UDC 004.4
DOI https://doi.org/10.32782/2663-5941/2024.3.1/22

Morozov A.V.
Zhytomyr Polytechnic State University

Levkivskyi V.L.
Zhytomyr Polytechnic State University

Plechystyy D.D.
Zhytomyr Polytechnic State University

ACTIVATION FUNCTIONS IN NEURAL NETWORKS: OVERVIEW AND COMPARISON

Neural networks have become one of the most powerful technologies of our time, revolutionizing many industries. Their capabilities allow solving complex tasks that were previously considered impossible. Neural networks consist of interconnected nodes called neurons that transmit and process information. Networks can have different structures, from simple ones with a few layers to complex ones with tens or even hundreds of layers. Each layer can contain thousands or millions of neurons, and connections between neurons can have different weights. Neural network training algorithms can be complex to tune and optimize, and the training process can be time-consuming, especially for large networks. Activation functions play a key role in neural networks by performing several important functions, namely, activation functions introduce nonlinearity into neural networks, which makes them capable of learning from complex data and performing complex tasks. In addition, activation functions determine the neuron's output value, which can be interpreted as a probability, magnitude, or other type of value, depending on the task. Neural networks are highly dependent on the activation functions used in their neurons and play a critical role in shaping the behavior and performance of neural networks. The article presents a comprehensive overview and comparative analysis of various activation functions commonly used in deep learning. Deep learning is a branch of machine learning that uses neural networks with a deep architecture. The paper examines various activation functions (Sigmoid, Tanh, ReLU, LeakyReLU, ELU, SELU, Swish, Mish, Softmax) used in neural networks. Their mathematical properties, advantages and limitations of each activation function are described in detail, as well as empirical data from experiments performed on reference datasets. The conducted studies provide insight into the selection and optimization of activation functions for neural network architecture.

Key words: *Activation functions, Neural networks, Deeplearning, Sigmoid, Tanh, ReLU, LeakyReLU, ELU, SELU, Swish, Mish, Softmax.*

Formulation of the problem. Neural networks are a fundamental building block of modern machine learning systems, and the choice of activation function profoundly impacts their expressiveness and optimization behavior. A deeper understanding of activation functions can improve the performance of neural networks, optimize them for specific tasks, and understand the principles of their operation. Studying different activation functions and their impact on neural networks is an active area of research. Activation functions introduce nonlinearity into the network, allowing it to learn complex relationships in the data. Just as humans need senses to understand the world, neural networks need activation functions to interpret information correctly. Activation functions are like filters that decide which signals to let through and which to ignore. They are crucial because they add an element of decision-making, allowing neural networks to understand complex relationships in data.

This paper will present a study and comparison of the characteristics of different activation functions. How activation functions affect how neural networks learn and perform. The world around us constantly changes, depending on how we perceive it. It is similar to trying on different pairs of glasses. Each pair provides a unique view of the world. Some glasses can improve your vision; others can change how you see colors or add special effects, making the world more attractive or unusual. Similarly, each activation function changes the way the neural network interprets data. Some make learning easier, while others can slow it down or cause confusion.

This research paper aims to provide a clear and comparative view of the different activation functions used in deep learning. It also aims to make complex mathematical concepts accessible to a broader audience.

The study will review different activation functions, focusing on their advantages and disadvantages. To better understand them, an experiment based on the results of actual experiments using reference datasets will be conducted. The experiment will provide an opportunity to see how different activation functions work in practice, gaining valuable insights into their effectiveness.

Analysis of recent research and publications. F. Kamalov, A. Nazir, M. Safaraliev, A. K. Cherukuri, and R. Zgheib delve into a lesser-explored dimension of neural network architecture: the correlation between activation functions and geometric model performance in feature space. While previous studies have extensively examined the impact of activation functions on model accuracy, little attention has been paid to their impact on the underlying geometry of neural network models. This study uses an empirical analysis to examine how different activation functions affect the geometric arrangement of trained neural network models. In particular, the analysis aims to elucidate the interaction between activations in the hidden and output layers, the geometry of the trained models, and their overall performance. To facilitate understanding, the paper proposes a visualization of trained neural network models, making it easier for researchers to intuitively understand the impact of activation functions on model geometry and performance [1].

