2026 Volume: 02 Issue: 04 www.ijrpa.com ISSN 2456-9995 Research Article

International Journal Research Publication Analysis

Page: 01-07

EFFECT OF ACTIVATION FUNCTIONS ON VANISHING GRADIENTS

***[1]Dr. Ramya BN [2]Bindu S., [3]Varshini Venkatesh Adabaddi**

[1]Associate Professor, Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.

[2,3]Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.

Article Received: 17 March 2026
Article Revised: 07 April 2026
Published on: 27 April 2026

***Corresponding Author: Dr. Ramya BN**
Associate Professor, Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.

DOI: https://doi-doi.org/101555/ijrpa.7195

ABSTRACT

Deep neural networks have achieved significant success in various domains such as image processing, natural language processing, and pattern recognition. However, training deep networks is often hindered by the vanishing gradient problem, where gradients become extremely small during backpropagation, slowing or completely stopping learning. One of the major factors influencing this issue is the choice of activation function. This survey paper analyzes the effect of different activation functions, such as Sigmoid, Tanh, ReLU, and their variants, on the vanishing gradient problem. It highlights how traditional activation functions lead to gradient decay, while modern activation functions help maintain gradient flow. The study also reviews recent advancements in hybrid activation functions designed to overcome these limitations. The results indicate that selecting appropriate activation functions significantly improves training efficiency, convergence speed, and overall model performance.

I. INTRODUCTION

Deep learning models rely on backpropagation to update weights and learn patterns from data. During this process, gradients are propagated from the output layer to earlier layers. However, in deep networks, gradients can become extremely small, especially when certain activation functions are used. This phenomenon is known as the vanishing gradient problem. Activation functions play a crucial role in introducing non-linearity into neural networks. Early neural networks primarily used Sigmoid and Tanh functions due to their smooth and

continuous nature. However, these functions suffer from saturation, where their derivatives become very small for large input values. As a result, gradients diminish rapidly when propagated across multiple layers, making it difficult for the network to learn effectively To address this issue, new activation functions such as ReLU (Rectified Linear Unit) were introduced. These functions maintain larger gradients and allow faster learning. This paper explores how different activation functions affect gradient flow and compares their performance in mitigating vanishing gradients.

II. METHODOLOGY

This survey is based on a comparative analysis of different activation functions and their impact on gradient flow in deep neural networks. The methodology includes:

- Reviewing existing research papers on activation functions and vanishing gradients
- Analyzing mathematical properties of activation functions
- Comparing gradient behavior across multiple layers
- Evaluating performance based on convergence speed and accuracy

The study focuses on the following activation functions:

- **Sigmoid Function:** Maps values between 0 and 1 but suffers from gradient saturation
- **Tanh Function:** Maps values between -1 and 1, reducing gradient issues compared to sigmoid but still prone to vanishing gradients
- **ReLU Function:** Outputs zero for negative inputs and linear for positive inputs, maintaining stronger gradients
- **Advanced Functions:** Leaky ReLU, ELU, and hybrid functions designed to overcome limitations

The comparison is based on how gradients behave during backpropagation and how effectively each function supports deep learning.

III. SYSTEM ARCHITECTURE AND DATA FLOW

The system architecture of a deep neural network consists of multiple layers:

1. **Input Layer:** Receives input data
2. **Hidden Layers:** Apply weights, biases, and activation functions
3. **Output Layer:** Produces final predictions

Data Flow Process:

- Input data is fed into the network
- Each layer processes the data using weights and activation functions
- The output is generated at the final layer
- During backpropagation, gradients are computed and propagated backward

Role of Activation Functions:

Activation functions are applied at each hidden layer. Their derivatives directly influence gradient flow:

- In **Sigmoid/Tanh**, derivatives are small → gradients shrink
- In **ReLU**, derivatives remain large (for positive inputs) → gradients preserved.

