International Journal on Perceptive and Cognitive Computing (IJPCC) Vol 10, Issue 2 (2024)
https://doi.org/10.31436/ijpcc.v10i2.490

A Comparative Performance of Different Convolutional Neural Network Activation Functions on Image Classification

Muhammad Zulhazmi Rafiqi Azhary, Amelia Ritahani Ismail*

Department of Computer Science, Kulliyyah of Information and Communication Technology,
International Islamic University Malaysia, Kuala Lumpur, Malaysia

*Corresponding author amelia@iium.edu.my
(Received: 8th June 2024; Accepted: 16th July 2024; Published on-line: 30th July 2024)

Abstract— Activation functions are crucial in optimising Convolutional Neural Networks (CNNs) for image classification. While CNNs excel at capturing spatial hierarchies in images, the activation functions substantially impact their effectiveness. Traditional functions, such as ReLU and Sigmoid, have drawbacks, including the "dying ReLU" problem and vanishing gradients, which can inhibit learning and efficacy. The study seeks to comprehensively analyse various activation functions across different CNN architectures to determine their impact on performance. The findings suggest that Swish and Leaky ReLU outperform other functions, with Swish particularly promising in complicated networks such as ResNet. This emphasises the relevance of activation function selection in improving CNN performance and implies that investigating alternative functions can lead to more accurate and efficient models for image classification tasks.

Keywords— Activation Functions, Convolutional Neural Network , Image Classification

I. Introduction

Convolutional Neural Networks (CNNs) is a common machine learning algorithm used for image classification tasks. Image inputs are suitable for CNNs because of their ability to capture spatial hierarchies through convolutional layers. There are several factors that affect the effectiveness of a CNN in learning complex image patterns and features; this includes the CNN architecture, optimisation algorithms, and hyperparameters such as activation functions [1].

Activation functions are an important factor that impact the performance of neural networks. This is because they introduce non-linearity into the model. As such, they enable the model to learn from complex data and performs machine learning tasks such as classification. Currently, there are various activation functions that have been developed and are available to be used. These activation functions have their own strengths and limitations over each other. The most commonly used activation functions include Rectified Linear Unit (ReLU), Sigmoid, Tanh, Leaky ReLU, Exponential Linear Unit (ELU), and a recently proposed Swish [2][10].

The purpose of this study is to provide an extensive analysis of these activation functions across different CNN architectures on image classification tasks. We intend to discover further on how the selection of activation function affects the effectiveness of CNNs by systematically evaluating the performance of a simple CNN, VGG-like CNN, and ResNet-like CNN models using an array of activation functions. Our evaluations are done on the CIFAR-10 image dataset.

The findings of this study will assist researchers in selecting optimal activation functions for their CNN models. This will result in a more accurate and efficient neural networks for image classification.

II. Related work

This study presents Cone and Parabolic-Cone activation functions, which outperform ReLU and Sigmoidal functions on CIFAR-10 and Imagenette benchmarks. These new functions enable finer input space division, improving accuracy and training speed with fewer neurons [1]. This suggests a potential shift in neural network design, as they provide superior performance and efficiency compared to traditional ReLU and Sigmoidal functions, particularly for complex, non-linear datasets.

This study compares past and current functions, noting that while ReLU excels in classification, it struggles in physics-informed tasks. Alternatives functions like hyperbolic tangent, Swish, and sine, especially adaptive ones, perform better in complex scenarios [2]. This is because they offer smoother gradients and better adaptability for complex and physics-informed problems compared to ReLU, which can struggle with gradient consistency and specific task requirements.

This study introduces the "seagull" activation function, which, when used in layers handling exchangeable variables, greatly improves performance and reduces errors, even for

https://doi.org/10.31436/ijpcc.v10i2.490

high-dimensional data like CIFAR10 [3]. It can notably enhance neural network performance and error reduction, making it a valuable approach for both low and high-dimensional data.

This study surveys activation functions (AFs) in deep learning, covering types like Logistic Sigmoid, Tanh, ReLU, ELU, Swish, and Mish. It reviews their characteristics and compares the performance of 18 AFs across different networks and datasets to help researchers and practitioners choose the best options [4]. The survey highlights that understanding and choosing the right activation function is crucial for optimising neural network performance, as different functions offer distinct advantages depending on the network and dataset.

