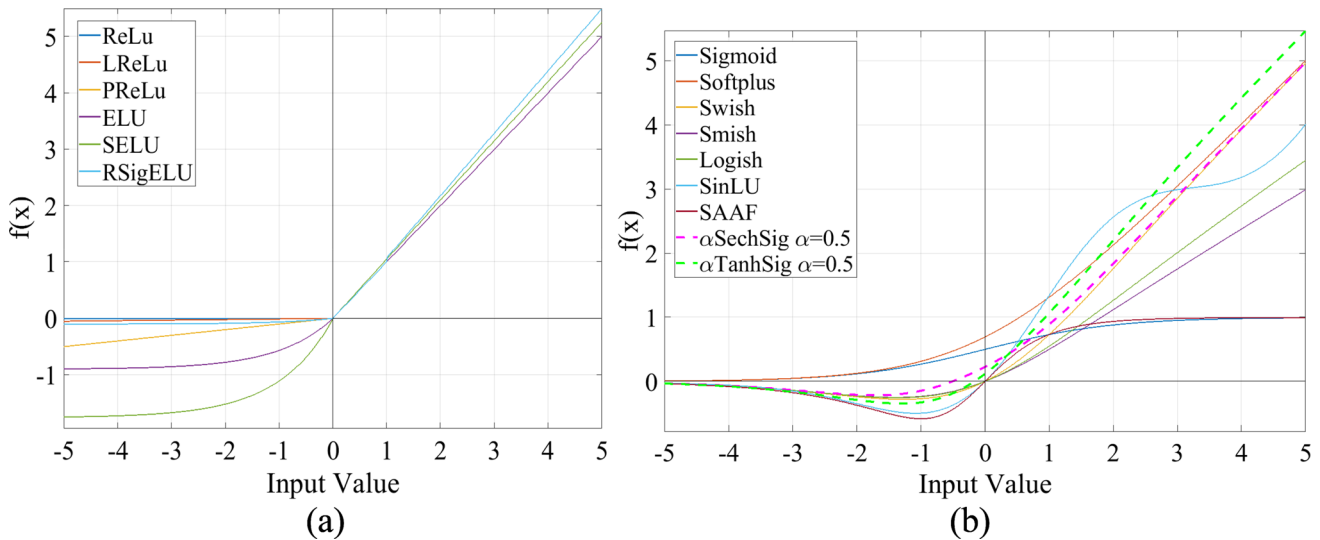


Fig. 1 Behaviors of proposed and mostly used activation functions, (a) piecewise (b) non-piecewise functions

algebraic combinations in a complex way. An example of a more regularly defined function as  $f(x) = (x + g(x)) * h(x)$  is SinLU. The functions proposed in this study are  $f(x) = (x + g(x)) * h(x)$ . The basic states of the proposed functions are given in Eqs. (14) and (15).

$$f_{SechSig}(x) = (x + \text{sech}(x)) \frac{1}{1 + e^{-x}}, \tag{14}$$

$$f_{TanhSig}(x) = (x + \tanh(x)) \frac{1}{1 + e^{-x}}. \tag{15}$$

The basic forms and derivatives of the proposed functions are given in Fig. 2.

An  $\alpha$  parameter is used in the activation function to obtain modified gradients that can be changed with used  $\alpha$  parameter. With this  $\alpha$  parameter, modified gradients can be obtained in the functions around the  $x = 0$  line, that is, around the activation value axis, and in the positive region of the input value in the second function. If we update the

form of the proposed functions as  $f(x) = (x + \alpha g(x + \alpha))h(x)$ , the functions with  $\alpha$  parameters are given in Eqs. (16) and (17). Functions with  $\alpha$  parameters are used in the study to control the regional properties. Although  $\alpha$  is defined as the slope parameter in linear-based activation functions, it has the effect of changing the lower limit and the  $x$ -axis cutoff point in addition to the slope in the proposed functions.  $x$  given in equations is expressed as input parameter.

$$f_{\alpha SechSig}(x, \alpha) = (x + \alpha \text{sech}(x + \alpha)) \frac{1}{1 + e^{-x}}, \tag{16}$$

$$f_{\alpha TanhSig}(x, \alpha) = (x + \alpha \tanh(x + \alpha)) \frac{1}{1 + e^{-x}}. \tag{17}$$

Equations (18) and (19) are obtained by derivation of proposed AF with respect to  $x$ .

$$\frac{df_{\alpha SechSig}(x, \alpha)}{dx} = \frac{e^{-x}(x + \text{sech}(x + \alpha))}{(1 + e^{-x})^2} + \frac{1 - \alpha \tanh(x + \alpha)\text{sech}(x + \alpha)}{1 + e^{-x}}, \tag{18}$$

$$\frac{df_{\alpha TanhSig}(x, \alpha)}{dx} = \frac{e^{-x}(x + \tanh(x + \alpha))}{(1 + e^{-x})^2} + \frac{1 - \alpha \text{sech}^2(x + \alpha)}{1 + e^{-x}}. \tag{19}$$

The graphs and types of the proposed functions obtained in the intervals  $\alpha = (0, 1)$  and  $x = (-5, 5)$  are given in Figs. 3 and 4, respectively. Both functions are Swish functions for  $\alpha = 0$ .

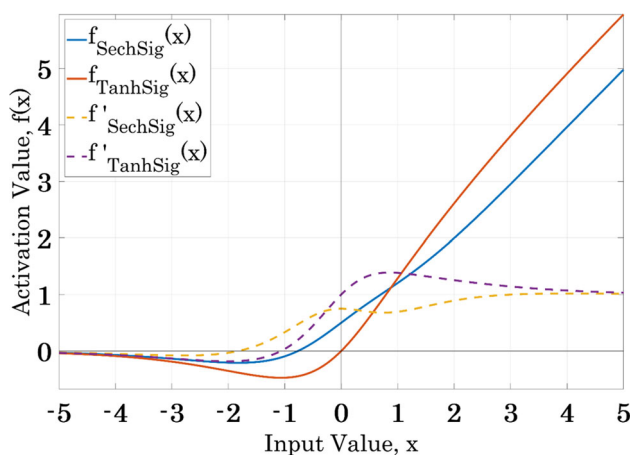
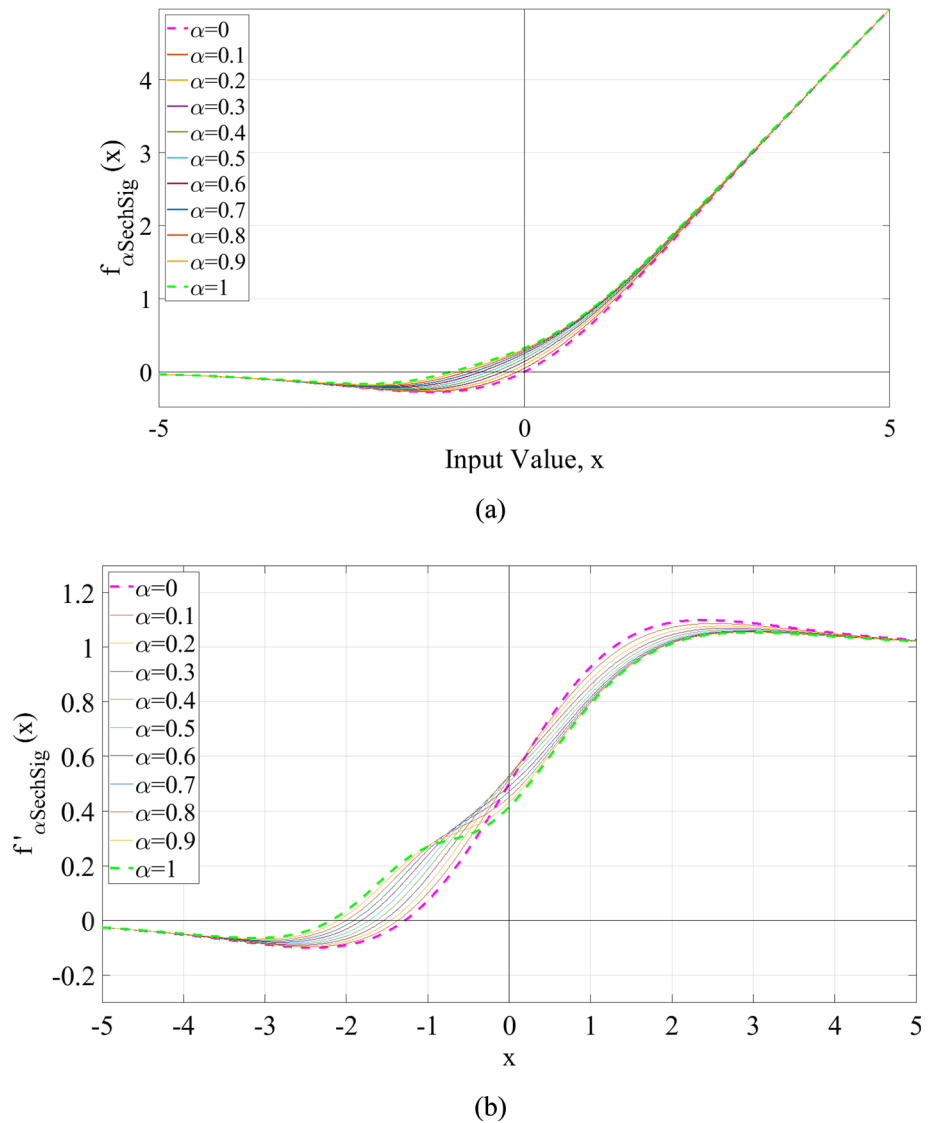


