KICS The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com
ScienceDirect

ICT Express 9 (2023) 320–325

www.elsevier.com/locate/icte

Self-gated rectified linear unit for performance improvement of deep neural networks☆

Israt Jahan[a,b], Md. Faisal Ahmed[a], Md. Osman Ali[a,c], Yeong Min Jang[a,*]

[a] Department of Electronics Engineering, Kookmin University, Seoul 02707, Republic of Korea
[b] Department of Electrical and Electronic Engineering, Daffodil International University, Dhaka 1341, Bangladesh
[c] Department of Electrical and Electronic Engineering, Noakhali Science and Technology University, Noakhali 3814, Bangladesh

Received 8 October 2021; received in revised form 24 December 2021; accepted 24 December 2021
Available online 3 January 2022

Abstract

This technical paper proposes an activation function, self-gated rectified linear unit (SGReLU), to achieve high classification accuracy, low loss, and low computational time. Vanishing gradient problem, dying ReLU, noise vulnerability are also resolved in our proposed SGReLU function. SGReLU's performance is evaluated on MNIST, Fashion-MNIST, and Imagenet datasets and compared with seven highly effective activation functions. We obtained that the proposed SGReLU outperformed other activation functions in most cases in VGG16, Inception v3, and ResNet50. In VGG16 and Inception v3, it achieved an accuracy of 90.87% and 95.01%, respectively, exceeding other functions with the second-fastest computing time in these networks.

Keywords: Image classification; Activation function; Deep neural network; Accuracy; Time complexity

1. Introduction

An activation function is considered to be the heart of a deep neural network (DNN) because it is the basic building block of a multilayer perceptron (MLP) and convolutional neural network (CNN), which helps the network learn complex patterns in the data. The performance of NNs, such as accuracy, computational complexity, and loss, highly depends on the characteristics of the activation function used. To improve the overall performance of NNs, researchers have proposed several activation functions over the last decade. Some remarkable activation functions are sigmoid, tanh, rectified linear unit (ReLU) [1,2], leaky ReLU (LReLU) [3], exponential linear unit (ELU) [4], scaled-exponential linear unit (SELU) [5], and swish [6].

ReLU is the most widely used activation function for NNs, which was first developed for limited Boltzmann machines [1,2]. Despite having low computational complexity, it suffers from the dying ReLU problem that refers to the idle state of neurons as it only provides zero as output irrespective of the input [3,7]. Several activation functions, such as LReLU, parametric ReLU (PReLU), and randomized ReLU (RReLU), have overcome the dying ReLU problem. However, they lack noise-robustness in the deactivation state [4]. Sigmoid linear unit (SiLU) [8] uses sigmoid gating technique to remove dying ReLU problem, which increases the time complexity due to the power operation of sigmoid function. Swish [6], a modification of SiLU needs additional parameter training to achieve better accuracy at the expense of higher computational complexity. Table 1 presents brief descriptions of various popular activation functions with their expressions and challenges.

To overcome the limitation of ReLU and swish, we have proposed a self-gated ReLU (SGReLU) that also overcomes the major limitations of other activation functions, such as vanishing gradient, neuron death, and output offset. The performance of the proposed SGReLU is evaluated in MLP and some benchmark CNNs, such as VGG16, Inception v3, and ResNet50. Consequently, we obtained exceptional performance improvement.

☆ This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2018-0-01396) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

* Corresponding author.
E-mail addresses: israt.eee2k11@gmail.com (I. Jahan), faisal.ahmed@ieee.org (M.F. Ahmed), osman.eee@nstu.edu.bd (M.O. Ali), yjang@kookmin.ac.kr (Y.M. Jang).

Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2021.12.012

Table 1
Notable activation functions with their limitations.