This study finds that combining ReLU, tanh, and sin activation functions can optimise neural network performance. ReLU is dominant, but initial layers favor ReLU or LeakyReLU, while deeper layers perform better with more convergent functions [5]. This suggests that the practice of optimising activation functions by combining different types can enhance neural network performance, with ReLU being dominant in early layers and more convergent functions benefiting deeper layers.

This study introduces Saturated Gaussian Error Linear Units (SGELU), SSiLU, and SMish, new activation functions that combine ReLU with non-monotonic functions. Experiments on CIFAR-100 show these functions outperform existing activation functions in various deep learning models [6]. This can significantly enhance performance in deep learning models, as demonstrated by their superior results on CIFAR-100.

This study offers an updated overview of popular activation functions, addressing their properties and evolution from traditional ones like logistic and ReLU to newer functions. It serves as a useful resource for understanding and applying activation functions in neural networks [7]. This indicates that understanding the properties and evolution of both traditional and new activation functions is crucial for effectively applying them in neural networks and deep learning.

This study introduces four new oscillatory activation functions that allow neurons to learn functions like XOR and outperform popular functions in classification tasks. These functions also address the vanishing gradient problem, which occurs when gradients become too small during backpropagation, preventing proper weight updates and training. The paper also discusses various activation functions, including the widely used sigmoid function, known for its nonlinearity and computational efficiency [8]. This enhances learning and performance in neural networks by overcoming the vanishing gradient problem, offering improvements over traditional functions like sigmoid.

This study examines how ReLU activation functions contribute to vulnerabilities in deep learning models to adversarial examples. It proposes a modified ReLU function that enhances robustness against such attacks and shows through experiments that this modification, combined with adversarial training, improves model resilience [9]. This suggests that by modifying ReLU activation functions, it can improve deep learning models' resilience to adversarial attacks, and combining this with adversarial training further enhances robustness.

This study compares a CNN using Swish activation (76% accuracy) with one using Adaptive Piecewise Linear activation (74.4%) for skin cancer detection, showing that Swish as a separate layer improves accuracy and reduces loss [10]. This indicates that using Swish activation as a separate layer in a CNN for skin cancer detection improves accuracy and reduces loss compared to using Adaptive Piecewise Linear (APL) activation, demonstrating the effectiveness of Swish in enhancing model performance.

Fig. 1 shows a comparison between six different activation functions used in neural networks: ReLU, Sigmoid, Tanh, Leaky ReLU, ELU, and Swish. These functions are critical for defining how neurons in a neural network activate, which affects the model's learning ability and performance. It depicts how each function converts input values (x-axis) to output values (y-axis). ReLU (blue line) activates only positive inputs and outputs them directly, whereas Sigmoid (green dashed line) and Tanh (red dashed line) squish inputs to a range of (0, 1) and (-1, 1), respectively, yielding smooth gradients but potentially vanishing gradients. Leaky ReLU (purple dotted line) introduces a slight slope for negative inputs, which alleviates the dying ReLU problem.

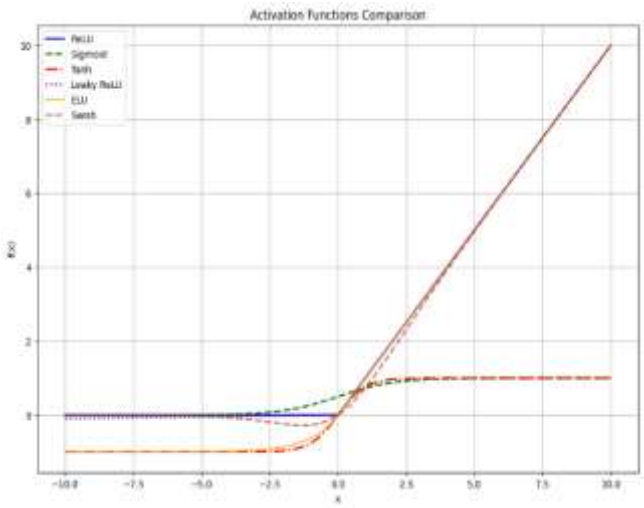

Fig. 1 Activation functions graphs comparison

III. METHODOLOGY

https://doi.org/10.31436/ijpcc.v10i2.490

A. Dataset

The image dataset used in this experiment is CIFAR-10. This dataset consists of 60,000 colour images with an image size of 32 *px* x 32 *px*. It has 10 classes, with 6,000 images per class. It is a common dataset used for image classification algorithms benchmark due to the diversity of its classes and a relatively small image size.

B. Data Preprocessing

The image dataset was transformed to tensors and normalised with a mean and standard deviation of 0.5 for each colour channels.