Fig. 2 Graphs of proposed base functions and derivatives

**Fig. 3** **a** Shape, **b** first derivative of  $\alpha$ -SechSig activation function



### 3.2 Characteristics of functions

In terms of two important features desired in activation functions, the proposed AF do not have an upper limit and a lower limit. The absence of an upper limit prevents saturation caused by slopes close to zero. Having a lower bound creates a strong regularization effect. The proposed functions are monotonic, increasing and decreasing to the left and right of their lower bounds.

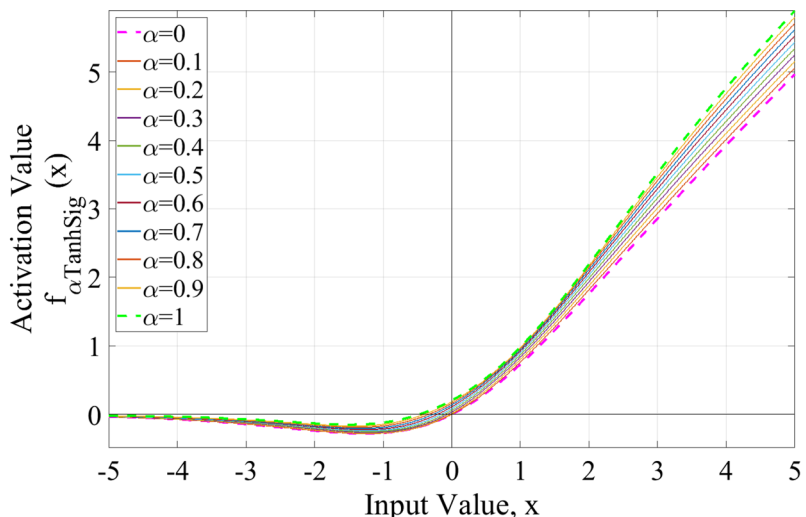
Let's explain this situation on the  $\alpha$ -SechSig function for the value of  $\alpha = 0.5$ . According to the Rolle mean value theorem. Given in Eqs. (20) and (21)

$$f_{\alpha SechSig}(-2) = f_{\alpha SechSig}(-1.4738) = -0.21307, \alpha = 0.5, \tag{20}$$

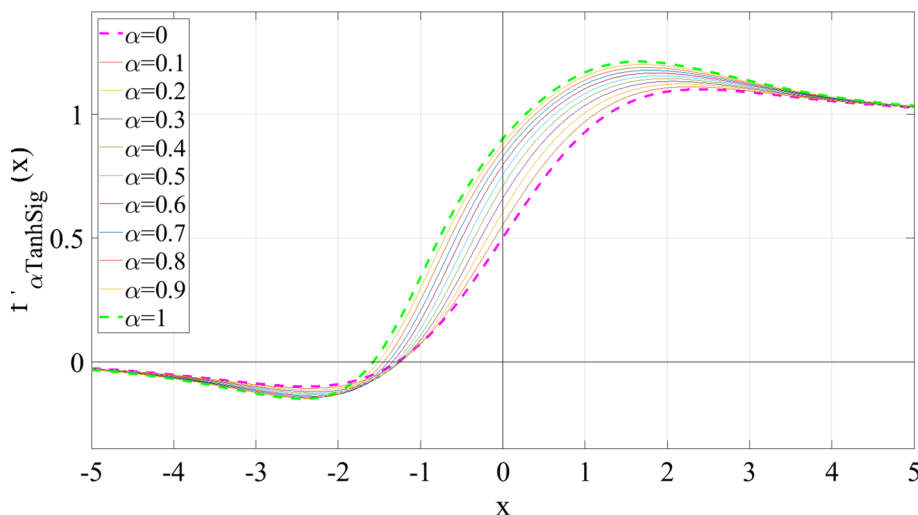
$$\frac{df_{\alpha SechSig}(x, \alpha)}{dx} = 0, x = -1.71967, x \in [-2, -1.71967]. \tag{21}$$

Minimum value of function is  $f_{\alpha SechSig}(-1.71967) = -0.219976$ . The function decreases monotonically by taking  $(0, -0.219976]$  values in the range of input values  $(-\infty, -1.71967]$ . The function increases monotonically by taking  $[-0.219976, +\infty)$  values in the range of input values  $[-1.71967, +\infty)$ . So for  $\alpha = 0.5$  the function  $f_{\alpha SechSig}(x)$  is non-monotonic in the range  $(-\infty, +\infty)$  and the value range of the function is  $[-0.219976, +\infty)$ . The above rules also apply to the other values of the  $\alpha$  parameter and the  $\alpha$ -TanhSig function. The approximate input value ranges and approximate value ranges of both functions are given in Tables 1 and 2.

**Fig. 4** **a** Shapes, **b** first derivatives of  $\alpha$ -TanhSig activation function



(a)



(b)

To provide back-propagation in convolutional neural networks, the AF must be differentiable over the entire range. The proposed  $\alpha$ -SechSig and  $\alpha$ -TanhSig functions are differentiable over the entire range. They are non-linear functions because their derivatives are not constant. Since the proposed functions do not satisfy the conditions for odd or even, they are neither odd nor even and asymmetric (Baş 2018; Korpınar and Baş 2019).

Tables 1 and 2 show regions of increasing monotone and decreasing monotone. Thus, it is shown that the proposed activation functions are non-monotonic. It is known that state-of-the-art activation functions are non-monotonic and are effective in optimizing the network with their modifiable shapes. The tables also offer a new activation function analysis approach with the values and order they contain.

## 4 Materials and methods

### 4.1 Datasets

In the study, MNIST, Cifar-10, KMNIST, SVHN-cropped, and STL-10 benchmark datasets were obtained from the tensor-flow catalog web page and experimental evaluations were carried out.

70,000 handwritten digits in the MNIST dataset range from 0 to 9 (Lecun et al. 1998). MNIST dataset, 60,000 of them were used for training, while 10,000 of them were used for testing. The dataset has 10 classes, each representing  $28 \times 28$  pixel numbers from 0 to 9.

The total number of color pictures in the CIFAR-10 collection is 60,000 (Krizhevsky et al. 2012). 10,000



**Table 1** Non-monotonicity table of  $\alpha$ -SechSig activation function

$\alpha$ -	Monotonic Interval		
	Decreasing	Increasing	Range
0.0	$(-\infty, -1.279]$	$[-1.279, +\infty)$	$[-0.279, +\infty)$
0.1	$(-\infty, -1.364]$	$[-1.364, +\infty)$	$[-0.267, +\infty)$
0.2	$(-\infty, -1.451]$	$[-1.451, +\infty)$	$[-0.255, +\infty)$
0.3	$(-\infty, -1.539]$	$[-1.539, +\infty)$	$[-0.244, +\infty)$
0.4	$(-\infty, -1.629]$	$[-1.629, +\infty)$	$[-0.232, +\infty)$
0.5	$(-\infty, -1.720]$	$[-1.720, +\infty)$	$[-0.220, +\infty)$
0.6	$(-\infty, -1.811]$	$[-1.811, +\infty)$	$[-0.208, +\infty)$
0.7	$(-\infty, -1.904]$	$[-1.904, +\infty)$	$[-0.197, +\infty)$
0.8	$(-\infty, -1.997]$	$[-1.997, +\infty)$	$[-0.186, +\infty)$
0.9	$(-\infty, -2.091]$	$[-2.091, +\infty)$	$[-0.175, +\infty)$
1.0	$(-\infty, -2.186]$	$[-2.186, +\infty)$	$[-0.164, +\infty)$

**Table 2** Non-monotonicity table of  $\alpha$ -TanhSig activation function

$\alpha$ -	Monotonic Interval		
	Decreasing	Increasing	Range
0.0	$(-\infty, -1.278]$	$[-1.278, +\infty)$	$[-0.278, +\infty)$
0.1	$(-\infty, -1.248]$	$[-1.248, +\infty)$	$[-0.297, +\infty)$
0.2	$(-\infty, -1.237]$	$[-1.237, +\infty)$	$[-0.313, +\infty)$
0.3	$(-\infty, -1.244]$	$[-1.244, +\infty)$	$[-0.328, +\infty)$
0.4	$(-\infty, -1.265]$	$[-1.265, +\infty)$	$[-0.340, +\infty)$
0.5	$(-\infty, -1.298]$	$[-1.298, +\infty)$	$[-0.350, +\infty)$
0.6	$(-\infty, -1.341]$	$[-1.341, +\infty)$	$[-0.356, +\infty)$
0.7	$(-\infty, -1.391]$	$[-1.391, +\infty)$	$[-0.361, +\infty)$
0.8	$(-\infty, -1.447]$	$[-1.447, +\infty)$	$[-0.362, +\infty)$
0.9	$(-\infty, -1.508]$	$[-1.508, +\infty)$	$[-0.362, +\infty)$
1.0	$(-\infty, -1.573]$	$[-1.573, +\infty)$	$[-0.359, +\infty)$

images of the CIFAR-10 dataset were used for testing, while 50,000 of them were used for training. The dataset has 10 classes and represents  $32 \times 32$ -pixel color images, each of which consists of different objects.