During backpropagation, gradients are multiplied across layers. If each layer produces small derivatives (e.g., sigmoid ≈ 0.25), the gradient reduces exponentially, causing vanishing gradients

IV. RESULT AND DISCUSSION

Sigmoid Function

- Suffers from saturation at extreme values
- Gradients approach zero → slow learning
- Not suitable for deep networks

Tanh Function

- Better than sigmoid due to zero-centered output
- Still suffers from vanishing gradients in deep layers

ReLU Function

- Maintains gradient for positive inputs
- Enables faster training and convergence
- Widely used in modern deep learning

Advanced Activation Functions

- Leaky ReLU and ELU reduce "dying neuron" problem
- Hybrid functions (e.g., RSigELU) combine benefits of multiple functions
- Provide stable gradient flow and improved performance

Key Findings:

- ReLU-based models show better accuracy and stable gradients

- Sigmoid-based models experience severe vanishing gradient issues
- Combining activation functions with optimizers (e.g., Adam) improves results.

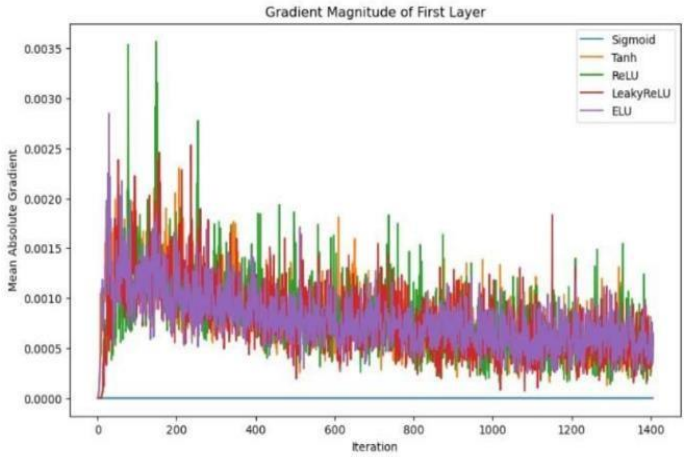

Figure 1: Gradient Magnitude of First Layer.

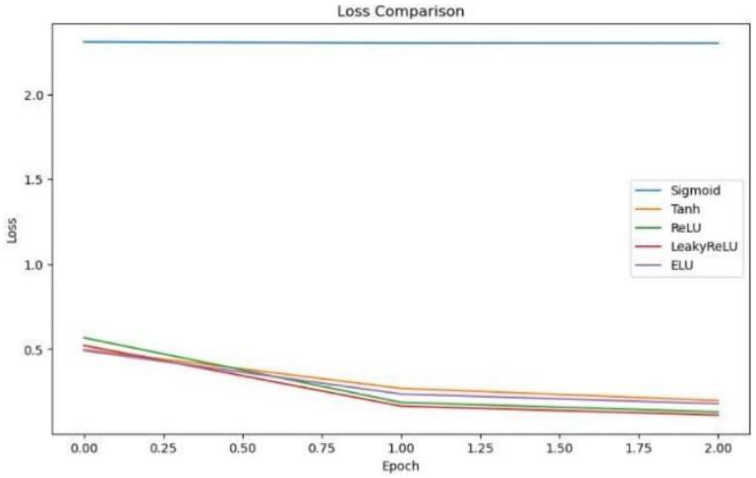

Figure 2: Loss Comparison.