Activation function	Expression	Challenges
Sigmoid [3]	$\frac{1}{1+e^{-x}}$	• Gradient disappearance problem • Not zero-centered
Tanh [3]	$\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	• Vanishing gradient problem
ReLU [1–3]	x, for $x \geq 0$ 0, for $x < 0$	• Dying ReLU problem
LReLU [4]	x, for $x \geq 0$ $0.01x$, for $x < 0$	• Lack noise robustness in deactivation state
Swish [3,6,9]	$\frac{x}{1+e^{-\beta x}}$	• In case of β-swish, the value of β should be trained. • Higher computational complexity
ELU [4,10]	x, for $x \geq 0$ $a(e^{x}-1)$, for $x < 0$	• Computationally expensive
SELU [8,11]	$\lambda \begin{cases} x, \text{ for } x > 0 \\ a(e^{x}-1), \text{ for } x \leq 0 \end{cases}$	• Requires LeCun normal method for weight initialization. • Needs alpha dropout.
Softplus [3,6]	$\log\,(1+e^{x})$	• Computationally expensive • Strictly positive and monotonic
Softsign [12]	$\frac{x}{1+\lvert x\rvert}$	• Complex derivatives • Gradient sometimes yields extremely low/high values.

The most remarkable upsides of our proposed SGReLU are as follows:

1. The dying ReLU and vanishing gradient problem are settled by adding a small negative bump on the negative side.
2. After the bump, the negative value tends to zero for larger negative inputs; thus, making the deactivation state noise-robust.
3. Our proposed function has achieved higher accuracy, which surpasses ReLU and swish by 0.42% and 0.97% in VGG16, 1.11% and 1.51% in Inception v3, and 0.625% and 0.7% in ResNet50, respectively, using Imagenet dataset.
4. It is computationally efficient than swish due to its linear characteristics on the positive side.

The remainder of the paper is organized as follows. The proposed function is described in Section 2. Section 3 provides the results and discussions. Finally, Section 4 presents the conclusion.

2. Proposed function

A general structure of activation functions can be written as a binary function, $b\,(x, g\,(x))$ [5], where x is the raw preactivation, which is the input to the final binary function. SGReLU is proposed maintaining the same binary format having $g\,(x) = \alpha.x.\sigma\,(x)$, which is expressed as follows:

$$f\,(x) = \begin{cases} x, & x \geq 0 \\ \alpha.x.\sigma\,(x), & x < 0 \end{cases} \tag{1}$$

where $\sigma\,(x) = \frac{1}{1+e^{-x}}$, and α is a hyper-parameter ranging from 0.1 to 1.

In the positive region, the function shows similar unbounded linear behavior as ReLU. Therefore, it avoids the saturation problem and has a non-zero gradient, which accelerates the training process. In contrast to swish, the absence of the sigmoid function eliminates the necessity of power operations, resulting in a lower computational time. On the other hand, self-gating technique is applied to the negative region of the function to make it compatible with a single input. As a result, a small negative bump is developed at the beginning of the negative region that approaches zero for a large value of preactivation, x. For small negative inputs, SGReLU offers a negative bump that prevents the dying ReLU problem. It has some portions on the negative side, but unlike LReLU and PReLU, it is bounded below because it approaches zero for a large value of x that introduces sparsity in the network which reduces the complexity. Consequently, the overfitting problem is removed, and noise-robustness is achieved in the network simultaneously. Thus, non-monotonicity and self-regularization are attained that makes the network free from neural death and compatible with any data, respectively.

