Efficient and Effective INR: A Dive into Levels-of-Experts (LoE) and Sine Activation

Md Atikuzzaman
Department of Artificial Intelligence
Kyung Hee University
Yongin-si 17104, Republic of Korea
atikuzzaman@khu.ac.kr

Sung-Ho Bae
Department of Computer Science and Engineering
Kyung Hee University
Yongin-si 17104, Republic of Korea
shbae@khu.ac.kr

***Abstract*—Implicit neural representations have emerged as a transformative approach in machine learning and computer graphics, enabling the generation and manipulation of intricate data domains including images, 3D models, and physical simulations. However, scaling INRs to large datasets or high-resolution signals poses significant challenges due to the exponential growth of network parameters with input space size. In this study, we introduce a method based on Levels-of-Experts (LoE) for implicit neural representations that proves to be more efficient and promising. We have identified that the activation associated with LoE results in sharp discontinuities, which in turn diminish the performance of the implicit neural representations. To tackle this problem, we substituted the Leaky ReLU activation function with the Sine activation function, without the need for any additional initialization schemes. This straightforward yet potent approach surpasses many previous techniques and is capable of representing large datasets or high-resolution signals effectively.**

***Index Terms*—Implicit Neural Representations, INR, Levels-of-Experts,**

I. Introduction

Implicit Neural Representations (INRs) have emerged as a groundbreaking approach for capturing latent representations of data spaces without explicitly defining the correlations between inputs and outputs. Traditional data representation methods tend to explicitly map these relationships, but INRs leverage the power of implicit functions, which are highly adaptable and capable of handling complex, high-dimensional datasets. These implicit functions eschew the need for a direct representation of the data's underlying structure, allowing for a more fluid and flexible modeling approach. The potential of INRs in representing complex data spaces is substantial. They are equipped with the intrinsic ability to comprehend and represent intricate patterns and correlations within the data, a feature that has begun to revolutionize various domains of study and application. The efficiency and adaptability of INRs are particularly notable when contrasted with more conventional data representation strategies such as point clouds or voxel grids. These traditional methods often fall short in their ability to scale or maintain fidelity when dealing with the vastness and complexity of current data requirements. INRs stand out in their ability to interpolate and extrapolate from known data points, thus offering a powerful tool for tasks that involve predictions or reconstructions from incomplete datasets. This capability makes them especially suited for applications in fields such as computer vision, where they can be used for image reconstruction, and in physics, where they assist in simulating complex systems that are challenging to model with traditional approaches. Furthermore, the implicit nature of these representations allows them to capture continuous signals and surfaces at an arbitrary resolution, which is particularly beneficial in graphics and 3D modeling. As a result, INRs have started to pave the way for innovations in rendering, animation, and the creation of digital environments, wherein detail and realism are of paramount importance. Moreover, INRs offer a unique advantage in terms of data compression. By representing data through continuous functions rather than discrete samples, they can achieve significant reductions in the storage space required, all while preserving the fidelity of the original data. This characteristic is invaluable in the era of big data, where the ability to efficiently store and process large volumes of information is critical. As the development of INRs continues, their integration into diverse applications signifies a shift towards more efficient and adaptable data processing techniques. They promise not only to enhance current methodologies but also to unlock new possibilities in the handling and understanding of complex data structures. With their capacity to model high-dimensional spaces and capture nuanced relationships within the data, INRs are poised to be a cornerstone technology in the advancement of machine learning and artificial intelligence.

Nevertheless, there are challenges when it comes to scaling INRs to accommodate large datasets or high-resolution signals. This is primarily due to the exponential rise in network parameters corresponding to the growth in input space dimensions. In light of this challenge, Hao et al. [1] presented the Levels-of-Experts (LoE) framework. This framework arranges network weights in a hierarchical manner. Within each layer, there are multiple experts, each possessing a unique weight matrix. These experts are arranged in a tiling pattern, and responsible for distinct input space regions. The LoE framework exhibits several notable advantages over traditional INRs. Firstly, it achieves improved efficiency by preventing the exponential growth of parameters with input space size. Secondly, it offers enhanced flexibility, enabling a wider range of data modeling capabilities.

However, in the Levels-of-Experts, the activation function is non-smooth, leading to the introduction of sharp discontinuities in the INR. This adversely impacts the INR's performance. To mitigate this problem, we substituted the Leaky ReLU activation function with the Sine activation function. The Sine function avoids introducing sharp discontinuities in the INR, enhancing its performance. Additionally, it can represent a diverse array of shapes and patterns. Consequently, our method exhibits enhanced accuracy, capturing complex input-output relationships more effectively than the original Levels-of-Experts (LoE) framework.