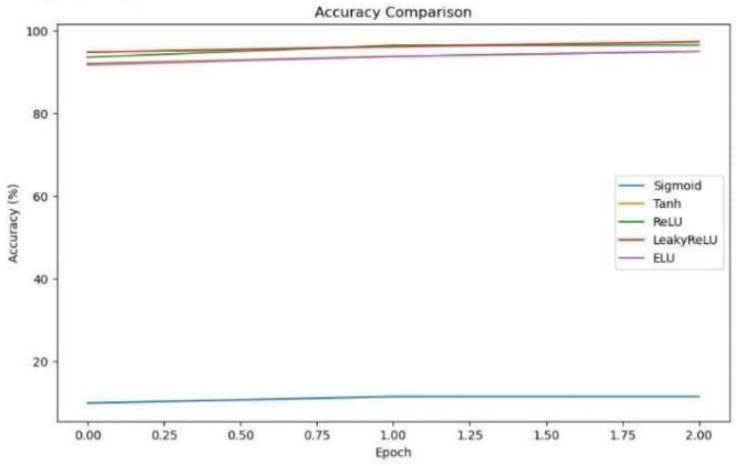

Figure 3: Accuracy Comparison.

```
Using device: cpu
Dataset loaded ✅
Model defined ✅
Training function ready ✅

Training with Sigmoid activation...
Epoch [1/3] - Loss: 2.3095, Accuracy: 9.80%
Epoch [2/3] - Loss: 2.3025, Accuracy: 11.35%
Epoch [3/3] - Loss: 2.3015, Accuracy: 11.35%

Training with Tanh activation...
Epoch [1/3] - Loss: 0.4970, Accuracy: 91.63%
Epoch [2/3] - Loss: 0.2680, Accuracy: 93.73%
Epoch [3/3] - Loss: 0.1963, Accuracy: 94.93%

Training with ReLU activation...
Epoch [1/3] - Loss: 0.5662, Accuracy: 93.53%
Epoch [2/3] - Loss: 0.1854, Accuracy: 96.38%
Epoch [3/3] - Loss: 0.1293, Accuracy: 96.47%

Training with LeakyReLU activation...
Epoch [1/3] - Loss: 0.5208, Accuracy: 94.74%
Epoch [2/3] - Loss: 0.1639, Accuracy: 96.08%
Epoch [3/3] - Loss: 0.1112, Accuracy: 97.28%

Training with ELU activation...
Epoch [1/3] - Loss: 0.4908, Accuracy: 91.98%
Epoch [2/3] - Loss: 0.2349, Accuracy: 93.71%
Epoch [3/3] - Loss: 0.1777, Accuracy: 94.97%
Training completed ✅
```

V. CONCLUSION

The vanishing gradient problem remains a critical challenge in training deep neural networks. This survey highlights that activation functions play a major role in either causing or mitigating this issue. Traditional activation functions like Sigmoid and Tanh lead to gradient decay due to their saturation properties, making them less suitable for deep architectures.

Modern activation functions such as ReLU and its variants have significantly improved training efficiency by maintaining gradient flow. Additionally, recent research on hybrid activation functions shows promising results in overcoming both vanishing gradients and other limitations.

In conclusion, selecting an appropriate activation function is essential for building efficient deep learning models. Future research should focus on developing adaptive and hybrid activation functions that can further enhance gradient stability and overall model performance.

VI. ACKNOWLEDGMENT

I would like to express my sincere gratitude to all those who have supported and guided me throughout the completion of this survey paper on *"Effect of Activation Functions on Vanishing Gradients."*

First and foremost, I would like to thank my faculty guide for their valuable guidance, continuous encouragement, and insightful suggestions, which helped me understand the topic in depth and complete this work successfully.

I am also thankful to my institution and department for providing the necessary resources, environment, and support required for carrying out this study. Their

encouragement played a significant role in enhancing my knowledge in the field of deep learning and artificial intelligence.

I would like to extend my gratitude to the authors and researchers whose work I referred to while preparing this survey. Their contributions provided a strong foundation and helped me gain a clear understanding of activation functions and their impact on neural network performance.

REFERENCES

1. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016. (Covers vanishing gradients and activation functions in detail)
2. X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Proceedings of AISTATS*, 2010. https://proceedings.mlr.press/v9/glorot10a.html
3. V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," *Proceedings of ICML*, 2010. https://www.cs.toronto.edu/~hinton/absps/reluIC ML.pdf
4. D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," *ICLR*, 2016. https://arxiv.org/abs/1511.07289
5. A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," *ICML Workshop*, 2013. https://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf
6. K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *ICCV*, 2015. https://arxiv.org/abs/1502.01852
7. S. Hochreiter, "The Vanishing Gradient Problem During Learning Recurrent Neural Nets

and Problem Solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 1998.

8. GeeksforGeeks, "Tanh vs Sigmoid vs ReLU," 2024. https://www.geeksforgeeks.org/deep-learning/tanh-vs-sigmoid-vs-relu/
9. J. Brownlee, "A Gentle Introduction to the Vanishing Gradient Problem," *Machine Learning Mastery*, 2019. https://machinelearningmastery.com/vanishing-gradients-activation-function/
10. S. Ramachandran, B. Zoph, and Q. V. Le, "Searching for Activation Functions," *arXiv preprint*, 2017. https://arxiv.org/abs/1710.05941