The Kuzushiji-MNIST (KMNIST) dataset consists of 70,000 images of Japanese Hiragama characters (Clanuwat et al. 2023). KMNIST dataset, of which 60,000 were used for training and 10,000 for testing. The dataset has 10 classes and each consists of a string of  $28 \times 28$  pixel Hiragama characters from the Japanese language.

The svhn\_cropped dataset consists of 600,000 color images of house numbers between 0 and 9 in Google Street view images (Netzer et al. 2011). The svhn\_cropped dataset consists of 73,257 training and 26,032 test data. The

class number of the dataset is 10 and consists of  $32 \times 32$ -pixel house door numbers obtained from real-world data.

The STL-10 dataset consists of color images inspired by the CIFAR-10 dataset (Coates et al. 2011). The STL-10 dataset consists of 8,000 training and 5,000 test data. The dataset has 10 classes and represents  $96 \times 96$ -pixel color images, each of which consists of different objects. Images were obtained from labeled samples on ImageNet.

## 4.2 Convolutional neural network (CNN)

Deep learning architectures are a subset of machine learning that are investigated and feature a multi-layered structure (LeCun et al. 2015). In deep learning, the most widely used model is convolutional neural networks (CNN). The CNN is a feedforward neural network designed to recognize patterns directly from image pixels (or other signals) by combining feature extraction and classification (Kiliçarslan and Celik 2021; LeCun et al. 2015). Convolutional layers, pooling layers, activation layers, dropout layers, fully connected layers, and classification layers at the output make up the CNN architecture (Lecun et al. 1998, 2015; Kiliçarslan et al. 2021; Adem and Kiliçarslan 2021; Adem et al. 2019; Pacal and Karaboga 2021). The CNN architecture used is shown in Fig. 5.

In Fig. 5, the CNN architecture consists of convolutional, activation, pooling, dropout, normalization, flatten, and fully connected layers (Elen 2022). In the proposed method, three convolutional layers, two fully concatenated layers and one output layer are used. The convolution layer allows us to achieve significant accuracy in the CNN architecture using a filter size of  $3 \times 3$ . The image obtained with the filter becomes the input image of the next layer. The proposed non-linear non-monotonic  $\alpha$ -SechSig and  $\alpha$ -TanhSig AF were used as the activation layer. The pooling layer is used to reduce the size of the resulting feature map and lower the computational cost. The filter size in the pooling layer was set to  $2 \times 2$ . The dropout ratio was set to 0.4 to prevent overfitting. After the pooling process, all layers are connected with two fully connected layers with 512 neurons, and the model is prepared for the classification process (Adem and Közkurt 2019). The Softmax layer is preferred to successfully perform the classification process (LeCun et al. 2015; Kiliçarslan et al. 2021; Adem and Kiliçarslan 2021). Adam optimizer was used in the experiments (Gorur et al. 2022).

## 4.3 Transfer learning methods

Transfer learning is the practice of applying features discovered while solving one problem to address new, related problems. For instance, features that were discovered for animal picture classification might be applied to categorize

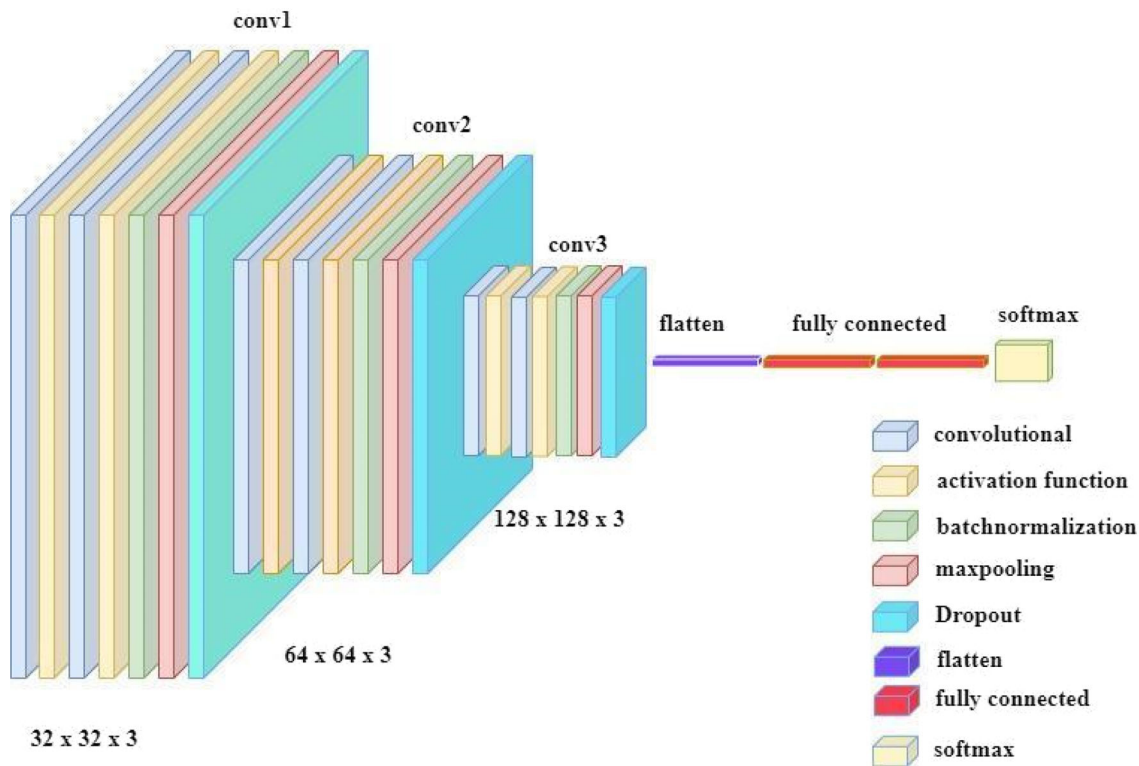


Fig. 5 CNN Architecture Structure

flowers. In other words, transfer learning is the process of learning a new activity more effectively by applying what has already been learned about a related task. Transfer learning is a viable alternative to starting from scratch when we only have a few data sets to train a model on.

ResNet50 is a variation of the ResNet architecture, a deep residual neural network. ResNet’s the main idea behind it is that by introducing shortcut links, also known as jump links, deep is to alleviate the gradient problem that is absent in neural networks. These connections allow the network to intermediate the input allows direct transmission to a deeper layer, bypassing layers.

### 5 Experimental results

In the experimental studies, five datasets, namely MNIST, KMNIST, CIFAR-10, STL-10, and SVHN-Cropped, were used to evaluate the performance of the proposed activation functions.

The  $\alpha$  slope parameter of the proposed AF was analyzed with ten different values equally partitioned between 0.1 and 1. In the experiments, Sigmoid, ReLU, LReLU, Swish, Mish, Smish, Logish and SinLU AF were used to compare the experimental results of the proposed activation functions. Experiments were performed with 50 epochs where the model converges enough to fully train the model. Also

selection of 50 epochs minimized the deviation in mean of result values.

#### 5.1 Convergence speed

Gironés et.al. state that the speed of convergence should be taken into consideration while choosing the derivative of activation function implementation (Gironés et al. 2005). So, the slope of the first derivative of the activation function affects the speed of the deep learning model. Derivative graphs of the activation functions used in the experiments and the functions we proposed are shown in Fig. 6.

To compare the convergence speeds of the proposed AF experimentally with other AF, the validated accuracy results for 50 epochs and 7 epochs of the experiments on the MNIST dataset are given in Fig. 6, and the valued loss results are given in Fig. 7.