C. Model Architectures

- Simple CNN: This architecture includes two convolutional layers with the following parameters: filters of [32, 64], kernel size of 3, stride and padding of 1. Then, activation functions are applied to the convolutional layers, followed by a max pooling layer with a kernel size and stride of 2. Finally, a fully connected layer with filters of 512 for classification. It serves as a baseline model to assess basic performance of activations functions.
- VGG-like CNN: This architecture is based on the VGG architecture, it incorporates several convolutional layers with the following parameters: filters of [64, 128, 256], kernel size of 3, and padding of 1. Then, each convolutional blocks are followed by activation functions and max pooling layers with a kernel size of 2 and stride of 2. It ends with three fully connected layers with filter of 512, followed by activation functions for the first two layers. This architecture aims to capture more complex features and evaluate the impact of activation functions in deeper networks.
- ResNet-like CNN: This architecture is based on the ResNet architecture, it includes residual blocks that allow for deep networks by addressing the vanishing gradient problem through skip connections. The convolutional layers in the model have filters of [64, 128, 256, 512] with a kernel size of 3, stride and padding of 1. Then, followed by batch normalisation layers and activation functions. Before passing into the final fully connected layer, the output wen through an average pooling layer. This architecture is used to investigate the performance of activation functions in very deep networks.

D. Activation Functions

- ReLU: It outputs the input directly if it is positive; otherwise, it outputs zero. While it is effective in many situations, it can suffer from the "dying ReLU" issue, where neurons can become inactive and stop learning if they consistently receive negative inputs.
- Sigmoid: It outputs values between 0 and 1, making it useful for binary classification tasks as it can be interpreted as a probability. However, in deep networks, it often encounters vanishing gradients due to its tendency to saturate at the extremes, which can slow down the learning process.
- Tanh: It outputs values between -1 and 1, providing stronger gradients compared to sigmoid. This helps with learning in deeper networks, but it still suffers from gradient saturation at its extreme values, which can hinder training speed and effectiveness.
- Leaky ReLU: It addresses the "dying ReLU" issue by allowing a small, non-zero gradient when the input is negative. This adjustment helps prevent neurons from becoming inactive, maintaining learning efficiency while preserving the simplicity and speed of ReLU.
- ELU: It offers a smooth gradient for negative inputs, which helps mitigate the vanishing gradient issue. It allows for a more gradual learning curve by providing a small gradient when the input is negative, which can lead to faster and more stable training.
- Swish: It combines the input with a sigmoid function applied to that input, creating a smooth and non-monotonic activation function. This characteristic often results in better performance and training efficiency compared to tradition functions like ReLU, especially in deeper networks.

E. Experimental Setup

Each CNN architecture was trained using each activation function on CIFAR-10 dataset. All models were trained using the same set of hyperparameters as shown in Table I. This is done to keep the differences in results solely dependent to the changes of activation function used in the training; hence, ensuring a fair comparison.

TABLE I
CONSTANT HYPERPARAMETERS VALUES FOR EXPERIMENT

	Hyperparameter
Optimizer	Adam
Criterion	Cross-Entropy
Learning rate	0.001
Batch size	64
Epochs	15

F. Performance Evaluation Metrics

Results from this experiment is evaluated using a test set consisting of 10,000 images from CIFAR-10 with the following metrics: accuracy and loss. Accuracy would indicate the overall effectiveness of the model in predicting correct labels, while the loss value computed using the cross-entropy loss function providing insights how well the predictions of the model align with the true labels.

https://doi.org/10.31436/ijpcc.v10i2.490

IV. Results and Discussion

The results obtained from the experiment provide an extensive evaluation of the impact of different activation functions on three distinct CNN architectures: Simple CNN, VGG-like CNN, and ResNet-like CNN, with the CIFAR-10 dataset. The Adam optimiser, cross-entropy loss, a batch size of 64, 15 epochs, and a learning rate of 0.001 are the constant hyperparameters used throughout every experiment. These constant settings enable a fair comparison of the activation functions' performance.

Table II shows that ReLU, Tanh, and Leaky ReLU achieved reasonably high test accuracies of 70.96%, 71.30%, and 72.57%, respectively with the simple CNN. Despite the constant hyperparameters, all activation functions demonstrated varying levels of success, most likely due to their unique mathematical features. Leaky ReLU slightly beat the others in terms of test accuracy, whereas Tanh had the smallest train loss of 0.0017, indicating efficient training. However, Sigmoid and ELU performed poorly, with Sigmoid getting the lowest accuracy of 66.88% and ELU having the largest test loss of 2.382. Swish performed moderately, with a test accuracy of 71.58%, suggesting a modest improvement over ReLU but a greater test loss.

