Writing project summary

Xin Quy Hung[1], Doan Duc Tuan[2]

[1] Email: xqhung23@apcs.fitus.edu.vn, Student ID: 23125005
[2] Email: ddtuan23@apcs.fitus.edu.vn, Student ID: 23125021

1 Introduction

Optimizing neural networks is a fundamental objective in deep learning. Among various techniques, such as enhancing a model's architecture (e.g., ResNet [1] and DenseNet [2]), selecting appropriate activation functions (AFs) and weight initializers (WIs) is crucial, as they serve as core components in every neural network. The choice of AFs and WIs can significantly impact model performance [3], and an inappropriate selection may lead to issues such as vanishing or exploding gradients [4]. Furthermore, certain activation functions exhibit greater compatibility with specific initialization methods; for example, ReLU performs more effectively with He initialization than with Xavier initialization [1].

In addition to traditional AFs such as sigmoid, tanh, and ReLU [5] and established weight initialization techniques like He initialization, numerous new approaches have been proposed to further enhance model performance. This fast-developing rate makes it challenging to determine the optimal combination for a given application. Therefore, it is important to comprehensively evaluate these methods by comparing different AFs and examining how they interact with various initialization strategies.

In this project, "Activation Functions and Weight Initializers for CNNs: A comparative analysis", we aim to investigate the performance of different AFs and WIs on both theoretical and practical sides. We will discuss each AF and WI, then conduct experiments to evaluate them based on their effectiveness across a range of datasets and models.

2 Methods

Since ReLU is preferred over Sigmoid and Tanh, we decided to investigate ReLU and popular AFs proposed after it, namely Leaky ReLU [6], PReLU [7], ELU [8], GELU [9], Swish [10], and Mish [11].

Furthermore, we will revisit traditional WIs, specifically Xavier [12] and He initialization, while also exploring more recent initializers, such as LSUV [13] and FixUp [14], to assess whether they can outperform older methods.

First, we will examine each AF and WI from a theoretical perspective. Second, since performance may vary with model complexity, we will evaluate these methods across different architectures, ranging from compact networks like LeNet to deeper models such as ResNet and DenseNet. The evaluation will be conducted on three datasets: MNIST, CIFAR-10, and ImageNet100. For each dataset, we will select models that best match the level of complexity. Our assessment will focus on common performance metrics, including loss convergence speed, overall accuracy, and training stability.

3 Conclusion

In this project, we explore how different AFs work with various WIs. We will compare these methods on models of varying complexity, from smaller networks like LeNet to deeper ones such as ResNet and DenseNet. We use three common datasets, MNIST, CIFAR-10, and ImageNet100, choosing a model that fits each dataset's level of difficulty.

As aforementioned, our evaluation focuses on key measures such as how quickly the loss decreases, the overall accuracy, and the stability of the training process. By looking at these metrics across different combinations of AFs and WIs, we can learn how impactful they are to the model.

The outcomes of our study are to help guide future deep learning projects by providing clear recommendations on which AFs and weight initialization techniques to use. This work could provide a better understanding of how these choices affect the performance of neural networks and aims to make it easier for researchers to build more effective models.

References

[1] 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.

[2] 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.

[3] Dabal Pedamonti. Comparison of non-linear activation functions for deep neural networks on mnist classification task. *arXiv preprint arXiv:1804.02763*, 2018.

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

[5] 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.

[6] 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.

[7] 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.

[8] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). arxiv 2015. *arXiv preprint arXiv:1511.07289*, 10, 2020.

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

[10] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017.

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

[12] 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.

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

[14] Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321*, 2019.