M. Kaloev and G. Krustev explore the dynamic landscape of neural network research, which has witnessed rapid progress that has led to the emergence of various activation functions, each boasting unique characteristics. Consequently, the compatibility and interchangeability of these activation functions are becoming increasingly relevant in the field. This article addresses these issues by comprehensively studying deep neural networks' design, training, and evaluation (DNNs), focusing on their use in pattern recognition tasks. Through a thorough analysis, we investigate the performance and behavior of three standard activation functions – the sigmoid function (Sigmoid), the hyperbolic tangent (tanh), and the rectified linear unit (ReLU) – when used in the hidden layers of a neural network. By elucidating the nuances of these activation functions, the study aims to provide valuable information crucial for the optimal design and selection of activation functions in ANNs. Ultimately, this knowledge will contribute to developing neural network architecture and application areas [2].

Garrett Bingham, Risto Miikkulainen propose an innovative method to automate the customization of activation functions, which leads to a consistent improvement in network performance. Using evolutionary search, the proposed approach reveals the fundamental structure of the activation function, while gradient descent optimizes its parameters in different network segments and during training. Experimental validation on four different neural network architectures using the CIFAR-10 and CIFAR-100 image classification datasets has confirmed the effectiveness of this method. In particular, it identifies general-purpose activation functions and adapts specialized functions for different network architectures, consistently outperforming ReLU and other alternative activation functions by a significant margin. Thus, this automated optimization method is a promising way to improve the application of deep learning in various tasks [3].

Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri offer a detailed analysis and overview of activation functions (AFs) in neural networks for deep learning. It delves into various classes of FAs, including logistic sigmoid and tangent, ReLU-based, ELU-based, and learning-based functions. In addition, it discusses in detail various characteristics of ANNs, including input range, monotonicity, and smoothness. In addition, the paper evaluates performance by comparing 18 state-of-the-art ANNs with different network architectures and datasets. The findings from this analysis are intended to provide researchers with valuable guidance for further research and help practitioners select the most appropriate ANNs for their specific applications [4].

Andrea Apicella, Francesco Donnarumma, Francesco Isgro, and Roberto Prevete review the different models of learnable activation functions, starting with a discussion of the terminology used in the literature on "activation function". After that, he introduces a taxonomy of learnable activation functions, outlines the common and unique characteristics of current and past models, and thoroughly analyzes this methodology's main advantages and disadvantages. Notably, the article emphasizes that numerous proposed methodologies resemble integrating additional layers of neurons using fixed (non-trained) activation functions combined with simple local rules governing the respective weight layers [5].

Activation functions in artificial neural networks play a crucial role, allowing the network to comprehend and model complex nonlinear relationships between inputs and corresponding outputs. In [6], SiddharthSharma, SimoneSharma, and AnidhyaAthaiya emphasize the need for an activation function and nonlinearity in neural networks. This study also provides a brief description of various activation functions used in deep learning and describes the importance of activation functions for developing an effective model and improving the performance of artificial neural networks.

Choosing the optimal activation function for a neural network is one of the critical tasks in machine learning. Throughout the field's history, many different activation functions have been proposed, making choosing the optimal one challenging. Over time, new and improved activation functions appear, making the choice even more difficult [7–9].

The choice of activation functions in deep networks has a significant impact on the dynamics of learning and task performance. The most successful and widely used activation function is RectifiedLinearUnit (ReLU). Although various alternatives to ReLU have been proposed, they have yet to be able to replace it due to conflicting benefits. In this paper, the authors propose to use automatic search methods to discover new activation features. The researchers verify the effectiveness of searches by conducting an empirical evaluation with the best-discovered activation function [7].

Activation functions are crucial in deep learning networks because the nonlinear capability of activation functions endows deep neural networks with accurate artificial intelligence. Nonlinear nonmonotonic activation functions such as straightened linear units, hyperbolic tangent (tanh), sigmoid, Swish, Mish, and Logish work well in deep learning models; however, only a few of them are widely used in most applications due to their inconsistency. In [8], a new nonlinear activation function called Smish (SmoothMish) was proposed. Unlike other activation functions, such as ReLU, Smish has a smooth derivative, making it more resistant to vanishing gradients during neural network training. Smish is non-monotonic, which allows it to model more complex nonlinear dependencies. It has a range of output values from –1 to 1. The potential impact of Smish on deep learning and artificial intelligence is significant, inspiring further exploration and research.