II. Related Works

Implicit Neural Representations (INRs) have recently risen to prominence as a robust mechanism for depicting and modeling intricate signals. INRs are designed to learn a continuous function that bridges input coordinates to corresponding output values. This unique capability enables efficient querying at any position within the input domain, making INRs versatile for a plethora of tasks such as signal fitting [2], novel view synthesis [3], and generative modeling [4].

Sitzmann et al. [5] suggest the use of periodic activation functions, like sine and cosine, to address the limitations associated with coordinate-based MLPs. Periodic activation functions offer numerous advantages over conventional activation functions, such as ReLU, especially in the context of INR tasks. However, the constrained encoding size curtails the overall representational capability of the model.

In another study, Sitzmann et al. [6] unveil a novel neural network architecture for neural representations, termed Adaptive Coordinate Networks (ACORNs). ACORNs are specifically crafted to surmount the inherent challenges posed by traditional coordinate networks, which include computational inefficiency and the struggle to represent intricate signals with high precision. The unique strength of ACORNs stems from their hybrid design, merging implicit and explicit network architectures.

Furthermore, Lindell et al. [7] present the Band-limited Coordinate Networks (BACON). This innovative network architecture is aimed at addressing the existing limitations of the preceding models. One of the standout features of BACON networks is their analytical Fourier spectrum. This allows for easy analysis and prediction of their behavior. Moreover, BACON networks are adept at representing signals across various scales without necessitating direct supervision.

Nevertheless, coordinate-based MLPs are not without their shortcomings. Primarily, their efficacy is bound by the model's parameter count. Furthermore, they can prove to be computationally demanding, particularly when dealing with high-resolution signals. Recognizing these constraints, recent studies have ventured into the domain of hybrid INR representations. These hybrids amalgamate coordinate-based MLPs with alternative representation modalities, such as sparse voxels [8] or Fourier harmonics [9]. While hybrid INRs have the potential to outpace coordinate-based MLPs in both capacity and efficiency, they might introduce added intricacies in terms of implementation and training.

III. Methodology

A typical coordinate-based multi-layer perceptron (MLP) is constructed as a sequence of layers, which can be mathematically represented as:

$$f : p \mapsto (g^k \circ g^{k-1} \circ \cdots \circ g^1 \circ \gamma)(p), \tag{1}$$

In this representation, p denotes the input coordinate at which the MLP is assessed. The function γ corresponds to an input mapping, similar to the sine-cosine positional encoding. The term ϕ symbolizes a non-linear activation function. Furthermore, $g^i : x \mapsto W^i x + b^i$ designates the i-th linear layer, which performs a transformation on the input using a weight matrix W^i and a bias vector b^i. As training progresses, both W^i and b^i are adjusted through gradient descent to make the MLP better fit the data. In the context of the Levels-of-Experts

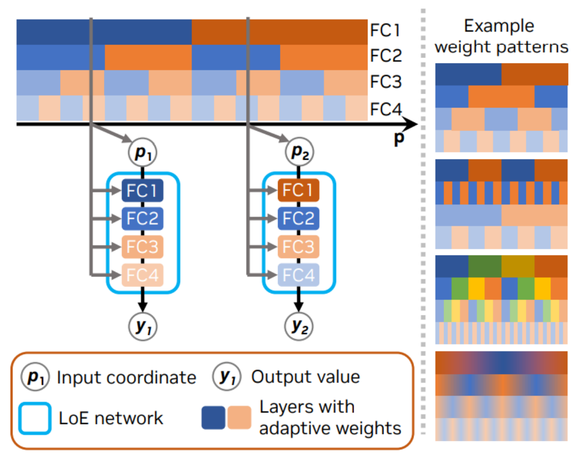

Fig. 1. Each linear layer possesses multiple sets of weights, organized in a repetitive grid pattern. Given an input, $\boldsymbol{P}$, the weight selection for each layer is contingent upon the position where $\boldsymbol{P}$ is situated on the grid. To optimize performance, varying grid scales are employed for distinct layers. Figure adapted from [1].