Table III indicates that Swish and Leaky ReLU outperformed other VGG-like CNNs, with test accuracies of 75.84% and 78.67%, respectively. Swish's smooth and non-monotonic characteristics resulted in a considerably decreased train loss, indicating effective gradient flow and better convergence. This performance highlights the potential of novel activation functions such as Swish in deeper networks. However, Sigmoid underperformed significantly, with a test accuracy of 10.00%, showing that it failed to train successfully in this more complex design, most likely due to issues such as vanishing gradients and ineffective convergence.

Table IV shows that ReLU and Swish had the highest test accuracies of 84.03% and 84.43%, respectively for the ResNet-like CNN. Swish outperforming ReLU marginally in terms of train and test losses. The residual connections of ResNet-like architectures take advantage of Swish's characteristics, resulting in greater performance. ELU also produced competitive results, with an accuracy of 83.14% and minimal train and test losses. Sigmoid continues to underperform, supporting the argument that it may not be appropriate for such architectures.

Across all three CNN architectures, the consistent performance of ReLU and its variations (Leaky ReLU and ELU) demonstrates their dependability in different situations. Swish emerged as a formidable competitor, especially in complex architectures such as ResNet, where it achieved the best overall performance. The significant underperformance of Sigmoid in the VGG-like CNN, with a test accuracy of only 10.00%, implies that its fundamental constraints, such as gradient saturation and slower convergence, are not properly compensated by the constant hyperparameters employed.

TABLE II
PERFORMANCE EVALUATION RESULTS FOR SIMPLE CNN

	Test accuracy	Train loss	Test loss
ReLU	70.96%	0.0536	1.9209
Sigmoid	66.88%	0.2307	1.0959
Tanh	71.30%	0.0017	1.4290
Leaky ReLU	72.57%	0.0333	1.9569
ELU	70.03%	0.0665	2.3822
Swish	71.58%	0.0401	2.2202

TABLE III
PERFORMANCE EVALUATION RESULTS FOR VGG-LIKE CNN

	Test accuracy	Train loss	Test loss
ReLU	78.22%	0.1344	1.0914
Sigmoid	10.00%	2.3027	2.3026
Tanh	67.33%	0.8008	0.9240
Leaky ReLU	78.67	0.1109	1.0210
ELU	76.50%	0.1799	1.4022
Swish	75.84	0.1334	1.2459

TABLE IV
PERFORMANCE EVALUATION RESULTS FOR RESNET-LIKE CNN

	Test accuracy	Train loss	Test loss
ReLU	84.03%	0.0491	0.7939
Sigmoid	52.41%	0,3399	1.9650
Tanh	76.72%	0.0794	1.1109
Leaky ReLU	84.24%	0.0453	0.8017
ELU	83.48%	0.0756	0.6840
Swish	84.43%	0.0445	0.7374

The choice of activation function is crucial for optimising CNN performance on the CIFAR-10 dataset. Fig. 2, Fig. 3 and Fig.4 shows that while traditional functions like ReLU and its derivatives continue to be useful, emerging functions such as Swish shows promise, particularly in more complex architectures. This study emphasises the significance of experimenting with different activation functions to obtain optimal performance in specific CNN designs.