When Figs. 6 and 7 are examined, it is observed that the proposed activation functions provide a high initial convergence rate in both accuracy and loss graphs compared to the activation functions used in the experiments.

In terms of convergence speed and ultimate accuracy,  $\alpha$ -SechSig and  $\alpha$ -TanhSig exceed Sigmoid, ReLU, LReLU, Swish, Mish, Smish, Logish, and SinLU AF. It showcases the quick parameter updates capabilities of  $\alpha$ -SechSig and  $\alpha$ -TanhSig and forces the network to more effectively



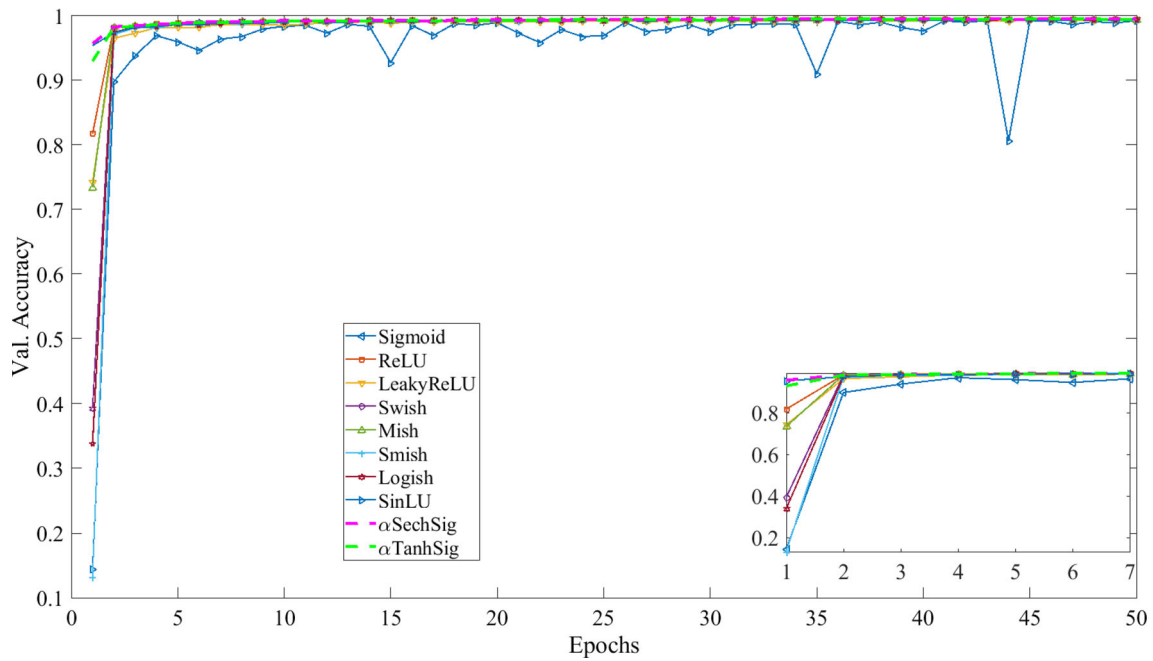


Fig. 6 Validated accuracy via 50 epochs and zoomed to 7 epochs on MNIST with different activation functions

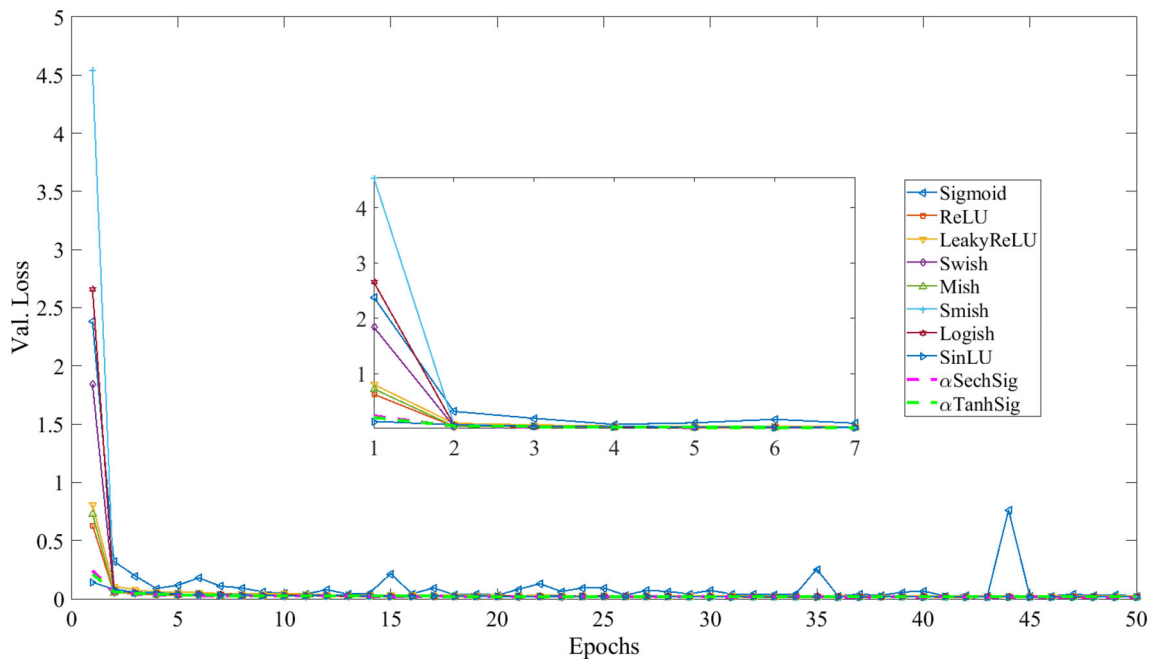


Fig. 7 Validated loss via 50 epochs and zoomed to 7 epochs on MNIST with different activation functions

match the dataset, resulting in high accuracy and minimal loss.

In Fig. 8, the first derivative graphs of the activation functions used in the experiments are given. The derivatives of the proposed activation functions do not converge to zero, thus overcoming the vanishing gradient problem. The derivative graph of the sigmoid function converges to zero. Activation functions whose derivatives do not

converge to zero and whose slopes are high have a high convergence speed.

### 5.2 Train and validation results

Classification experiments of mean scores of CNN, with proposed  $\alpha$ -SechSig and  $\alpha$ -TanhSig with varying  $\alpha$  parameters from 0.1 to 1, presented first for MNIST in

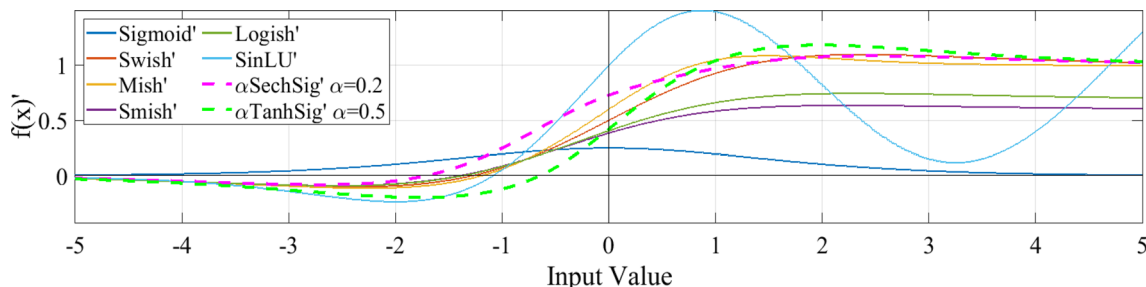


Fig. 8 Derivative graphs of activation functions that used in the experiments

Table 3 Mean scores of the  $\alpha$ -SechSig and  $\alpha$ -TanhSig on the MNIST dataset

	$\alpha$	Epoch	Train Loss	Train Acc	Val. Loss	Val. Acc
$\alpha$ -SechSig	0.10	50	0.0539	0.9837	0.0960	0.9780
	0.20	50	<b>0.0516</b>	<b>0.9844</b>	<b>0.0246</b>	<b>0.9925</b>
	0.30	50	0.0558	0.9830	0.0745	0.9795
	0.40	50	0.0545	0.9831	0.0945	0.9765
	0.50	50	0.0567	0.9830	0.0661	0.9805
	0.60	50	0.0566	0.9826	0.0440	0.9859
	0.70	50	0.0578	0.9821	0.1066	0.9734
	0.80	50	0.0572	0.9824	0.0693	0.9765
	0.90	50	0.0580	0.9821	0.1037	0.9742
	1.00	50	0.0580	0.9821	0.1203	0.9715
$\alpha$ -TanhSig	0.10	50	0.0565	0.9828	0.0275	0.9905
	0.20	50	0.0540	0.9837	0.0327	0.9892
	0.30	50	0.0540	0.9836	0.0387	0.9865
	0.40	50	<b>0.0521</b>	<b>0.9840</b>	0.0292	0.9900
	0.50	50	0.0527	0.9838	<b>0.0259</b>	<b>0.9916</b>
	0.60	50	0.0558	0.9829	0.0329	0.9888
	0.70	50	0.0538	0.9837	0.0374	0.9867
	0.80	50	0.0540	0.9834	0.0254	0.9915
	0.90	50	0.0551	0.9828	0.0296	0.9903
	1.00	50	0.0549	0.9830	0.0286	0.9905

Table 3, then for KMNIST in Table 4, CIFAR-10 in Table 5, STL-10 in Table 6, and SVHN-Cropped in Table 7, respectively.