(LoE) model shown in Figure.1, each W^i is not just viewed as a singular matrix that can be learned. Rather, its complexity is defined as a function, $\psi^i(.)$, dependent on the input coordinate p. This concept leads to the creation of a dynamic-weight linear layer, articulated as $h^i : (x, p) \mapsto \psi^i(p)x + b^i$. Here, x refers to the layer's inputs, and p determines the position where the MLP is evaluated. By replacing the traditional linear layers g^i in the MLP with these dynamic-weight layers h^i, the MLP evolves to have input-dependent weights:

$$f : p \mapsto (h^k \circ \phi \circ h^{k-1} \circ \cdots \circ \phi \circ h^1 \circ \gamma)(p). \tag{2}$$

Given the fact that the resulting weight matrix, which is position-dependent, has dimensions much larger than its input and output vectors, and considering it will be assessed at numerous query points, it's imperative that the weight generation functions, $\psi^i(p)$, remain efficient, economical, and expressive. This need eliminates the feasibility of widely used weight-prediction networks found in hypernetwork-based methods, where predicting a high-dimensional weight for each position becomes a requirement. Instead, a simpler and more effective method based on coordinate interpolation is adopted. This involves storing multiple potential values for the weight matrix within a systematic grid (or tile) and performing cyclic interpolation contingent on the input coordinates.

Consider a situation where a grid is populated with N matrices $W_0^i, \ldots, W_{N-1}^i$. Here, i denotes the depth of the layer, and N is a non-negative integer. This model is primarily geared towards scenarios where $N > 1$, given that $N = 1$ reverts to the traditional positive integer MLP configuration. For a 1D coordinate expressed as $p = (p)$, the weight of layer i that depends on the input, W^i, is formulated as:

$$W^i = \psi^i(p) = \sum_{j=0}^{N-1} B_{j,N}(\alpha^i p + \beta^i) W_j^i, \quad (3)$$

Within this equation, α^i and β^i act as hyperparameters that adjust the grid's scale and offset for each layer, respectively. Concurrently, $B_{j,N}$ functions as the blending algorithm that calculates the blending coefficient for the j-th matrix candidate. This coefficient can be represented in multiple ways.

IV. Experimental Setups

In our experiments, we assessed the best two hierarchical arrangements of weight grids from Levels-of-Experts (LoE), including Fine to Coarse and Quad Tree configurations. We used the "camera" test image from the scikit-image Python package as a reference for our test [10]. This "camera" image is a grayscale picture with dimensions of 512 x 512 pixels.

In all of our experiments, we employed a 10-layer network with 64 hidden channels. The architecture comprises nine position-dependent linear layers, sequentially labeled from 1 to 9, and concludes with a final linear layer. Each of these position-dependent layers utilizes a 2×2 weight tile. For the Levels-of-Experts (LoE) layers, we selected the Leaky ReLU activation function with a negative slope of 0.2. For our specific configuration, we chose the sine activation function and refrained from introducing any unique initialization techniques. Notably, aspects like layers, hidden channels, and tiles remained consistent throughout all the experiments.

V. Results and Analysis

In this study, two grid patterns were analyzed using a novel Sine Activation method, which demonstrated superior performance in terms of Peak Signal-to-Noise Ratio (PSNR) values when compared to the baseline Leaky ReLU activation. The comparison is visually represented in Figure 2, where the Sine Activation method not only achieves higher PSNR values but also requires fewer training iterations to do so. The specific improvement curve for the Sine Activation is detailed in Figure 3, showcasing the efficiency of this method in reaching optimal PSNR values swiftly and the detailed specific curve for (LoE) Leaky ReLU Activation is shown in Figure 4

Further empirical evidence of the Sine Activation method's performance is found in Table I, which presents a comparative analysis of different hierarchical grid patterns. Notably, the

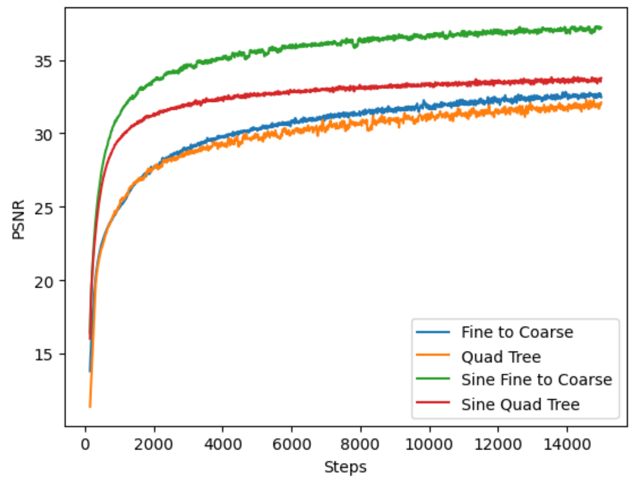

Fig. 2. PSNR vs training iterations curve for Sine Activation and (LoE) Leaky ReLU Activation