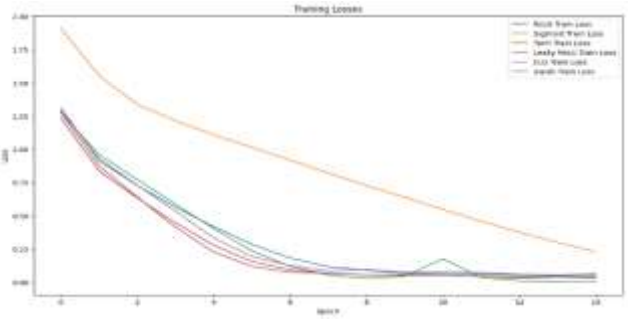

Fig. 2 Simple CNN varied activation functions training losses

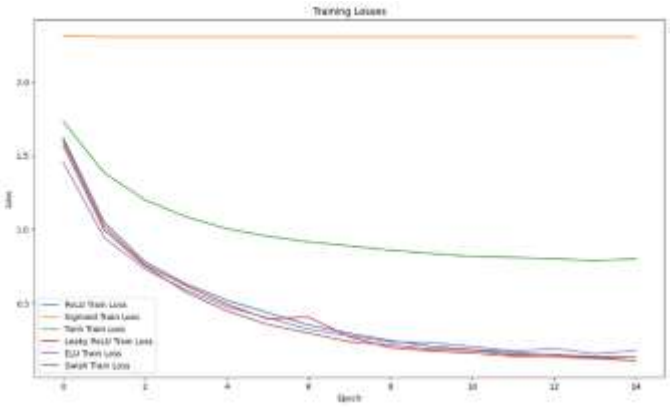

Fig. 3 VGG-like CNN varied activation functions training losses

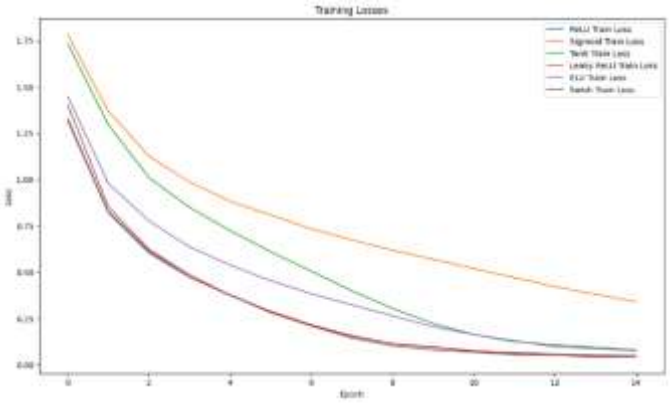

Fig. 4 ResNet-like CNN varied activation functions training losses

V. CONCLUSION

In conclusion, this comparative analysis of activation functions on Simple CNN, VGG-like CNN, and ResNet-like CNN architectures using the CIFAR-10 dataset demonstrates the importance of activation function selection for model performance. Despite constant hyperparameters (Adam optimiser, cross-entropy loss, batch size of 64, 15 epochs, and learning rate of 0.001), activation functions such as Leaky ReLU and Swish performed better than others. Swish outperformed Sigmoid overall, particularly in complex architectures such as ResNet. This study emphasises the need of experimenting with various activation functions to improve CNN performance for specific tasks.

ACKNOWLEDGMENT

The authors hereby acknowledge the review support offered by the IJPCC reviewers who took their time to study the manuscript and find it acceptable for publishing.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

REFERENCES

[1] M. Mathew, M. Noel, and Y. Oswal, "A significantly better class of activation functions than ReLU like activation functions," *arXiv*, 2024. [Online]. Available: https://doi.org/10.48550/arxiv.2405.04459.

[2] A. D. Jagtap and G. E. Karniadakis, "How important are activation functions in regression and classification? A survey, performance comparison, and future directions," *Journal of Machine Learning for Modelling and Computing*, vol. 4, no. 1, pp. 21-75, 2023.

[3] F. Gao and B. Zhang, "Data-aware customisation of activation functions reduces neural network error," *arXiv*, 2023. [Online]. Available: https://doi.org/10.48550/arxiv.2301.06635.

[4] D. Sukau, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92-108, 2022.

[5] V. Bansal, "Activation Functions: Dive into an optimal activation function," *arXiv*, 2022. [Online]. Available: https://doi.org/10.48550/arxiv.2202.12065.

[6] J. Chen and Z. Pan, "Saturated Non-Monotonic Activation Functions," *arXiv*, 2023. [Online]. Available: https://doi.org/10.48550/arxiv.2305.07537.

[7] J. Lederer, "Activation Functions in Artificial Neural Networks: A Systematic Overview," *arXiv*, 2021. [Online]. Available: https://arxiv.org/abs/2101.09957.

[8] P. Liu, "A survey on recently proposed activation functions for Deep Learning," *arXiv*, 2022. [Online]. Available: https://doi.org/10.48550/arxiv.2204.02921.

[9] S. Korn, G. Hamerly, and P. Rivas, "Is ReLU Adversarially Robust?," *arXiv*, 2024. [Online]. Available: https://doi.org/10.48550/arxiv.2405.03777.

[10] M. F. R. Mariam, M. F. Farheen, M. M. Manjushree, and M. K. Pandit, "Skin Cancer Detection using CNN with Swish Activation Function," *International Journal of Engineering Research and Technology*, vol. 8, no. 14, 2020.