On the MNIST dataset experiments,  $\alpha$ -SechSig received the greatest score of 0.9925 success rate and 0.0246 error rate with an  $\alpha$  of 0.2 and  $\alpha$ -TanhSig received the greatest score of 0.9916 success rate and 0.0259 error rate with an  $\alpha$  of 0.5.

On the KMNIST dataset experiments,  $\alpha$ -SechSig received the greatest score of 0.9485 success rate and 0.2031 error rate with an  $\alpha$  of 0.6 and  $\alpha$ -TanhSig received the greatest score of 0.9527 success rate and 0.1746 error rate with an  $\alpha$  of 0.5.

On the CIFAR-10 dataset experiments,  $\alpha$ -SechSig received the greatest score of 0.7088 success rate and 0.8452 error rate with an  $\alpha$  of 0.1 and  $\alpha$ -TanhSig received

the greatest score of 0.7105 success rate and 0.8519 error rate with an  $\alpha$  of 0.2.

On the STL-10 dataset experiments,  $\alpha$ -SechSig received the greatest score of 0.3315 success rate and 2.9525 error rate with an  $\alpha$  of 0.2 and  $\alpha$ -TanhSig received the greatest score of 0.3181 success rate and 2.8513 error rate with an  $\alpha$  of 0.2.

On the SVHN-Cropped dataset experiments,  $\alpha$ -SechSig received the greatest score of 0.9139 success rate and 0.3044 error rate with an  $\alpha$  of 0.4 and  $\alpha$ -TanhSig received the greatest score of 0.9164 success rate and 0.2962 error rate with an  $\alpha$  of 0.2.

Since the characteristics of each data set are different in the experiments carried out, high success rates are obtained in different slope parameters. In the proposed  $\alpha$ -SechSig and  $\alpha$ -TanhSig activation functions, it is seen that high success rates are obtained on all datasets with values

**Table 4** Mean scores of the  $\alpha$ -SechSig and  $\alpha$ -TanhSig on the Fashion KMNIST dataset

	$\alpha$	Epoch	Train Loss	Train Acc	Val. Loss	Val. Acc
$\alpha$ -SechSig	0.10	50	<b>0.0859</b>	<b>0.9734</b>	0.2098	0.9429
	0.20	50	0.0878	0.9728	0.2042	0.9449
	0.30	50	0.0905	0.9719	0.2105	0.9436
	0.40	50	0.0882	0.9726	0.2131	0.9422
	0.50	50	0.0917	0.9715	0.2129	0.9423
	0.60	50	0.0894	0.9720	<b>0.2031</b>	<b>0.9485</b>
	0.70	50	0.0937	0.9707	0.2271	0.9387
	0.80	50	0.0955	0.9700	0.2261	0.9380
	0.90	50	0.0983	0.9695	0.2297	0.9375
	1.00	50	0.0991	0.9689	0.2436	0.9342
$\alpha$ -TanhSig	0.10	50	0.0843	0.9740	0.1989	0.9478
	0.20	50	0.0849	0.9736	0.1949	0.9511
	0.30	50	0.0898	0.9722	0.1942	0.9467
	0.40	50	0.0906	0.9719	0.1975	0.9487
	0.50	50	<b>0.0832</b>	<b>0.9742</b>	<b>0.1746</b>	<b>0.9527</b>
	0.60	50	0.0874	0.9729	0.1957	0.9498
	0.70	50	0.0884	0.9725	0.1952	0.9498
	0.80	50	0.0911	0.9715	0.2021	0.9474
	0.90	50	0.0948	0.9704	0.2368	0.9394
	1.00	50	0.0887	0.9725	0.2024	0.9475

**Table 5** Mean scores of the  $\alpha$ -SechSig and  $\alpha$ -TanhSig on the CIFAR-10 dataset

	$\alpha$	Epoch	Train Loss	Train Acc	Val. Loss	Val. Acc
$\alpha$ -SechSig	0.10	50	<b>0.8198</b>	<b>0.7123</b>	<b>0.8452</b>	<b>0.7088</b>
	0.20	50	0.8228	0.7120	0.8721	0.7025
	0.30	50	0.8361	0.7071	0.9484	0.6787
	0.40	50	0.8304	0.7081	1.0862	0.6451
	0.50	50	0.8572	0.6985	1.1844	0.6173
	0.60	50	0.8627	0.6969	1.1789	0.6179
	0.70	50	0.8478	0.7033	1.3392	0.5887
	0.80	50	0.8792	0.6908	1.3362	0.5790
	0.90	50	0.8732	0.6942	1.4469	0.5629
	1.00	50	0.8912	0.6871	1.6254	0.5292
$\alpha$ -TanhSig	0.10	50	0.8255	0.7099	0.8982	0.6953
	0.20	50	0.8226	0.7106	<b>0.8519</b>	<b>0.7105</b>
	0.30	50	0.8213	0.7111	0.8636	0.7070
	0.40	50	<b>0.8062</b>	<b>0.7162</b>	0.8623	0.7101
	0.50	50	0.8139	0.7139	0.8678	0.7041
	0.60	50	0.8234	0.7102	0.8467	0.7058
	0.70	50	0.8250	0.7096	0.9576	0.6796
	0.80	50	0.9175	0.6777	1.0273	0.6494
	0.90	50	0.8151	0.7128	1.0041	0.6658
	1.00	50	0.8260	0.7098	1.0798	0.6453

between 0.1 and 0.5 by weight due to the structural characteristics of the proposed functions. Tables 8 and 9 present the  $\alpha$ -SechSig and  $\alpha$ -TanhSig AF proposed for the MNIST dataset, and the experimental evaluations of the Sigmoid, ReLU, LReLU, Swish, Mish, Smish, Logish, and

SinLU activation functions. The alpha coefficient used for LReLU was 0.01, and for SinLU alpha and beta were 1.

When Table 8 is examined, it is seen that the best training accuracy and training error values for all datasets are in the proposed  $\alpha$ -SechSig and  $\alpha$ -TanhSig activation

**Table 6** Mean scores of the  $\alpha$ -*SechSig* and  $\alpha$ -*TanhSig* on the STL-10 dataset

	$\alpha$	Epoch	Train Loss	Train Acc	Val. Loss	Val. Acc
<i><math>\alpha</math>-SechSig</i>	0.10	50	1.2714	0.6257	3.8879	0.2729
	0.20	50	<b>1.0397</b>	<b>0.6743</b>	<b>2.9525</b>	<b>0.3315</b>
	0.30	50	1.2446	0.6251	2.5543	0.3264
	0.40	50	1.2360	0.6336	3.1778	0.2919
	0.50	50	2.3139	0.1834	3.1804	0.1036
	0.60	50	3.2464	0.1461	2.9360	0.1013
	0.70	50	1.1722	0.6262	3.2679	0.2997
	0.80	50	1.2623	0.6123	2.7193	0.3082
	0.90	50	1.2890	0.6067	3.1988	0.2960
	1.00	50	1.2904	0.6022	4.8612	0.2340
<i><math>\alpha</math>-TanhSig</i>	0.10	50	1.4149	0.5644	2.8265	0.2890
	0.20	50	<b>1.1715</b>	<b>0.6437</b>	<b>2.8513</b>	<b>0.3181</b>
	0.30	50	1.3637	0.5999	2.8328	0.3039
	0.40	50	1.3408	0.6030	2.9187	0.3021
	0.50	50	1.2095	0.6320	3.0668	0.3016
	0.60	50	1.3709	0.6157	3.1890	0.3084
	0.70	50	1.3122	0.6168	3.2939	0.2990
	0.80	50	1.2590	0.6300	3.4874	0.2990
	0.90	50	1.6460	0.5277	4.2678	0.2441
	1.00	50	1.3928	0.5920	4.0202	0.2870