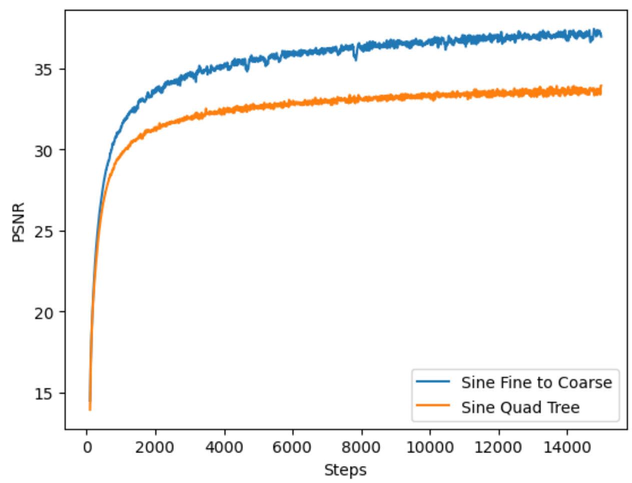

Fig. 3. PSNR vs training iterations curve for Sine Activation

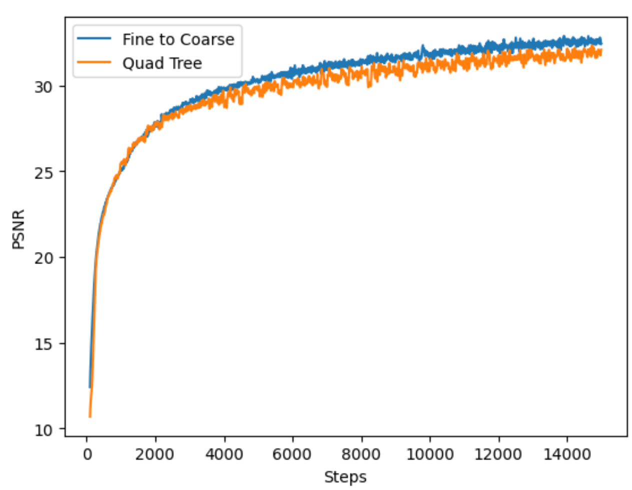

Fig. 4. PSNR vs training iterations curve for (LoE) Leaky ReLU Activation

TABLE I
COMPARISON OF DIFFERENT HIERARCHICAL ARRANGEMENTS OF GRID PATTERN IN PSNR

Grid Pattern	Peak PSNR
Fine to Coarse	34.26
Quad Tree	33.46
Sine Fine to Coarse	**39.05**
Sine Quad Tree	35.33

traditional 'Fine to Coarse' grid pattern peaked at a PSNR value of 34.26, whereas the same pattern utilizing the Sine Activation method markedly improved the peak PSNR to an impressive 39.05. Additionally, the 'Quad Tree' pattern exhibited a peak PSNR of 33.46 with the baseline activation, but with the Sine Activation, this value was enhanced to 35.33.

This significant increase in PSNR with the Sine Activation method underscores its potential for enhancing image quality and training efficiency. Such improvements suggest that the Sine Activation method can be a powerful tool in the optimization of neural networks, particularly for tasks involving image processing where PSNR is a critical measure of performance.

VI. CONCLUSION

For various hierarchical arrangements of weight grids, the activation using the Sine method often requires integration with specific initialization procedures, which demand more multiply-accumulate operations (MACs). However, in our approach, we abstained from using any specialized initialization methods. This constraint potentially limited the accuracy of our method for certain grid patterns. In future research, we aim to identify a more precise activation function that can seamlessly operate with all grid pattern types.

ACKNOWLEDGEMENTS

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Korea Government (MSIT) under Grant 2022-0-00759 and in part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-RS-2023-00258649) supervised by the IITP(Institute for Information Communications Technology Planning Evaluation)

REFERENCES

[1] Z. Hao, A. Mallya, S. Belongie, and M.-Y. Liu, "Implicit neural representations with levels-of-experts," *Advances in Neural Information Processing Systems*, vol. 35, pp. 2564–2576, 2022.

[2] V. Sitzmann, J. N. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," in *Proc. NeurIPS*, 2020.

[3] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[4] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," in *Proc. NeurIPS*, 2021.

[5] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in neural information processing systems*, vol. 33, pp. 7462–7473, 2020.

[6] J. N. Martel, D. B. Lindell, C. Z. Lin, E. R. Chan, M. Monteiro, and G. Wetzstein, "Acorn: adaptive coordinate networks for neural scene representation," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–13, 2021.

[7] D. B. Lindell, D. Van Veen, J. J. Park, and G. Wetzstein, "Bacon: Band-limited coordinate networks for multiscale scene representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 252–16 262.

[8] Z. Hao, A. Mallya, S. Belongie, and M.-Y. Liu, "Gancraft: Unsupervised 3d neural rendering of minecraft worlds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 072–14 082.

[9] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547, 2020.

[10] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "Scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014.