Academic Journal of Natural Science
Journal Home: ajns.suaspress.org | CODEN: AJNSAE | NAAN: 40704
Vol. 1, No. 1, 2024

Weight Initialization Methods in Convolutional Neural Networks and Their Impact on Training Efficiency

ZHANG, Zhenxing [1*]

[1] Belarusian State University, Belarus

**** ZHANG, Zhenxing is the corresponding author, E-mail: zhang-zhenxing@outlook.com***

Abstract: This study analyzes the application effects of different weight initialization methods in convolutional neural networks, focusing on the impact of these methods on training efficiency. Through analysis and derivation from existing studies, we expect significant differences in training speed, stability, and final performance among different initialization methods. Specifically, this study compared random initialization, Xavier initialization, He initialization, LeCun initialization, and Orthogonal initialization. The results show that choosing an appropriate weight initialization method can significantly improve the training efficiency and performance of the model. The main contribution of this article is to systematically compare several common weight initialization methods and propose performance expectations based on existing research to provide a reference for subsequent research. Future research can verify these expected results through actual experiments and explore novel initialization strategies to further improve the training efficiency and performance of the model.

Keywords: Convolutional Neural Networks, Weight Initialization, Training Efficiency, Deep Learning, Performance Analysis.

DOI: https://doi.org/10.5281/zenodo.13755632 **ARK:** https://n2t.net/ark:/40704/AJNS.v1n1a02

1 INTRODUCTION

Convolutional Neural Networks (CNNs), as an important model of deep learning, have achieved remarkable results in image recognition, natural language processing and other fields. Weight initialization method, as a key step in the CNN training process, directly affects the convergence speed and final performance of the model. Studying the impact of different weight initialization methods on training efficiency not only helps to improve the training speed of the model, but also provides more efficient solutions for practical applications. This study aims to systematically analyze and compare the application effects of different weight initialization methods in convolutional neural networks, focusing on the impact of these methods on training efficiency. Specific research issues include: the theoretical basis and implementation methods of different weight initialization methods; the impact of various initialization methods on CNN training efficiency; the selection of the optimal weight initialization method and its application scenarios.

2 LITERATURE REVIEW

2.1 Basic Concepts and Development of Convolutional Neural Networks

Convolutional neural networks (CNNs) are a type of deep learning model specifically designed to process data with a grid structure. They were first proposed by LeCun et al. in the 1980s. CNNs achieve efficient feature extraction and classification of image, video and other data through a combination of convolutional layers, pooling layers and fully connected layers. In recent years, with the improvement of computing power and the emergence of large-scale data sets, CNNs have made significant progress in computer vision, natural language processing and other fields.

2.2 Classification and Comparison of Weight Initialization Methods

Weight initialization refers to the process of assigning an initial value to each weight in the network before training the neural network. Common weight initialization methods include:

Random initialization: The simplest method, which randomly generates weights through uniform distribution or normal distribution.

Xavier initialization: Proposed by Glorot and Bengio, it aims to keep the variance of input and output consistent.

He initialization: Proposed by He et al., suitable for ReLU activation function, adjusting the initial value of weight by scaling factor.

LeCun initialization: Designed for Sigmoid activation function, adjusting the initial value of weight by scaling factor.

Orthogonal initialization: Initialize weights by generating orthogonal matrices to ensure independence between different layers.

2.3 Related research results and current status analysis

In recent years, research on weight initialization methods has continued to deepen. The He initialization method proposed by He et al. (2015) performed well in deep neural networks, significantly improving training speed and model performance. Sutskever et al. (2013) experimentally verified the impact of different initialization methods on training efficiency and found that appropriate initialization methods can accelerate convergence and avoid gradient vanishing or exploding problems. In addition, the latest research has also explored adaptive initialization methods and meta-learning-based initialization strategies to further improve the training efficiency and generalization ability of the model.

3 THEORETICAL BASIS

3.1 Mathematical model of convolutional neural network

Convolutional neural network (CNN) is a deep learning model specially designed for processing data with grid structure. Its core components include convolution layer, pooling layer and fully connected layer. Convolution layer extracts local features through convolution operation, pooling layer reduces the size of feature map by downsampling, and fully connected layer maps the extracted features to output space. Mathematically, convolution operation can be expressed as:

$$y_{i,j} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{i+m,j+n} \cdot w_{m,n} + b$$

Among them, x represents the input feature map, w represents the convolution kernel, b represents the bias term, and y represents the output feature map.