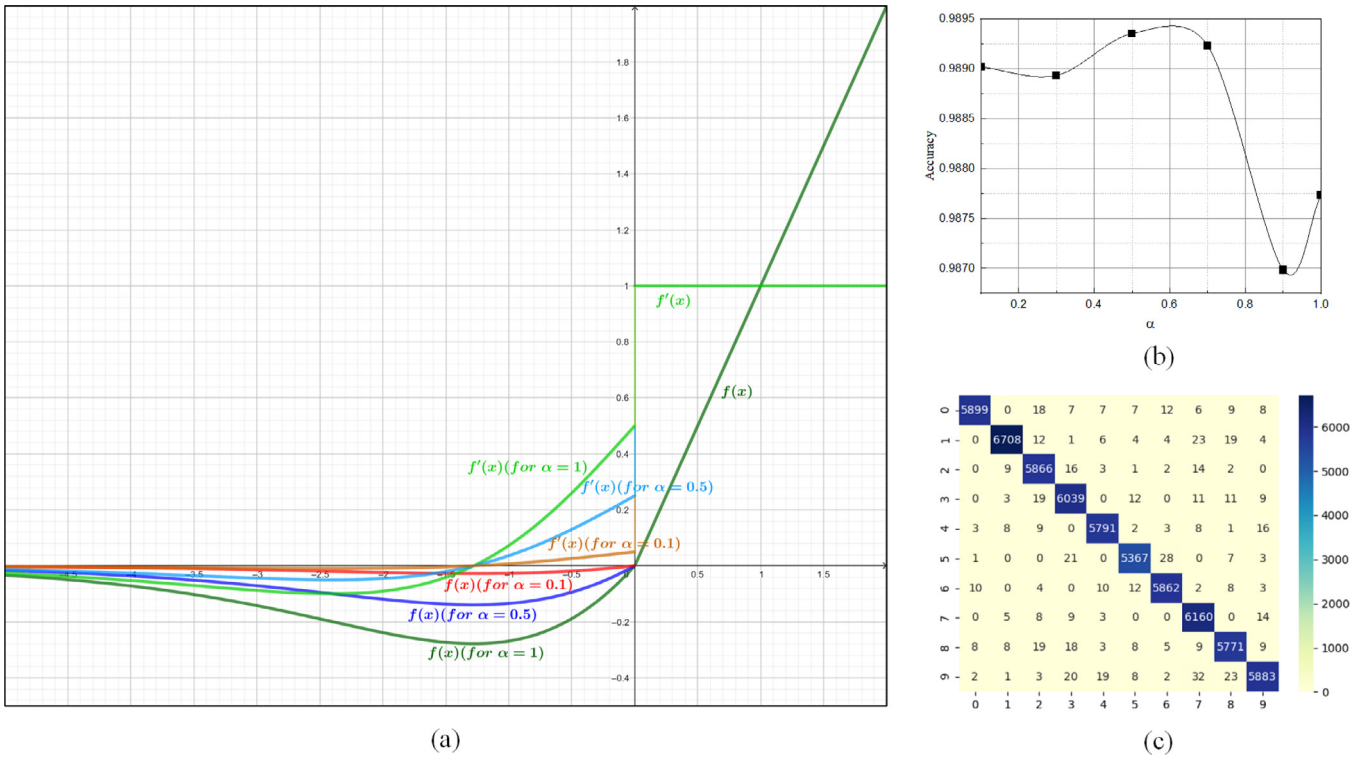

Fig. 1. (a) The proposed SGReLU activation function and its first derivative, (b) the variation in training accuracy (after 10^{th} epoch) with the variation of hyper-parameter (α), and (c) the heatmap of confusion matrix in case of MNIST dataset when $\alpha = 0.5$ (after 10^{th} epoch).

The first derivative of the continuous SGReLU function is as follows:

$$f'(x) = \begin{cases} 1, & x \geq 0 \\ \alpha\left[\frac{1}{1+e^{-x}} + \frac{xe^{-x}}{(1+e^{-x})^2}\right], & x < 0 \end{cases} \quad (2)$$

The derivative is always one for non-zero inputs in the positive region; thus, alleviating the attenuation and vanishing gradient problem. However, the negative region is a combination of two functions; therefore, a partial derivative is performed to obtain the gradient at the time of backpropagation. Fig. 1(a) shows the effect of α on SGReLU and its derivative.

Here, α controls the depth of the negative bump of the SGReLU function. It also controls the shape and values of the non-saturated part of the negative regime of the derivative during backpropagation. If α gets close to zero, the SGReLU function will behave like ReLU. To determine the optimum value of α, different values of α were investigated in MLP, VGG16, Inception v3, and ResNet50 on the Modified National Institute of Standards and Technology (MNIST) [13], Fashion-MNIST [14], and Imagenet dataset using hyper-parameter tuning. In general, hyper-parameter tuning is a very efficient way to find the optimum value of hyper-parameter. It is noticed that SGReLU function for $\alpha = 0.5$ performs better than other values of α. In the next section, the performance analysis of SGReLU for different datasets and NN structures has been described. Fig. 1(b) shows the training accuracy in the MLP model for the MNIST dataset after the 10^{th} epoch with the variation of α. The maximum accuracy is achieved at $\alpha = 0.5$, and the corresponding heat map of the confusion matrix is shown in Fig. 1(c).

