Activation Functions and Weight Initializers for CNNs: A comparative study

Quy–Hung Xin[1]*, Duc-Tuan Doan[1]
[1]Faculty of Information Technology, University of Science-VNUHCM, Vietnam
*Corresponding author: xqhung23@apcs.fitus.edu.vn

***Abstract*—**
***Index Terms*—Activation functions, initializer, deep neural networks.**

I. Introduction

In recent years, optimizing neural networks, particularly convolutional neural networks (CNNs), has been a major focus in the deep learning (DL) field. The primary objectives include enhancing learning capacity, improving accuracy for real-world applications, and ensuring model stability during training while maintaining computational efficiency. One strategy to achieve these objectives is improving key components of CNN models, particularly activation functions (AFs) and weight initializers (WIs).

AFs introduce non-linearity into models, enabling them to learn and represent complex patterns in data. Other optimization techniques for CNNs, such as Batch Normalization [8], Skip connections [5], and Dropout [16], while they can improve model performance, often add more additional complexity to the model. In contrast, AFs play a direct role in shaping the training process and significantly influence the learning dynamics, particularly in deep architectures, where models are more prone to vanishing and exploding gradient problems [3]. In fact, the depth of CNN architectures has increased significantly in the past decade (One of the most popular CNN models, ResNet, [5] has up to 152 hidden layers in its largest variant while DenseNet [7] extends to as many as 201 hidden layers), making it important to develop AFs that maintain stable gradient flow, accelerate convergence, and enhance model efficiency.

Traditional AFs, such as Sigmoid and Tanh, suffer from vanishing gradient problems, which negatively affect the learning process in deep networks. To overcome these limitations, ReLU [12] was introduced and became widely used due to its computational efficiency and effectiveness in deep architectures. Further advancements led to more sophisticated activation functions, including Leaky ReLU [9], PReLU [4], and Swish [13]. However, given the wide range of available AFs, selecting the most suitable one for a particular model remains a challenging and model-dependent task.

WIs, on the other hand, determine the initial values of the weights in a neural network before training begins. A poor choice of WI can lead to common issues such as vanishing or exploding gradients, slow convergence, and unstable optimization. Moreover, the combination of AFs and WIs should be carefully considered, as it significantly impacts model performance. For instance, ReLU typically performs better with He initialization than with Xavier initialization [4]. Together, AFs and WIs play a pivotal role in determining the training efficiency, stability, and overall effectiveness of CNNs.

This study aims to conduct a comparative analysis of AF and WI pairs by evaluating their practical performance. First, we provide an analysis of popular activation functions—including ReLU, LeakyReLU, PReLU, ELU[1], GELU[6], Swish, and Mish[11]—as well as initialization methods such as Xavier, He, Orthogonal [15] and LSUV[10]. Then, we will conduct experiments to assess the empirical effectiveness of various activation function and weight initializer pairings. Our experiments span three widely used datasets (MNIST, CIFAR-10, and ImageNet100) across diverse model architectures. Our evaluation will be based on the outcome accuracy and convergence speed. Our contributions are two-fold:

- We provide a comprehensive analysis of state-of-the-art AFs and WIs. As a result, we could have a deeper understanding of the techniques behind these AFs and WIs, which help bridge the gap between observed performance and the underlying mathematical foundations, ultimately leading to more reliable and efficient models.
- Our experiment can thoroughly assess the empirical effectiveness of various AFs and WIs pairings. The results would provide insight into how the interaction between activation functions and weight initializers influences training dynamics, convergence speed, and overall model performance across different network architectures and datasets.

II. Related work

Previous work mainly focused on enhancing AFs. To address the vanishing and exploding gradient problems associated with early AFs (e.g. Sigmoid and Tanh), Nair and Hinton introduced the ReLU activation function, which has since become one of the most widely used AFs [12]. To further enhance network performance, numerous ReLU variants have been proposed. For example, Leaky ReLU [9] introduces a small negative slope to mitigate the dying ReLU problem, while PReLU [4] and ELU [1] generalize traditional ReLU by incorporating learnable parameters, improving a model's ability to learn complex representations. More recently, advanced AFs, such as Swish [13], Mish [11], and GELU [6],

have been introduced, demonstrating improved gradient flow and generalization in deep networks.

Meanwhile, the impact of WIs was also explored with He et al. [4] introducing He initialization, which was shown to outperform the earlier Xavier initializer proposed by Glorot and Bengio [2] on networks using the ReLU function. In an effort to stabilize training, Mishkin and Matas introduced LSUV initialization [10], examining its interaction with different AFs. However, their study was limited to traditional AFs, such as Sigmoid, Tanh, ReLU, and PReLU, leaving the impact of modern AFs largely unexplored.

While prior research has extensively examined AFs and WIs, the interaction between them remains an area of ongoing exploration. Tomasz Szanda [17] investigated the performance of multiple AFs, but did not analyze their compatibility with different weight initialization strategies. More recently, Kit Wong, Rolf Dornberger, and Thomas Hanne [18] studied various AF-WI combinations in a simple feedforward neural network (FNN) [14]. However, the simplicity of their chosen datasets and models may not fully capture the potential of AF-WI interactions in real-world applications.

III. Methodology

IV. Results

V. Conclusion

References

[1] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[2] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[3] Boris Hanin. Which neural net architectures give rise to exploding and vanishing gradients? *Advances in neural information processing systems*, 31, 2018.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[7] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[9] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, GA, 2013.

[10] Dmytro Mishkin and Jiri Matas. All you need is a good init. *arXiv preprint arXiv:1511.06422*, 2015.

[11] Diganta Misra. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681*, 2019.

[12] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[13] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Swish: a self-gated activation function. *arXiv: Neural and Evolutionary Computing*, 2017.

[14] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[15] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[16] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[17] Tomasz Szandała. Review and comparison of commonly used activation functions for deep neural networks. *Bio-inspired neurocomputing*, pages 203–224, 2021.

[18] Kit Wong, Rolf Dornberger, and Thomas Hanne. An analysis of weight initialization methods in connection with different activation functions for feedforward neural networks. *Evolutionary Intelligence*, 17(3):2081–2089, 2024.