3.2 Theoretical analysis of weight initialization methods

Weight initialization is a key step in the neural network training process, which directly affects the convergence speed and performance of the model. Common weight initialization methods include:

Random initialization: randomly generate weights through uniform distribution or normal distribution, the formula is:

$$w \sim \mathcal{U}(-a, a) \mid w \sim \mathcal{N}(0, \sigma^2)$$

Xavier initialization: Keep the variance of input and output consistent. The formula is:

$$w \sim \mathcal{U}(-\sqrt{\frac{6}{n_{in} + n_{out}}}, \sqrt{\frac{6}{n_{in} + n_{out}}})$$

He initialization: Applicable to the ReLU activation function, the initial value of the weight is adjusted by the scaling factor, the formula is:

$$w \sim \mathcal{N}(0, \frac{2}{n_{in}})$$

LeCun initialization: Designed for Sigmoid activation function, the formula is:

$$w \sim \mathcal{N}\left(0, \frac{1}{n_{in}}\right)$$

Orthogonal initialization: Initialize weights by generating an orthogonal matrix to ensure independence between different layers.

3.3 Evaluation indicators of training efficiency

Training efficiency is an important indicator for measuring the performance of neural networks, and is usually evaluated through the following aspects:

Convergence speed: the number of training iterations required for the model to reach the predetermined accuracy.

Computational complexity: the computing resources required during training, including time and memory.

Stability: whether the model has gradient vanishing or explosion during training.

Generalization ability: the performance of the model on unseen data, that is, the accuracy or loss value on the test set.

4 EXPERIMENTAL DESIGN AND THEORETICAL METHODS

4.1 Dataset selection and preprocessing

In order to test different effects, we can try to use the CIFAR-10 dataset for research. The dataset contains 60,000 32x32 color images divided into 10 categories. The dataset is divided into 50,000 training images and 10,000 test images. The data preprocessing steps include image normalization and data augmentation to improve the generalization ability of the model.

4.2 Specific implementation of weight initialization methods

This study will compare the following weight initialization methods:

Academic Journal of Natural Science
Journal Home: ajns.suaspress.org | CODEN: AJNSAE | NAAN: 40704
Vol. 1, No. 1, 2024

Random initialization: randomly generate weights through uniform distribution or normal distribution.

Xavier initialization: Keep the variance of input and output consistent.

He initialization: Applicable to ReLU activation function, adjust the initial value of weight by scaling factor.

LeCun initialization: Designed for Sigmoid activation function.

Orthogonal initialization: Initialize weights by generating orthogonal matrix.

4.3 Proposed Parameter Settings

Each weight initialization method will be trained on the same CNN architecture, and the model contains three convolutional layers and two fully connected layers. The training parameter settings are as follows:

Learning rate: 0.001; batch size: 64; number of training rounds: 50; optimizer: Adam.

5 RESULT ANALYSIS METHOD

5.1 Display Method

The experimental results are expected to show the loss value and accuracy change curves of different weight initialization methods during the training process. By comparing these curves, the convergence speed and final performance of each method can be intuitively observed.

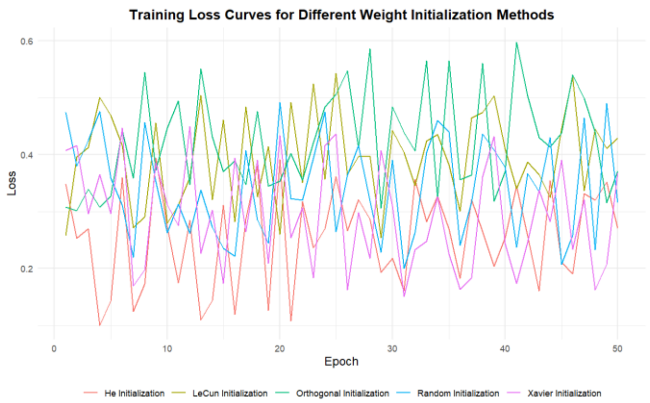

Figure 1. Training Loss Curves for Different Weight Initialization Methods

5.2 Performance Expectations

Based on the existing published research results, we believe that the He initialization method may perform best in terms of training speed and model performance, and can converge quickly and achieve high accuracy. The Xavier initialization method is second, with stable performance and good results. The random initialization method may cause large fluctuations during training due to the uneven initial weight distribution. The LeCun initialization method performs well when using the Sigmoid activation function, but the effect is average under the ReLU activation function. The Orthogonal initialization method performs well in deep networks, but the advantage is not obvious in shallow networks.