3. Results and discussions

3.1. Performance analysis of SGReLU using MLP

We designed a NN that had one hidden layer with 30 neurons. The batch size and learning rate were 16 and 0.01, respectively, and the random binary bias was used. SGReLU was used in the hidden layer, and the sigmoid function was used in the output layer as an activation function. We executed training based on weight updates through backpropagation. MNIST and Fashion-MNIST datasets were used to train the DNN models. MNIST consists of ten classes with 70,000 handwritten digits, whereas the Fashion-MNIST has the same number of apparel photos. Photos of both datasets are 28×28 grayscale images. In both cases, 60,000 and 10,000 images were used for training and testing.

Fig. 2 shows the training accuracy, test accuracy, and mean square error loss for 15 epochs in the above mentioned datasets for ReLU, swish, and SGReLU (for α = 0.5 and 1). As shown in Fig. 2(a), SGReLU achieved the maximum average training accuracy of 98.3% with $\alpha = 0.5$ for the MNIST dataset. Additionally, SGReLU with $\alpha = 1$ achieved an average training accuracy of 98.22%, which is higher than that of the swish (98.1%) and ReLU (97.5%). SGReLU with $\alpha = 0.5$ and $\alpha = 1$ also achieved an average testing accuracy of 97.23% and 97.1%, respectively, which are higher than that of swish (97.04%) and ReLU's (96.32%) case, as shown in Fig. 2(b). As shown in Fig. 2(c), SGReLU increased the accuracy and reduced the loss by 4% and 3% from ReLU and swish, (when $\alpha = 1$) respectively, and 5.7% and 4.6% from ReLU and swish (when $\alpha = 0.5$) respectively.

