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

International Journal Research Publication Analysis

Page: 01-04

WEIGHT INITIALIZATION TECHNIQUES TO REDUCE VANISHING GRADIENTS

[*1]Dr Ramya B. N. , [2]Sankhya Hegde, [3]Shrilakshmi. T. Hegde

[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 B. N.**
Associate Professor, Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India. DOI: https://doi-doi.org/101555/ijrpa.1373

ABSTRACT

The training of deep neural networks is often affected by the vanishing gradient problem, where gradients become extremely small during backpropagation, resulting in ineffective learning in earlier layers. This issue leads to slow convergence and reduced model performance. In this work, an experimental study of different weight initialization techniques is presented to address this challenge. The techniques considered include Random initialization, Xavier initialization, He initialization, and Orthogonal initialization. A feedforward neural network is implemented and trained on the MNIST dataset of handwritten digits to evaluate the impact of these initialization methods. The performance of each method is analyzed using training loss and test accuracy. The experimental results demonstrate that appropriate weight initialization significantly improves gradient flow, accelerates convergence, and enhances model accuracy. Among the evaluated methods, He initialization provides the best performance, particularly for networks using ReLU activation functions.

I. INTRODUCTION

Deep learning has emerged as a powerful approach for solving complex problems in fields such as image classification, speech recognition, and natural language processing. Despite its success, training deep neural networks remains challenging due to issues such as vanishing gradients. The vanishing gradient problem occurs when gradients shrink exponentially as they propagate backward through layers during training, preventing effective updates to the weights of earlier layers. This results in slow learning or even failure to train deep networks.

To overcome this problem, various techniques have been proposed, among which weight initialization plays a crucial role. Proper initialization ensures that the variance of activations and gradients is maintained across layers, enabling stable and efficient training. This project focuses on analyzing different weight initialization techniques and studying their impact on the performance of a neural network model.

II. STRUCTURE OF THE SYSTEM

The proposed system is designed to evaluate the effectiveness of different weight initialization techniques in reducing the vanishing gradient problem. The system consists of several stages including data preprocessing, neural network modeling, weight initialization, training, and evaluation. Initially, the MNIST dataset is loaded and preprocessed by converting images into tensor format suitable for training. The neural network model is then defined as a feedforward architecture with multiple layers.

The model consists of an input layer corresponding to flattened image pixels, followed by two hidden layers with ReLU activation functions, and an output layer for classification. Before training, the weights of the network are initialized using different techniques such as Random, Xavier, He, and Orthogonal initialization. The model is then trained using backpropagation, and its performance is evaluated on the test dataset. This structured approach allows a clear comparison of how different initialization methods influence learning behavior and model accuracy.

III. METHODOLOGY

The methodology of the proposed system involves training a neural network using different weight initialization techniques and comparing their performance. The MNIST dataset, which contains grayscale images of handwritten digits, is used for training and evaluation. Each image is resized into a one-dimensional vector and fed into the neural network.

The training process is carried out using the Adam optimizer with a learning rate of 0.001. The CrossEntropyLoss function is used as the loss criterion for multi-class classification. The model is trained for multiple epochs with a batch size of 64.

During training, the network learns to classify input images into one of ten digit classes.
To evaluate performance, training loss and test accuracy are recorded for each epoch. The entire process is repeated for each initialization method, and the results are compared to

analyze the effectiveness of each technique in mitigating the vanishing gradient problem.

IV. IMPLEMENTATION

The implementation of the proposed system is carried out using the Python programming language with the PyTorch deep learning framework. The MNIST dataset is loaded using torchvision libraries, and data loaders are used to efficiently manage training and testing batches. The neural network is implemented using the nn.Module class, allowing modular definition of layers and forward propagation.

A custom weight initialization function is defined to apply different initialization techniques to the model. These include normal random initialization, Xavier uniform initialization, He initialization for ReLU activations, and orthogonal initialization. The training loop performs forward propagation, loss computation, backward propagation, and weight updates iteratively. The evaluation phase computes the accuracy of the model on unseen test data.

V. RESULTS AND DISCUSSION

The performance of the neural network is evaluated for different weight initialization techniques using accuracy and loss metrics. The results clearly indicate that initialization plays a significant role in determining the efficiency of training. Random initialization leads to slower convergence and lower accuracy due to unstable gradient flow. Xavier initialization provides improved stability and moderate performance by maintaining variance across layers. He initialization shows the best performance among all methods, achieving faster convergence and higher accuracy. This is because it is specifically designed for ReLU activation functions, ensuring proper scaling of weights. Orthogonal initialization also provides stable training but performs slightly slower compared to He initialization.

The graphical results obtained from the experiments illustrate the variation of accuracy across epochs for each initialization method. As shown in Fig. 1, He initialization consistently achieves the highest accuracy, while Random initialization lags behind. These observations confirm that proper weight initialization is essential for efficient deep learning.

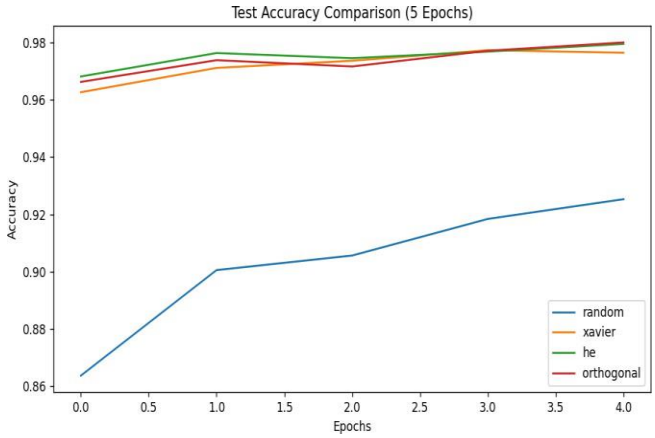

Fig. 1 – Test Accuracy Comparison across different initialization techniques.

VI. CONCLUSION

This study highlights the importance of weight initialization in deep learning. Proper initialization techniques significantly reduce the vanishing gradient problem and improve model performance. Among all methods, He initialization performs best for ReLU-based networks, making it the most suitable choice for modern deep learning models.

VII. ACKNOWLEDGEMENT

I would like to thank our guide, Dr. Ramya B N, for their continuous support and guidance throughout this project.

REFERENCES

1. Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep neural networks.
2. He, K., et al. (2015). Delving deep into rectifiers.
3. Goodfellow, I., et al. (2016). Deep Learning. MIT Press
4. PyTorch Documentation,