This paper proposes a universal activation function (UAF) that achieves near-optimal performance in quantification, classification, and reinforcement learning tasks [9].

Formulation of the goals of the article. The choice of activation function is an important step in designing a neural network. Different activation functions have different properties and can lead to different results. Therefore, the purpose of this study is to provide a detailed overview of various commonly used activation functions.

Outline of the main research material. Understanding and comparing different activation functions used in neural networks is an important research topic. Each activation function has unique properties that affect the dynamics and behavior of the neural network. The activation functions to be considered cover a diverse range, including Sigmoid, Tanh (hyperbolic tangent), ReLU (rectified linear unit), LeakyReLU, ELU (exponential linear unit), SELU (scaled exponential linear unit), Swish, Mish, Softmax.

Activation functions are mathematical algorithms that transform neuronal inputs. In other words, they determine whether a neuron will be active based on its input. There are many different activation functions, each with properties, as shown graphically in Figure 1.

The sigmoidal activation function, or the logistic function, is a nonlinear function widely used in neural networks. It converts input values into a range from 0 to 1 (see Figure 1) and is calculated using formula 1.

$$f(x) = \frac{1}{(1+\exp(-x))}. \qquad (1)$$

The sigmoidal activation function is used in various machine-learning tasks, including classification, prediction, natural language processing, image processing, and others. It has advantages over other activation functions.

The activation function's advantages include simplicity, nonlinearity, and interpretability. The sigmoid function is nonlinear, which allows neural networks to model complex relationships between input data and output values. The function is interpreted because it converts input data into probabilities and is easy to understand and implement.

However, it's essential to be aware of the potential challenges that the sigmoidal activation function can pose. Saturation, for instance, can occur when the input values become very large or very small, leading to a halt in the learning process of neurons. Similarly, the vanishing gradient problem, where the error gradient becomes very small during training, can also hinder the neural network's performance.

The hyperbolic tangent (Tanh) is a mathematical function often used in neural networks and other machine learning fields. It calculates the hyperbolic tangent of x, which can be any actual number.

Tanh (hyperbolic tangent) is similar to the sigmoid function but ranges from –1 to 1 (see Figure 1).