**Table 7** Mean scores of the  $\alpha$ -*SechSig* and  $\alpha$ -*TanhSig* on the SVHN-Cropped dataset

	$\alpha$	Epoch	Train Loss	Train Acc	Val. Loss	Val. Acc
<i><math>\alpha</math>-SechSig</i>	0.10	50	0.3751	0.8851	0.3126	0.9113
	0.20	50	0.3736	0.8855	0.3178	0.9107
	0.30	50	0.3677	0.8874	0.3138	0.9098
	0.40	50	<b>0.3628</b>	<b>0.8894</b>	<b>0.3044</b>	<b>0.9139</b>
	0.50	50	0.3871	0.8812	0.3272	0.9055
	0.60	50	0.3973	0.8773	0.3407	0.9020
	0.70	50	0.4038	0.8757	0.3474	0.8993
	0.80	50	0.4006	0.8766	0.3488	0.8978
	0.90	50	0.4038	0.8761	0.3445	0.9007
	1.00	50	0.3986	0.8776	0.3624	0.8960
<i><math>\alpha</math>-TanhSig</i>	0.10	50	0.3615	0.8898	0.3079	0.9136
	0.20	50	0.3581	0.8907	<b>0.2962</b>	<b>0.9164</b>
	0.30	50	0.3636	0.8886	0.3014	0.9149
	0.40	50	0.3580	0.8906	0.3013	0.9156
	0.50	50	0.3538	0.8919	0.2987	0.9162
	0.60	50	0.3734	0.8857	0.3232	0.9064
	0.70	50	0.3566	0.8911	0.2947	0.9161
	0.80	50	0.3599	0.8899	0.3057	0.9137
	0.90	50	<b>0.3535</b>	<b>0.8921</b>	0.3119	0.9102
	1.00	50	0.3631	0.8891	0.3259	0.9073

functions. In Table 8, it is seen that only the Mish activation function is the best training accuracy and training error value for the CIFAR-10 dataset, but there is a small difference of 0.08 as a percentage of accuracy.

During the training of the deep learning model, the  $\alpha$  values of the suggested AF according to the highest training accuracy scores are as follows:

**Table 8** Train loss and accuracy of the activation functions

Methods	MNIST		k-MNIST		CIFAR-10		STL-10		SHVN-C	
	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc
Sigmoid	0.0355	0.9887	0.0668	0.9789	0.7645	0.7313	1.1195	0.6036	0.3152	0.9045
ReLU	0.0210	0.9929	0.0289	0.9905	0.5599	0.8032	0.4480	0.8352	0.2067	0.9373
leakyReLU	0.0413	0.9873	0.0808	0.9739	0.8326	0.7149	0.9236	0.6710	0.4166	0.8833
Swish	0.0224	0.9926	0.0317	0.9898	0.5466	0.8062	0.4424	0.8488	0.2064	0.9380
Mish	0.0218	0.9930	0.0310	0.9899	<b>0.5354</b>	<b>0.8098</b>	0.3953	0.8616	0.2032	0.9389
Smish	0.0252	0.9916	0.0300	0.9900	0.5502	0.8052	0.5538	0.7982	0.2100	0.9358
Logish	0.0235	0.9923	0.0290	0.9906	0.5455	0.8055	0.5025	0.8162	0.2079	0.9372
sinLU	0.0273	0.9911	0.0381	0.9879	0.5905	0.7919	0.5986	0.7816	0.2119	0.9357
$\alpha$ SechSig	0.0224	0.9927	0.0290	0.9907	0.5428	0.8084	<b>0.3402</b>	<b>0.8770</b>	0.2029	0.9378
$\alpha$ TanhSig	<b>0.0220</b>	<b>0.9932</b>	<b>0.0270</b>	<b>0.9908</b>	0.5389	0.8090	0.4273	0.8472	<b>0.2029</b>	<b>0.9393</b>

**Table 9** Validated loss and accuracy of the activation functions

Methods	MNIST		k-MNIST		CIFAR-10		STL-10		SHVN-C	
	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc
Sigmoid	0.0213	0.9928	0.1937	0.9518	1.2441	0.5945	1.9183	0.4317	0.3040	0.9134
ReLU	0.0145	0.9955	0.1295	0.9722	0.5828	0.7977	1.4371	0.5334	0.2063	0.9468
leakyReLU	0.0235	0.9937	0.1861	0.9518	0.7936	0.7313	1.5582	0.5145	0.4090	0.9012
Swish	0.0167	0.9952	0.1398	0.9703	<b>0.5667</b>	<b>0.8103</b>	<b>1.5397</b>	<b>0.5677</b>	0.2146	0.9436
Mish	0.0160	0.9948	0.1297	0.9718	0.5699	0.8065	2.0084	0.5347	0.2155	0.9446
Smish	0.0173	0.9946	0.1395	0.9698	0.5721	0.8021	1.6410	0.5319	0.2272	0.9400
Logish	0.0191	0.9944	0.1291	0.9717	0.5781	0.8040	1.7381	0.5330	0.2293	0.9393
sinLU	0.0165	0.9943	0.1364	0.9687	0.6184	0.7909	1.9818	0.4895	0.2149	0.9437
$\alpha$ SechSig	<b>0.0139</b>	<b>0.9959</b>	0.1364	0.9723	0.5631	0.8085	1.6127	0.5676	0.2183	0.9443
$\alpha$ TanhSig	0.0145	0.9956	<b>0.1200</b>	<b>0.9731</b>	0.5668	0.8077	1.6331	0.5598	<b>0.2002</b>	<b>0.9476</b>

The  $\alpha$  values of  $\alpha$ -SechSig and  $\alpha$ -TanhSig in the MNIST dataset are  $\alpha = 0.2$  and  $\alpha = 0.3$ , respectively.

The  $\alpha$  values of  $\alpha$ -SechSig and  $\alpha$ -TanhSig in the kMNIST dataset are  $\alpha = 0.6$  and  $\alpha = 0.1$ , respectively.

The  $\alpha$  values of  $\alpha$ -SechSig and  $\alpha$ -TanhSig in the CIFAR-10 dataset are  $\alpha = 0.2$  and  $\alpha = 0.4$ , respectively.

In the STL-10 dataset, the  $\alpha$  values of both  $\alpha$ -SechSig and  $\alpha$ -TanhSig are  $\alpha = 0.2$ ,

The  $\alpha$  values of  $\alpha$ -SechSig and  $\alpha$ -TanhSig in the SHVN-Cropped dataset are  $\alpha = 0.2$  and  $\alpha = 0.1$ , respectively.

When Table 9 is examined, validation accuracy and error values are given for all activation functions.

When Table 9 is examined, it is observed that the best validation accuracy and validation error values in MNIST, k-MNIST, SHVN-C datasets are obtained in the suggested  $\alpha$ -SechSig and  $\alpha$ -TanhSig activation functions. In addition, the highest score in the other CIFAR-10 and STL-10 datasets belongs to the Swish activation function. Another striking point in these results is that the scores of the Swish and suggested  $\alpha$ -SechSig AF in the STL-10 dataset are very close to each other. In addition, thanks to parameters added, the flexibility feature of the AF has been gained.

Thus, it will continue the learning process by avoiding the errors that may occur from the trainable parameters obtained for each neuron.  $\alpha$ -SechSig and  $\alpha$ -TanhSig non-monotonic AF inherit merits of smooth AF such as Sigmoid and Tanh and piecewise AF such as ReLU and its variants and avoids their deficiencies. When the AF in the literature are examined in general, it is observed that not all AF on all datasets give high success and consistent results. This study hypothesizes that as the positive input increases, the positive component approaches identity mapping rather than using it, potentially bringing non-linearity properties to the positive part and making it more resistant to data distribution.

During the validation of the deep learning model, the  $\alpha$  values of the suggested AF according to the highest accuracy scores are as follows:

- The  $\alpha$  values of  $\alpha$ -SechSig and  $\alpha$ -TanhSig in the MNIST dataset are  $\alpha = 0.2$  and  $\alpha = 0.5$ , respectively.
- The  $\alpha$  values of  $\alpha$ -SechSig and  $\alpha$ -TanhSig in the kMNIST dataset are  $\alpha = 0.6$  and  $\alpha = 0.5$ , respectively.
- The  $\alpha$  values of  $\alpha$ -SechSig and  $\alpha$ -TanhSig in the CIFAR-10 dataset are  $\alpha = 0.1$  and  $\alpha = 0.3$ , respectively.