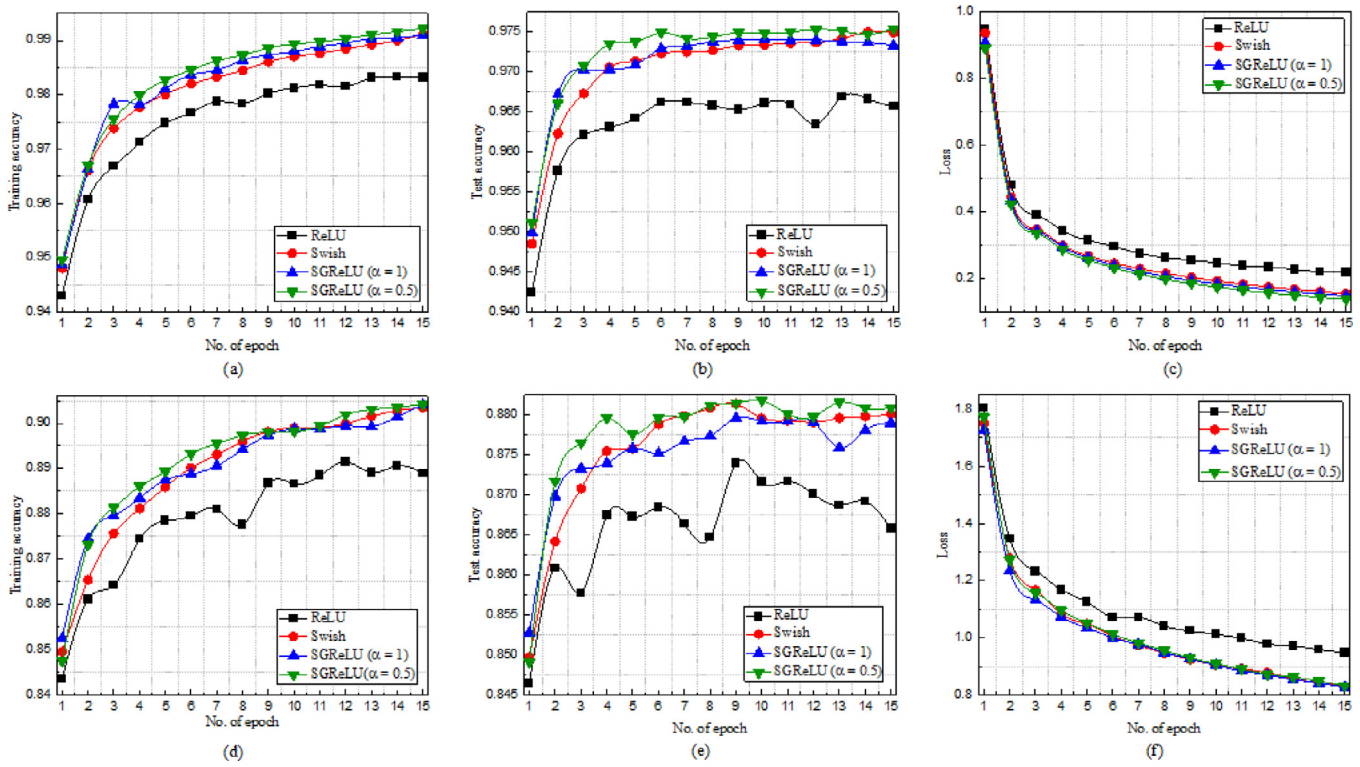

Fig. 2. (a) Training accuracy, (b) test accuracy, and (c) loss curve of the MNIST dataset; (d) training accuracy, (e) test accuracy, and (f) loss curve of the Fashion-MNIST dataset.

For Fashion MNIST, SGReLU with $\alpha = 0.5$ and $\alpha = 1$ displayed the top and next to the top average training accuracy of 89.15% and 89%, respectively. Similar to the previous case, swish and ReLU obtained the third and fourth positions by achieving average training accuracies of 88.94% and 87.8%, respectively. These results are shown in Fig. 2(d). Even in average testing accuracy (Fig. 2(e)), SGReLU with $\alpha = 0.5$ showed the leading accuracy of 87.74%, which is 0.24%, 0.18%, and 1.14% higher than the SGReLU with $\alpha = 1$, swish, and ReLU, respectively. In the case of loss, Fig. 2(f) shows that SGReLU with $\alpha = 1$ has the lowest average loss; however, the differences among the loss curves of swish, SGReLU ($\alpha = 1$), and SGReLU ($\alpha = 0.5$) are negligible.

3.2. Performance of SGReLU in different CNN structures

As mentioned previously, three special types of CNN structures were used for the performance analysis of the proposed SGReLU. First, we assessed the proposed SGReLU's performance in VGG16. It differs from the conventional CNN in a way that small stacked convolutional filters with fewer parameters constitute its blocks, showing similar performance with large convolutional filters. The second one was Inception v3, also known as GoogLeNet. It assists object detection and image processing. Finally, ResNet50 that has skip connections, i.e., an activation after a convolution, was being tested.

The performance of SGReLU was considered with two different values of α (0.5 and 1) in terms of accuracy and computational time. An image classification dataset, ImageNet 2012, which is considered the most significant image classification dataset, with two classes, 2,000 training images and 1,000 validation images of 28 × 28 pixels was used for performance evaluation. The results for SGReLU along with ReLU, softplus, ELU, SELU, softsign, and swish were taken five times and their average results were recorded in Table 2. Each operation was performed using batch size 20, 10 epochs, and 100 steps per epoch. In VGG16, SGReLU showed the finest result (90.87%) in terms of accuracy, and the computational time was second finest (905.44 s), which was near to the best shown by ReLU (899.95 s). An identical result was obtained in Inception v3, where SGReLU attained 1.11% better accuracy than ReLU but took 10.19 s more than ReLU for task accomplishment. In ResNet50, the accuracy (87.30%) and computational time (3046.96 s) were the second-best; however, they only differed from the best ones by 0.03% and 35.47 s, respectively.

3.3. Performance of SGReLU with learnable parameter

We designed an MLP that had an input layer, two hidden layers with 256 and 64 neurons, respectively, and an output layer with 10 nodes. The batch size and learning rate were 32 and 0.001, respectively. The SGReLU was used in the two hidden layers, and Adam optimizer was used for training the network. SGReLU-α is a form of SGReLU where α is trained as a learning parameter that changes continuously in every iteration to reduce the cost function. For 3,500 iterations, α value was updated frequently to reach an optimum solution. The initial value of α was chosen 0.1 and after 3,500 training