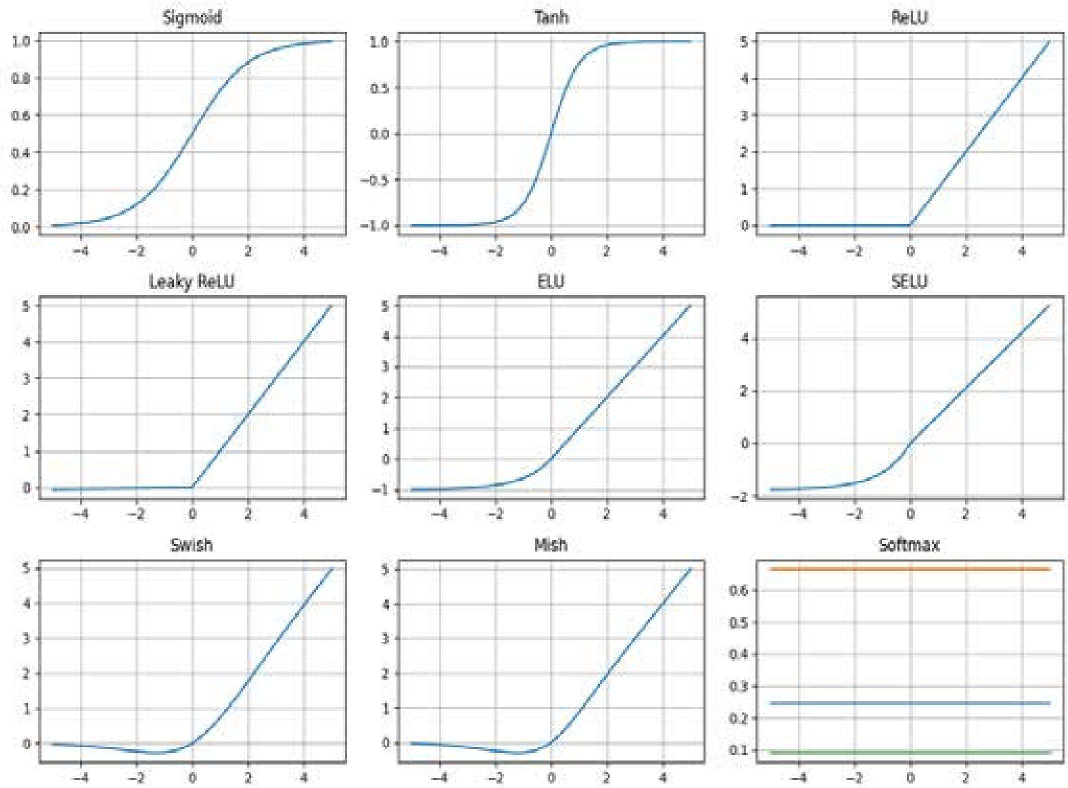

Fig. 1. Graphical representation of activation functions

It is calculated using the formula 2. Tanh is often used as an activation function in hidden layers of neural networks to help prevent the vanishing gradient problem. Tanh can be used to scale or center data.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \qquad (2)$$

The disadvantages of Tanh include incomplete saturation. The function becomes saturated at the edges of the range but never reaches –1 or 1, which means that for large values of x, the function ceases to affect the output significantly.

The hyperbolic tangent function is not symmetric around the origin ($f(x) \neq f(-x)$ for $x \neq 0$). It means that the value of tanh(x) for positive values of x is not equal to that of tanh($-x$) for negative values of x.

Formula (2) contains exponential functions that are not symmetric. The derivative of tanh (Formula 3) shows that the derivative of tanh is always greater than 0 when $x > 0$ and less than 0 when $x < 0$. It means that the graph of tanh increases in the positive direction and decreases in the negative direction.

$$\tanh'(x) = 1 - \tanh^2(x). \qquad (3)$$

ReLU (RectifiedLinearUnit) is a widely used nonlinear activation function in artificial neural networks, especially deep learning architectures. It works by linearly activating input values greater than zero and zeroing out negative values (see Figure 1). The activation function is simple and computationally efficient (Formula 4).

$$f(x) = \max(0, x). \qquad (4)$$

ReLU is simple and easy to understand and implement. Its computation requires few resources. It is usually used in the hidden layers of deep neural networks.

ReLU also has disadvantages. If the neuron receives only negative input values, it will always output 0. It can lead to the neuron becoming “dead” and not participating in training. The problem of the vanishing gradient appears again. When the gradient of the ReLU function is 0 for negative values of x, it can make it challenging to train the neural network. Despite these drawbacks, ReLU is one of the most popular activation functions. It is often used in neural networks for classification, prediction, computer vision, natural language processing, etc. ReLU is also used in generative models such as GAN (Generative Adversarial Networks) to generate images, text, and other data.

LeakyReLU (RectifiedLinearUnit) is a modification of the ReLU activation function used in neural networks. LeakyReLU extends the concept of ReLU by introducing a slight non-zero slope for negative input values, thus alleviating the problem of neuronal "death" that standard ReLU faces (see Figure 1). It also helps prevent the vanishing gradients that can occur when using ReLU.

The formula calculates the activation function of LeakyReLU:

$$f(x) = \max(\alpha x, x)\,, \qquad (5)$$

where x is the input value; α is slope coefficient, which can be 0.1/0.01/0.001

A small slope for negative values helps gradients propagate through the network during training. LeakyReLU can lead to better neural network performance, especially for tasks with a lot of negative data.

The disadvantages include the need to adjust the slope coefficient. Also, LeakyReLU may be a less effective activation function for tasks where the input data is mostly positive.

The LeakyReLU activation function is simple and effective and can help prevent overfitting. It can be used for object recognition in images, natural language processing, and text generation.