M  
N  
I  
S  
T

K  
M  
N  
I  
S  
T

C  
I  
F  
A  
R  
-  
1  
0

S  
T  
L  
-  
1  
0

S  
V  
H  
N  
-  
C  
r  
o  
p  
p  
e  
d

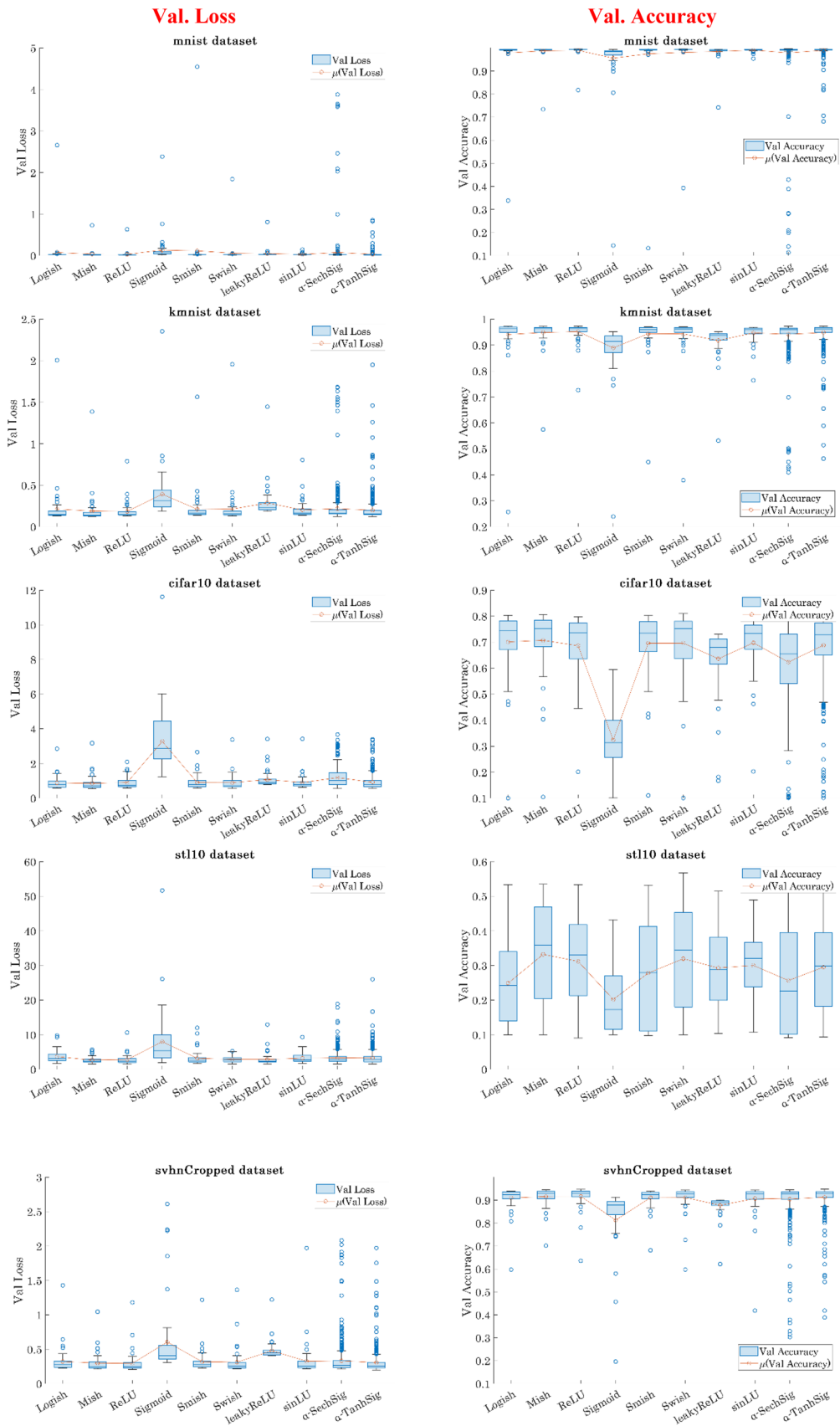


Fig. 9 Box charts of the AF according to validation loss and accuracy



**Table 10** Validated loss and accuracy of the activation functions for ResNet50

Methods	MNIST		k-MNIST		CIFAR-10		STL-10		SHVN-C	
	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc
Sigmoid	0.0349	0.9952	0.1931	0.9558	2.2598	0.5555	1.7104	0.4417	0.3141	0.9134
ReLU	0.0344	0.9956	0.1191	0.9555	0.6709	0.8090	1.4471	0.5444	0.2143	0.9447
leakyReLU	0.0367	0.9947	0.1811	0.9558	0.7305	0.8006	1.5502	0.5145	0.4191	0.9112
Swish	0.0511	0.9927	0.1398	0.9503	0.6602	0.8076	<b>1.5477</b>	<b>0.5677</b>	0.2354	0.9434
Mish	0.0370	0.9953	0.1191	0.9558	0.7075	0.8099	2.0004	0.5447	0.2044	0.9444
Smish	0.0377	0.9949	0.1391	0.9598	0.7035	0.8069	1.6410	0.5417	0.2272	0.9411
Logish	0.0318	0.9959	0.1191	0.9555	0.6859	0.8002	1.7401	0.5440	0.2293	0.9393
sinLU	0.0452	0.9923	0.1314	0.9585	0.6605	0.8005	1.7010	0.4075	0.2009	0.9437
$\alpha$ SechSig	0.0263	0.9977	0.1314	0.9653	0.6630	0.8105	1.6127	0.5676	0.2175	0.9443
$\alpha$ TanhSig	0.0262	0.9976	<b>0.1100</b>	<b>0.9635</b>	0.6061	0.8154	1.6441	0.5570	<b>0.2112</b>	<b>0.9474</b>

- The  $\alpha$  values of both  $\alpha$ -SechSig and  $\alpha$ -TanhSig in the STL-10 dataset are  $\alpha = 0.2$  and  $\alpha = 0.7$ , respectively.
- The  $\alpha$  values of  $\alpha$ -SechSig and  $\alpha$ -TanhSig in the SHVN-Cropped dataset are  $\alpha = 0.1$  and  $\alpha = 0.6$ , respectively.

In Fig. 9, box-plot graphs of AF according to validation accuracy and error values for MNIST, KMNIST, CIFAR-10, STL-10, and SVHN-Cropped are shown. It is seen that the validation error values of the proposed non-linear non-monotonic  $\alpha$ -SechSig and  $\alpha$ -TanhSig AF are better with small differences on all datasets compared to other activation functions. It also appears that the graph of the sigmoid function is always large over all datasets. When all validation error graphs are examined, it is seen that non-monotonic AF work stably with the proposed AF and produce results. When the validation accuracy graphs are examined for all datasets, it is observed that the other activation functions, except sigmoid, work with similar characteristics. The validation accuracy graphs of MNIST, KMNIST, and SVHN-Cropped values seem to produce similar results. When the box graphs are examined, it is seen that it gives high-performance results when the median line is skewed to the left inside the box (Kiliçarslan and Celik 2021, 2022). Also, the whisker lengths in the boxplots are nearly the same size, except for the sigmoid. The tiny whisker lengths in the box plots and the little difference between the lowest and best performance values observed in the present tests are indicators of the stability of the activation functions. Experiments with ResNet50 transfer learning method with the same parameters are given in Table 10.

When Table 10 is analyzed, it is seen that the best verification accuracy and verification error values in MNIST, k-MNIST, SHVN-C datasets are obtained with the proposed  $\alpha$ SechSig and  $\alpha$ TanhSig activation functions in the experiments performed with the ResNet50 model. other. The proposed activation function parameters in the

ResNet50 model were realized using the values in Table 9. Thus, the consistency in the results is observed.

In this study, non-linear non-monotonic  $\alpha$ SechSig and  $\alpha$ TanhSig AF are proposed. The proposed activation function is designed to improve the typical monotonic and non-monotonic AFs in deep learning architectures in terms of consistency and stability. When the AFs in the literature are analyzed in general, it is seen that not all AFs give high success and consistent results on all datasets (Kiliçarslan and Celik 2021). In addition, the proposed AF successfully overcomes the problems of ignoring negative weights with gradient measurements in the literature. The experimental results of the proposed AF on five datasets (MNIST, KMNIST, CIFAR-10, STL-10 and SVHN-Cropped) using both 3-layer CNN and ResNet50 model show that good results are obtained compared to other AFs. In addition, similar results were obtained with the swish activation function on a few datasets.

## 6 Conclusions

In deep learning architectures, AF plays an important role in processing the data entering the network to provide the most relevant output. In deep learning architectures, considerations such as avoiding model local minima and improving training performance are taken into account when constructing AF. Since experiments are usually performed on complex data sets, non-linear AF is mostly preferred in the literature. In addition, new AFs have been proposed in the literature to overcome the problems of missing gradients and ignoring negative weights. The  $\alpha$ SechSig and  $\alpha$ TanhSig AF proposed in our study can successfully overcome the existing problems. In deep learning architectures, non-monotonic activation functions are used instead of monotonic non-linear activation functions to ensure efficient operation of deep neural networks.