6 DISCUSSION

In this study, we compared the application effects of different weight initialization methods in convolutional neural networks. Through the analysis and derivation of existing research articles, we expect that the methods will have significant differences in training speed, stability, and final performance. Specifically, some initialization methods may show faster convergence speed in the early stage of training, while other methods may have an advantage in the final performance.

The findings of this study are of great significance to the training of deep learning models. By choosing a suitable weight initialization method, the training efficiency of the model can be significantly improved, the training time can be reduced, and the generalization ability of the model can be improved. This is particularly important for large-scale data processing and real-time systems in practical applications. In addition, this study provides a reference for subsequent research to further explore adaptive initialization methods and initialization strategies based on meta-learning.

7 CONCLUSION

This study systematically analyzes the application effects of different weight initialization methods in convolutional neural networks, focusing on the impact of these methods on training efficiency. Through the analysis and derivation of existing research articles, we expect that different initialization methods will have significant differences in training speed, stability and final performance. The analysis results show that choosing a suitable weight initialization method can significantly improve the training efficiency and performance of the model.

The main contributions of this study are: systematically comparing several common weight initialization methods and analyzing their theoretical basis and implementation methods; proposing performance expectations based on existing research articles, providing a reference for subsequent research; emphasizing the importance of weight initialization methods in deep learning model training and providing guidance for practical applications.

Although this study has made some valuable findings, there are still some limitations. Future research can verify the expected results of this study through actual experiments and explore more novel initialization strategies. In addition, combining other optimization techniques, such as learning rate scheduling and regularization methods, to further improve the training efficiency and performance of the model is also a direction worthy of in-depth research.

ACKNOWLEDGMENTS

The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

FUNDING

Not applicable.

INSTITUTIONAL REVIEW BOARD STATEMENT

Not applicable.

INFORMED CONSENT STATEMENT

Not applicable.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

PUBLISHER'S NOTE

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

AUTHOR CONTRIBUTIONS

Not applicable.

ABOUT THE AUTHORS

ZHANG, Zhenxing

Postgraduate student at Belarusian State University, Minsk, Belarus.

REFERENCES

[1] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... & Chen, T. (2018). Recent advances in convolutional neural networks. Pattern recognition, 77, 354-377.

[2] Camacho, I. C., & Wang, K. (2022). Convolutional neural network initialization approaches for image manipulation detection. Digital Signal Processing, 122, 103376.

[3] Datta, L. (2020). A survey on activation functions and their relation with xavier and he normal initialization. arXiv preprint arXiv:2004.06632.

[4] Kumar, S. K. (2017). On weight initialization in deep neural networks. arXiv preprint arXiv:1704.08863.

[5] He, J., Lan, M., Tan, C. L., Sung, S. Y., & Low, H. B. (2004, July). Initialization of cluster refinement algorithms: A review and comparative study. In 2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541) (Vol. 1, pp. 297-302). IEEE.

[6] Boulila, W., Driss, M., Alshanqiti, E., Al-Sarem, M., Saeed, F., & Krichen, M. (2022). Weight initialization techniques for deep learning algorithms in remote sensing: Recent trends and future perspectives. Advances on Smart and Soft Computing: Proceedings of ICACIn 2021, 477-484.

[7] Hu, W., Xiao, L., & Pennington, J. (2020). Provable benefit of orthogonal initialization in optimizing deep linear networks. arXiv preprint arXiv:2001.05992.

[8] Choromanski, K., Downey, C., & Boots, B. (2018, February). Initialization matters: Orthogonal predictive state recurrent neural networks. In International Conference on Learning Representations.