The ELU (Exponential Linear Unit) activation function was introduced in 2015 in the article "Fastand Accurate Deep Network Learning by Exponential Linear Units (ELUs)» by Djork-Arne Clevert, Thomas Unterthiner, and Sepp Hochreiter [10].

While ELUs share similarities with ReLUs, they exhibit distinct behavior (see Figure 1). The ELU behaves like a regular ReLU for positive values, preserving the input value. However, for negative values, the ELU's use of an exponential function results in a value closer to zero than that returned by the ReLU. This characteristic can be advantageous in specific deep-learning scenarios.

The ELU activation function (Formula 6) introduces an additional hyperparameter α for tuning, which can be set manually.

$$f(x) = \begin{cases} x, x \geq 0 \\ \alpha(e^{-x} - 1), x < 0 \end{cases}. \qquad (6)$$

ELU can help neural networks learn faster than other activation functions, such as ReLU or sigmoid function. It has stable gradients, which can help prevent the gradient from disappearing during neural network training. Smoothed nonlinearity with improved robustness to noisy input data compared to ReLU.

The disadvantages include relative complexity. ELU is a more complex activation function than ReLU or sigmoid, which can make it more computationally expensive. In addition, ELU has a hyperparameter that needs to be customized, making it difficult to use.

In [11], the authors proposed learning the ELU parameterization to determine the correct activation form at each level in CNN. The results obtained on the MNIST, CIFAR-10/100, and ImageNet datasets using NiN, Overfeat, All-CNN, and ResNet networks show that the proposed parametric ELU (PELU) performs better than the non-parametric ELU.

The Scaled Exponential Linear Unit (SELU) extends the ELU concept by including a normalization term to ensure that activations are self-normalized across the network (see Fig. 1).

SELU has fixed values for the parameters α and λ, guaranteeing self-normalization of the activation function. It means that the output values of SELU are always automatically scaled to the mean and standard deviation (Formula 7).

$$f(x) = \begin{cases} \lambda x, x \geq 0 \\ \lambda\alpha(e^{x} - 1), x < 0 \end{cases}, \qquad (7)$$

where α is the scaling constant; λ is the shift constant.

The Swish activation function, proposed as an alternative to ReLU, introduces smooth nonlinearity while maintaining computational efficiency (see Figure 1). It presents a similar performance to ReLU with potentially smoother activation behavior. Moreover, it introduces nonlinearity without losing computational efficiency.

The Swish activation function is calculated by formula (8).

$$f(x) = x \times sigmoid(x)\,. \qquad (8)$$

The Swish activation function improves neural network performance in tasks such as image recognition, natural language processing, and machine translation. Its smooth gradients can help in the training process. Also, the activation function can be used instead of ReLU or sigmoid without significantly changing the neural network architecture.

The Mish activation function (Mishra's Softmax-based activation function), characterized by a flat curve, offers a smooth alternative to ReLU and its variants (see Fig. 1). D. Mishra proposed the activation function in his study [12], which is calculated by the formula (9).

$$f(x) = x \times \tanh(\ln(1 + e^{x}))\,. \qquad (9)$$

Mish provides smooth, differentiable nonlinearity, making it convenient for use in neural networks. Its continuous derivative over the entire definition domain makes it robust to gradient vanishing. Mish has no upper or lower bound, allowing it to model a wide range of values. It demonstrates competitive performance compared to other activation functions.

The Softmax activation function is used in neural networks to normalize neuronal outputs to a probability distribution. The derivative of the Softmax function is more complex and depends on the cross-entropy loss function used in neural network training. It is used as an output layer activation function in multiclass classification tasks. It converts raw scores into a probability distribution, facilitating the interpretation of model results (see Fig. 1). Mathematically, the activation function is described by formula (10).

$$f(x) = \frac{e_i^x}{\sum_j e_j^x}. \tag{10}$$

Softmax can be used to classify images from a set of possible classes, to classify text documents, to translate text from one language into another by generating probabilities for each possible word in the translation, etc.

We will choose the CIFAR-10 dataset (Canadian Institute for Advanced Research, ten classes) to test the performance and obtain metrics of different activation functions.

The CIFAR-10 dataset is a widely used reference dataset in computer vision. General information about the CIFAR-10 dataset:

– It is a reference dataset widely used in computer vision and machine learning research.