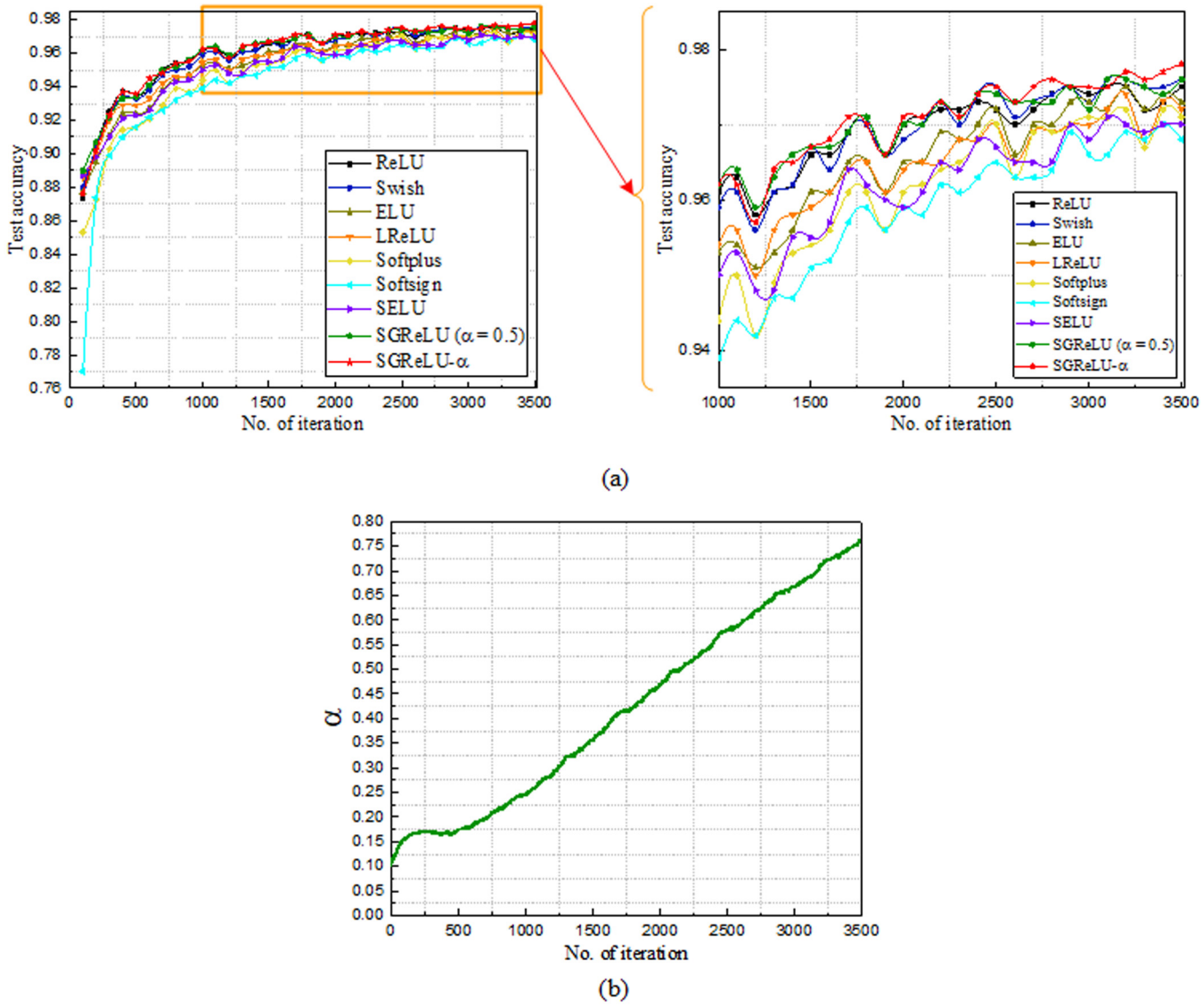

Fig. 3. For the MNIST dataset, (a) test accuracy vs. iteration of various activation functions, (b) variation of α with the number of iterations in case of SGReLU-α.

Table 2
Performance of SGReLU in VGG16, Inception v3, and ResNet50.

Activation function	VGG16		Inception v3		ResNet50	
	Top-10 accuracy (%)	Computational time (s)	Top-10 accuracy (%)	Computational time (s)	Top-10 accuracy (%)	Computational time (s)
ReLU	90.45	**899.95**	93.90	**290.63**	86.675	**3011.49**
SGReLU	**90.87**	**905.44**	**95.01**	**300.82**	**87.30**	**3046.96**
Softplus	89.00	908.29	93.84	376.14	86.855	3292.02
ELU	86.94	926.44	94.29	377.29	**87.33**	3236.65
SELU	87.73	917.71	94.90	379.17	86.00	3226.87
Softsign	87.85	917.62	94.9	375.47	78.20	3282.30
Swish	89.90	909.93	93.5	316.51	86.60	3322.39
LReLU	88.73	914.00	94.28	330	86.60	3251.90

Table 3
Performance comparison of SGReLU-α with other activation functions.