When the experimental evaluation results are examined, it is seen that the proposed non-monotonic AF is more successful than the monotonic ReLU activation function, which is widely used in the literature both with 3-layer CNN and ResNet50 model. In addition, the non-linearity and discriminability of the developed AF are among its important features. Because during the training and back-propagation of the model, it is necessary to calculate how much the curve will change the input data in which direction. In the experimental evaluations,  $\alpha$ SechSig and  $\alpha$ TanhSig AF were tested on MNIST, KMNIST, SVHN-Cropped, STL-10, and CIFAR-10 datasets. According to the results, non-monotonic Swish, Logish, Mish, Smish, and monotonic ReLU have higher classification scores than SinLU and LReLU. In future studies, the proposed AF can be tested for classification and segmentation on more specific image and video data to test the prevalence of their success.

**Author contributions** All authors have contributed equally.

**Funding** No funding was received to assist with the preparation of this manuscript.

**Data availability** All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The datasets generated during and/or analyzed during the current study are available in the [tensorflow] repository, [<https://www.tensorflow.org/datasets/catalog/overview>].

## Declarations

**Conflict of interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Ethical approval** The authors declare that they have no conflict of interest.

## References

- Adem K, Közkurt C (2019) Defect detection of seals in multilayer aseptic packages using deep learning. *Turk J Electr Eng Comput Sci* 27(6):4220–4230. <https://doi.org/10.3906/elk-1903-112>
- Adem K, Kiliçarslan S, Cömert O (2019) Classification and diagnosis of cervical cancer with stacked autoencoder and softmax classification. *Expert Syst Appl* 115:557–564
- Adem K, Kiliçarslan S COVID-19 Diagnosis Prediction in Emergency Care Patients using Convolutional Neural Network. *Afyon Kocatepe Üniversitesi Fen Ve Mühendis. Bilim. Derg.*, 21(2), Art. no. 2, Apr. 2021, <https://doi.org/10.35414/akufemubid.788898>.
- Apicella A, Donnarumma F, Isgrò F, Prevete R (2021) A survey on modern trainable activation functions. *Neural Netw* 138:14–32. <https://doi.org/10.1016/j.neunet.2021.01.026>

- Baş S (2018) A new version of spherical magnetic curves in the desitter space  $S^1 \times S^1$ . *Symmetry* 10(11):606
- Bawa VS, Kumar V (2019) Linearized sigmoidal activation: a novel activation function with tractable non-linear characteristics to boost representation capability. *Expert Syst Appl*. <https://doi.org/10.1016/j.eswa.2018.11.042>
- Clanuwat T, Bober-Irizar M, Kitamoto A, Lamb A, Yamamoto K, Ha D Deep learning for classical Japanese literature. *ArXiv181201718 Cs Stat*, 9999, <https://doi.org/10.20676/00000341>.
- Clevert D-A, Unterthiner T, Hochreiter S Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *ArXiv151107289 Cs*, Feb. 2016, Accessed: Apr. 27, 2022. [Online]. <http://arxiv.org/abs/1511.07289>
- Coates A, Ng A, Lee H An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics, 2011, pp. 215–223
- Elen A (2022) Covid-19 detection from radiographs by feature-reinforced ensemble learning. *Concurrency Computat Pract Exper* 34(23):e7179. <https://doi.org/10.1002/cpe.7179>
- Gironés RG, Gironés RG, Palero RC, Boluda JC, Boluda JC, Cortés AS (2005) FPGA implementation of a pipelined on-line backpropagation. *J VLSI Signal Process Syst Signal, Image Video Technol* 40:189–213
- Gorur K, Kaya Ozer C, Ozer I, Can Karaca A, Cetin O, and Kocak I, ‘Species-Level Microfossil Prediction for Globotruncana genus Using Machine Learning Models’, *Arab. J. Sci. Eng.*, pp. 1–18, 2022.
- He K, Zhang X, Ren S, Sun J Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, 2015, pp 1026–1034. Accessed: Apr. 27, 2022. [Online]. [https://openaccess.thecvf.com/content\\_iccv\\_2015/html/He\\_Delving\\_Deep\\_into\\_ICCV\\_2015\\_paper.html](https://openaccess.thecvf.com/content_iccv_2015/html/He_Delving_Deep_into_ICCV_2015_paper.html)
- Hendrycks D, Gimpel K Gaussian Error Linear Units (GELUs). *ArXiv160608415 Cs*, Jul. 2020, Accessed: Apr. 27, 2022. [Online]. <http://arxiv.org/abs/1606.08415>
- Kiliçarslan S, Celik M (2021) RSigELU: a nonlinear activation function for deep neural networks. *Expert Syst Appl* 174:114805. <https://doi.org/10.1016/j.eswa.2021.114805>
- Kiliçarslan S, Celik M (2022) KAF + RSigELU: a nonlinear and kernel-based activation function for deep neural networks. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-022-07211-7>
- Kiliçarslan S, Közkurt C, Baş S, Elen A (2023) Detection and classification of pneumonia using novel Superior Exponential (SupEx) activation function in convolutional neural networks. *Expert Syst Appl* 217:119503
- Kiliçarslan S, Adem K, Çelik M (2021) An overview of the activation functions used in deep learning algorithms. *J. New Results Sci* 10(3), Art. no. 3, <https://doi.org/10.54187/jnrs.1011739>.
- Klambauer G, Unterthiner T, Mayr A, Hochreiter S Self-Normalizing Neural Networks. *ArXiv170602515 Cs Stat*, Sep. 2017, Accessed: Apr. 27, 2022. [Online]. <http://arxiv.org/abs/1706.02515>
- Korpinar T, Baş S (2019) A new approach for inextensible flows of binormal spherical indicatrices of magnetic curves. *Int J Geom Methods Mod Phys* 16(02):1950020
- Krizhevsky A, Sutskever I, Hinton GE ImageNet classification with deep convolutional neural networks. In: Advances in neural information processing systems, 2012, vol. 25. Accessed: Apr. 28, 2022. [Online]. <https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html>
- Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324. <https://doi.org/10.1109/5.726791>

- LeCun Y, Bengio Y, Hinton G (2015) Deep learning, Nature 521(7553), Art. no. 7553, <https://doi.org/10.1038/nature14539>.
- Maas AL, Hannun AY (2013) Ng AY Rectifier nonlinearities improve neural network acoustic models
- Misra D, Mish: A Self Regularized Non-Monotonic Activation Function, *ArXiv190808681 Cs Stat*, Aug. 2020, Accessed: Apr. 27, 2022. [Online]. <http://arxiv.org/abs/1908.08681>
- Nair V, Hinton GE Rectified Linear Units Improve Restricted Boltzmann Machines. In: Presented at the ICML, Jan. 2010. Accessed: Apr. 27, 2022. [Online]. Available: <https://openreview.net/forum?id=rkb15iZdZB>
- Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng AY (2011) Reading digits in natural images with unsupervised feature learning
- Pacal I, Karaboga D (2021) A robust real-time deep learning based automatic polyp detection system. *Comput Biol Med* 134: 104519–104519
- Paul A, Bandyopadhyay R, Yoon JH, Geem ZW, Sarkar R SinLU: Sinu-Sigmoidal Linear Unit. *Mathematics*, 10(3), Art. no. 3, Jan. 2022, <https://doi.org/10.3390/math10030337>.
- Ramachandran P, Zoph B, Le QV Searching for Activation Functions, *ArXiv171005941 Cs*, Oct. 2017, Accessed: Apr. 27, 2022. [Online]. <http://arxiv.org/abs/1710.05941>
- Scardapane S, Van Vaerenbergh S, Totaro S, Uncini A (2019) Kafnets: kernel-based non-parametric activation functions for neural networks. *Neural Netw* 110:19–32. <https://doi.org/10.1016/j.neunet.2018.11.002>
- Trottier L, Giguère P, Chaib-draa B Parametric Exponential linear unit for deep convolutional neural networks. *ArXiv160509332 Cs*, Jan. 2018, Accessed: Apr. 27, 2022. [Online]. <http://arxiv.org/abs/1605.09332>
- Wang X, Ren H, Wang A Smish: A novel activation function for deep learning methods, *Electronics* 11(4), Art. no. 4, Jan. 2022, <https://doi.org/10.3390/electronics11040540>.
- Ying Y, Su J, Shan P, Miao L, Wang X, Peng S (2019) Rectified exponential units for convolutional neural networks. *IEEE Access* 7:101633–101640. <https://doi.org/10.1109/ACCESS.2019.2928442>
- Zhou Y, Li D, Huo S, Kung S-Y (2021) Shape autotuning activation function. *Expert Syst Appl* 171:114534. <https://doi.org/10.1016/j.eswa.2020.114534>
- Zhu H, Zeng H, Liu J, Zhang X (2021) Logish: a new nonlinear nonmonotonic activation function for convolutional neural network. *Neurocomputing* 458:490–499. <https://doi.org/10.1016/j.neucom.2021.06.067>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.