– The dataset consists of 60,000 32x32 color images in 10 classes (airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks), with 6,000 images in each class (5000 training and 1000 test images per class).

– Each image in the CIFAR-10 dataset is labeled with a single class label indicating the object category to which it belongs. The class labels are represented by integers from 0 to 9, corresponding to the ten classes in the dataset.

– The images are of relatively low resolution, which makes them suitable for training and evaluating image classification models.

– Researchers and practitioners use CIFAR-10 to compare the performance of different machine learning methods and algorithms.

Overall, the CIFAR-10 dataset is a valuable resource for training and evaluating image classification models and serves as a standard benchmark for the computer vision community.

To conduct the study, we built a hidden layer neural network model of 128 neurons and systematically evaluated its performance using each previously described activation function. Each activation function was applied to the hidden layer separately, which allowed us to comprehensively compare their effectiveness in influencing the behavior and performance of the network.

During the experiments with the neural network, various metrics were collected from each experiment to evaluate its performance under different activation functions. These metrics included but were not limited to, accuracy. Careful analysis of these metrics provided valuable insights into each activation function's relative strengths and weaknesses in the context of the CIFAR-10 dataset.

In addition, it is worth noting that the number of training epochs was intentionally limited to 10 for each experiment. This intentional limitation was introduced to model scenarios where computational resources or time constraints require shorter training periods. By imposing this restriction, the goal was to assess the readiness of different activation functions to adapt and perform optimally under limited training conditions.

Conducting experiments under such controlled conditions effectively assesses the reliability and generalizability of each activation function within a limited training time. This approach allows one to assess the absolute performance of each activation function and its ability to converge quickly and efficiently in a limited training environment.

The activation functions investigated included sigmoid, hyperbolic tangent, ReLU, leaky ReLU, ELU, SELU, Swish, Mish, and Softmax. Each activation function was tested in a controlled experimental setup with a limited number of epochs set to 10. The results of the experiment are shown in Figure 2.

The experiments revealed significant variations in the performance of neural network models depending on the choice of activation function. In particular, such activation functions as ELU, SELU, and ReLU proved to be the most effective in terms of classification accuracy and convergence speed within the limited ten epochs.

The experimental results emphasize the critical role of activation functions in shaping the behavior and performance of neural networks, especially in image classification tasks. Activation functions such as ELU, SELU, and ReLU demonstrated excellent performance, indicating their effectiveness in mitigating the vanishing gradient problem and promoting faster convergence.

Despite the limited number of epochs, the experiments provide valuable information about