Activation function	Top-5 average test accuracy (%)
ReLU	95.95
SGReLU (α=0.5)	96.04
SGReLU-α	**96.06**
Swish	95.91
ELU	95.41
LReLU	95.48
Softplus	94.85
Softsign	94.32
SELU	95.17

steps, the optimum value of α was 0.761, where the test accuracy was 97.8% after 5^{th} epoch. In Table 3, the performance analysis of different activation functions in the MNIST dataset has been shown in terms of average test accuracy. SGReLU-α had outperformed other activation functions i.e; ReLU by 0.11%, swish by 0.15%, ELU by 0.65%, LReLU by 0.58%, softplus by 1.21%, softsign by 1.74%, SELU by 0.89%, as shown in Table 3. SGReLU-α performed slightly better than the original form of SGReLU (with α=0.5). The performance of several activation functions in terms of test accuracy is shown in Fig. 3(a). In Fig. 3(b), the variation of α with the number of iterations in case of SGReLU-α has been shown.

4. Conclusion

The performance of activation functions varies with NN models and datasets. Therefore, this study identifies the challenges faced in developing a unique activation function suitable for all models and datasets. Our proposed SGReLU includes the most desirable features of an activation function, such as unbounded above, bounded below, non-monotonic, linear in the positive region, and smooth in the negative region. Moreover, it has shown the best classification accuracy among the most common functions without affecting the running time, which is the second-best. Therefore, it can be concluded that SGReLU with fixed α = 0.5, or hyper-parameter tuning or SGReLU-α with learnable parameters all are capable of enhancing the performance of the DNNs.

CRediT authorship contribution statement

Israt Jahan: Conceptualization of this study, Methodology, Implementation, Writing – original draft. **Md. Faisal Ahmed:** Data curation, Implementation, Writing – original draft. **Md. Osman Ali:** Methodology, Writing – original draft. **Yeong Min Jang:** Supervision.

Declaration of competing 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.

References

[1] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proc. International Conference on Machine Learning, Haifa, Israel, 2010, pp. 807–814.
[2] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: Proc. International Conference on Artificial Intelligence and Statistics Conference, Ft. Lauderdale, FL, USA, 2011.
[3] A. Apicella, F. Donnarumma, F. Isgrò, R. Prevete, A survey on modern trainable activation functions, Neural Netw. 138 (2021) 14–32.
[4] D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (ELUs), 2015, arXiv [cs.LG].
[5] G. Klambauer, T. Unterthiner, A. Mayr, S. Hochreiter, Self-normalizing neural networks, 2017, arXiv [cs.LG].
[6] P. Ramachandran, B. Zoph, Q.V. Le, Searching for activation functions, 2017, arXiv [cs.NE].
[7] L. Lu, Dying ReLU and initialization: Theory and numerical examples, Commun. Comput. Phys. 28 (5) (2020) 1671–1706.
[8] S. Elfwing, E. Uchibe, K. Doya, Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, Neural Netw. 107 (2018) 3–11.
[9] J. Sharma, Experiments with SWISH activation function on MNIST dataset, 2017, [Online]. Available: https://medium.com/@jaiyamsharma/experiments-with-swish-activation-function-on-mnist-dataset-fc89a8c79ff7.
[10] Z. Qiumei, T. Dan, W. Fenghua, Improved convolutional neural network based on fast exponentially linear unit activation function, IEEE Access 7 (2019) 151359–151367.
[11] C. Hansen, Activation functions explained - GELU, SELU, ELU, ReLU and more, 2019, [Online]. Available: https://mlfromscratch.com/activation-functions-explained/.
[12] T. Szandała, Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks, Bio-Inspired Neurocomputing, Springer, Singapore, 2021, pp. 203–224.
[13] L. Deng, The MNIST database of handwritten digit images for machine learning research [best of the web], IEEE Signal Process. Mag. 29 (6) (2012) 141–142.
[14] H. Xiao, K. Rasul, R. Vollgraf, Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms, 2017, arXiv [cs.LG].