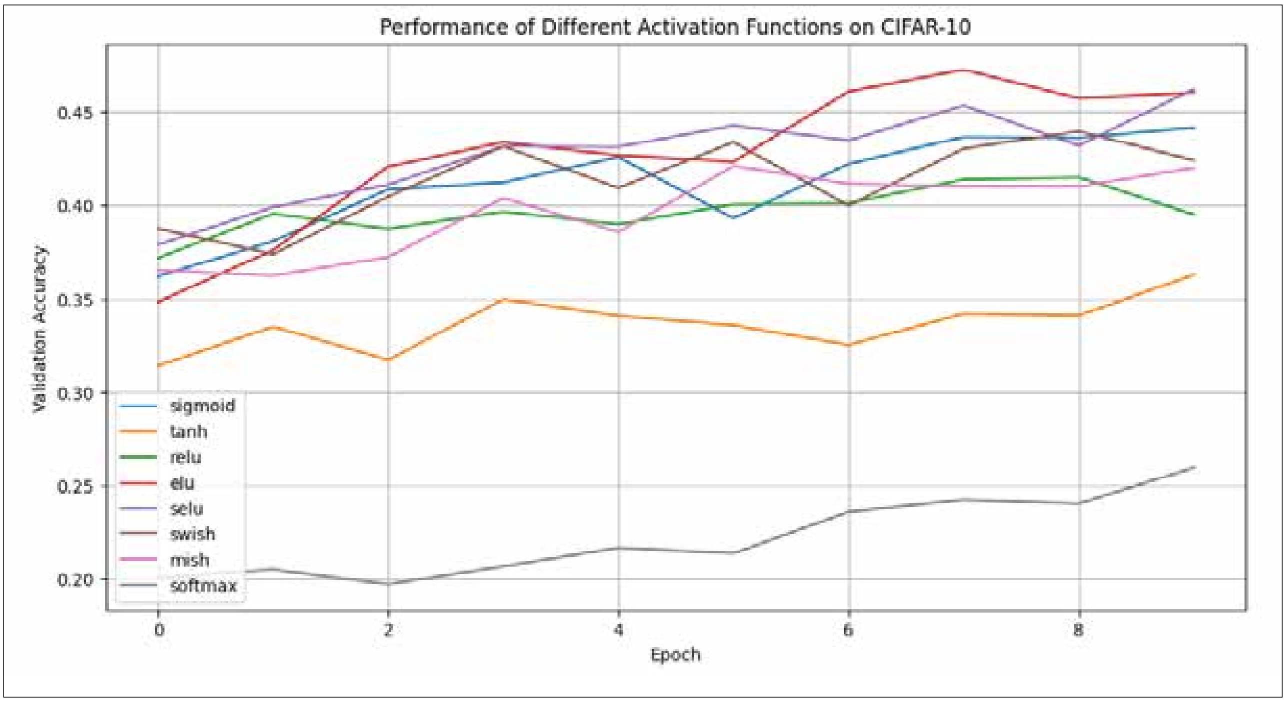

Fig. 2. Results of the experiment

the effectiveness of different activation functions in improving the performance of neural network models on the CIFAR-10 dataset. The results emphasize the importance of selecting appropriate activation functions to optimize the performance and efficiency of neural network models under limited training.

Conclusions. This study investigated the performance of different activation functions in the hidden layers of a neural network using the CIFAR-10 dataset. The activation functions studied were sigmoidal, tangent hyperbolic, ReLU, LeakyReLU, ELU, SELU, Swish, Mish, and Softmax.

Through extensive experiments and analysis, it was found that the choice of activation function significantly affects the neural network's performance. Among the tested activation functions, ELU, SELU, and ReLU proved the most effective in improving network performance on the CIFAR-10 dataset.

In particular, ELU, SELU, and ReLU demonstrated better classification accuracy and convergence speeds than other activation functions. These results align with previous studies that indicate the effectiveness of ELU and SELU in mitigating the vanishing gradient problem and promoting faster convergence in deep neural networks.

The results emphasize the importance of careful activation function selection when designing and training neural networks for image classification tasks. By selecting appropriate activation functions such as ELU, SELU, or ReLU, researchers and practitioners can improve their neural network models' overall performance and efficiency.

Further research can explore the application of these activation functions in other domains and datasets to test their effectiveness in different contexts. In addition, research into new activation functions and optimization methods could lead to further advances in deep learning and neural network design.

Overall, the study makes a valuable contribution to the selection and optimization of activation functions to improve the performance of neural networks, especially in the context of image classification tasks using the CIFAR-10 dataset.

Bibliography:

1. Kamalov F., Nazir A., Safaraliev M., Cherukuri A. K., Zgheib R. Comparative analysis of activation functions in neural networks. 2021. 28th IEEE International Conference on Electronics, Circuits, and Systems (ICECS), Dubai, United Arab Emirates, 2021, pp. 1c6, doi: 10.1109/ICECS53924.2021.9665646.

2. Kaloev M., Krastev G. Comparative Analysis of Activation Functions Used in the Hidden Layers of Deep Neural Networks. 2021. 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 2021, pp. 1–5, doi: 10.1109/HORA52670.2021.9461312.

3. Garrett Bingham, Risto Miikkulainen. Discovering Parametric Activation Functions, Neural Networks, Volume 148, 2022, Pages 48–65, ISSN 0893-6080, https://doi.org/10.1016/j.neunet.2022.01.001

4. Shiv Ram Dubey, Satish Kumar Singh, Bidyut Baran Chaudhuri. Activation functions in deep learning: A comprehensive survey and benchmark, Neurocomputing, Volume 503, 2022, Pages 92–108, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2022.06.111.

5. Andrea Apicella, Francesco Donnarumma, Francesco Isgrò, Roberto Prevete. A survey on modern trainable activation functions, Neural Networks, Volume 138, 2021, Pages 14–32, ISSN 0893-6080, https://doi.org/10.1016/j.neunet.2021.01.026.

6. Sharma S., Sharma S., Athaiya A. Activation functions in neural networks. *International Journal of Engineering Applied Sciences and Technology*, 2020. Vol. 4, Issue 12, ISSN No. 2455-2143, P. 310–316. Published Online April 2020 in IJEAST (http://www.ijeast.com). doi : 10.33564/IJEAST.2020.v04i12.054

7. Ramachandran Prajit, Barret Zoph, Quoc V. Le. Searching for activation functions. arXiv preprint arXiv:1710.05941 (2017)

8. Wang X., Ren H., Wang A. Smish: A Novel Activation Function for Deep Learning Methods. *Electronics,* 2022, 11, 540. https://doi.org/10.3390/electronics11040540

9. Yuen B., Hoang M.T., Dong X. et al. Universal activation function for machine learning. *Sci Rep.* 11, 18757 (2021). https://doi.org/10.1038/s41598-021-96723-8

10. Clevert D.A., Unterthiner T., Hochreiter S., Fast and accurate deep network learning by exponential linear units (elus). arXiv 2015. arXiv preprint arXiv:1511.07289, 2. 2016. https://doi.org/10.48550/arXiv.1511.07289

11. Trottier L., Giguere P., Chaib-draa B. Parametric exponential linear unit for deep convolutional neural networks. 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, 18–21 December 2017. 2017. URL: https://doi.org/10.1109/icmla.2017.00038

12. Misra D. Mish: A self regularized non-monotonic activation function. arXiv preprint arXiv:1908.08681v3 [cs.LG] 2020. https://doi.org/10.48550/arXiv.1908.08681

Морозов А.В., Левківський В.Л., Плечистий Д.Д. ФУНКЦІЇ АКТИВАЦІЇ В НЕЙРОННИХ МЕРЕЖАХ: ОГЛЯД ТА ПОРІВНЯННЯ

Нейронні мережі стали однією з найпотужніших технологій сучасності, що революціонізують безліч галузей. Їхні можливості дозволяють вирішувати складні задачі, які раніше вважалися нездійсненними. Нейронні мережі складаються з взаємозв'язаних вузлів, які називаються нейронами, що передають та обробляють інформацію. Мережі можуть мати різну структуру, від простих з кількома шарами до складних з десятками або навіть сотнями шарів. Кожен шар може містити тисячі або мільйони нейронів, а зв'язки між нейронами можуть бути з різними вагами. Алгоритми навчання нейронних мереж можуть бути складними для налаштування та оптимізації, а процес навчання може займати багато часу, особливо для великих мереж. Функції активації відіграють ключову роль в нейронних мережах, виконуючи декілька важливих функцій, а саме функції активації вводять нелінійність в нейронні мережі, що робить їх здатними вчитися на складних даних та виконувати складні задачі. Крім того, функції активації визначають вихідне значення нейрона, яке може бути інтерпретовано як ймовірність, величина або інший тип значення, залежно від задачі. Нейронні мережі значною мірою залежать від функцій активації, які використовуються в їхніх нейронах і відіграють вирішальну роль у формуванні поведінки та продуктивності нейронних мереж. У статті представлено всебічний огляд і порівняльний аналіз різних функцій активації, які зазвичай використовуються в глибокому навчанні. Глибоке навчання – це гілка машинного навчання, що використовує нейронні мережі з глибокою архітектурою. У роботі досліджуються різні функції активації (Sigmoid, Tanh, ReLU, LeakyReLU, ELU, SELU, Swish, Mish, Softmax), які використовуються в нейронних мережах. Детально описуються їх математичні властивості, переваги та обмеження кожної функції активації, а також емпіричні дані експериментів, проведених на еталонних наборах даних. Проведені дослідження надають уявлення про вибір та оптимізацію функцій активації для архітектури нейронної мережі.

Ключові слова: *функції активації, нейронні мережі, глибоке навчання, Sigmoid, Tanh, ReLU, LeakyReLU, ELU, SELU, Swish, Mish, Softmax.*