On the Effect of Normalization Layers in Deep Coordinate Networks

Lunjun Zhang
lunjun@cs.toronto.edu

Abstract

This paper investigates the potential of normalization layers as a solution to the low-frequency bias inherent in deep coordinate networks (DCNs). While DCNs have emerged as a powerful tool for computational imaging and graphics, their tendency towards low-frequency bias poses a significant challenge. Traditional approaches to this issue have relied on Fourier Features (FFs) or periodic activation functions like sine. We propose an alternative approach that leverages normalization layers, which are prevalent in other areas of deep learning but underexplored in the context of DCNs. By examining BatchNorm [4], LayerNorm [1], and RMSNorm [14] within DCNs, we aim to understand whether these normalization techniques can mitigate the low-frequency bias and improve the network's ability to learn high-frequency signals directly from input coordinates. Our results could pave the way for more effective and efficient DCN architectures in the future.

1 Motivation

In the realm of neural network research and application, a relatively new architecture known as deep coordinate networks (DCNs) has begun to distinguish itself [6, 9, 5, 3]. These networks, also known as *implicit representations*, are characterized by their use of low-dimensional coordinates as inputs, which stands in contrast to more traditional neural network structures. The appeal of DCNs lies in their exceptional flexibility, which is particularly advantageous for fields such as computational imaging and graphics. By using continuous coordinates of arbitrary resolutions as inputs, these networks can represent complex spatial relationships with remarkable precision, thereby unlocking new possibilities in the processing and generation of images, 3D scenes, and beyond.

Despite their potential, coordinate networks are not without their challenges. One of the most significant issues is the intrinsic low-frequency bias [7] that neural networks tend to exhibit. This bias makes it inherently difficult for DCNs to learn high-frequency signals which are often crucial for capturing detailed and nuanced aspects of images and graphics. The prevailing methods to circumvent this issue involve either the incorporation of Fourier Features [10] or the use of periodic activation functions such as sine and cosine [9]. These techniques allow for the encoding of high-frequency information in a manner that the networks can process, but they are not without their trade-offs and limitations [3, 5], which has driven the search for more effective solutions.

This project is primarily focused on exploring an alternative pathway to overcome the low-frequency bias in deep coordinate networks. Specifically, it examines whether normalization layers, which have become a cornerstone in the broader landscape of deep learning, could be repurposed to address this challenge in DCNs. Normalization layers, such as BatchNorm [4], LayerNorm [1], and RMSNorm [14], have revolutionized many aspects of neural network training, making them more stable and efficient. However, their potential impact on the performance of DCNs, particularly in learning high-frequency signals, remains largely uncharted territory. By investigating this possibility, the project seeks to offer a novel solution to one of the fundamental obstacles in the application of deep coordinate networks, potentially broadening their applicability and efficacy.

Project Report to CSC2529, Topics in Computational Imaging, 2023 Fall

2 Related Work

Deep Coordinate Networks (DCNs) have carved a niche in the landscape of neural architectures by offering a unique mapping mechanism from simple input coordinates to complex outputs, such as pixel values. This direct mapping capability is largely due to the use of Fourier Features (FFs) [10, 6], which are integral to enabling the network to interpret fine-grained spatial (or temporal) encodings. Borrowing concepts from the Transformer's positional encodings [12], FFs facilitate the DCN's understanding of coordinates by projecting them into periodic functions of varying frequencies. Furthermore, studies have demonstrated that FFs are especially crucial for multi-layer perceptrons (MLPs) that operate with ReLU activations [6], as they allow these networks to better approximate the high-frequency components of those functions. The employment of sine activation functions [9], as an alternative to FFs, has also been shown to yield promising results, allowing for the encoding and processing of high-frequency information within the network structure. In addition, the domain of DCNs has been advanced by the introduction of Multiplicative Filter Networks (MFNs) [3], which provide more controllability of the representation [5, 8]. This innovation has allowed for a linear function approximation over an exponential number of basis functions, demonstrating that MFNs can match or even outperform existing approaches that utilize Fourier features with ReLU networks or sinusoidal activation networks in certain domains. Despite the advancements brought forth by MFNs, the study of MLP-based DCNs remains critical, as they offer a versatile framework that can be easily integrated with existing deep learning architectures.

Deep Normalization Layers have long been a staple in the toolkit of deep learning, aimed at enhancing training stability and convergence rates. Within DCNs, however, the exploration of normalization layers has been limited. Techniques such as Batch Normalization [4], Layer Normalization [1], and Root-Mean-Square Normalization [14] have revolutionized training efficiency and model performance across various neural network models. Although certain types of normalization, such as Group Normalization [13] and Instance Normalization [11], might not be applicable for coordinate networks, many other types of normalization layers can be directly applied within the realm of DCNs. The potential of these normalization layers to mitigate the challenges of low-frequency bias in DCNs represents an intriguing avenue for research. By integrating these normalization strategies, there is potential not only to enhance the DCN's ability to learn high-frequency features but also to possibly speed up the overall learning process, making these networks more adept at handling complex, high-dimensional signals.

Effect of Batch Normalization on learning dynamics within deep neural networks is a complex phenomenon that has sparked extensive discussion among researchers. BatchNorm is known to accelerate neural network training and enhance generalization performance [4]. Some theory suggests that BatchNorm enhances the smaller eigenvalues to boost training speed at lower learning rates, and suppreses larger eigenvalues to maintain stability at higher learning rates [2]. In our work, we also find that Batch Normalization is very effective for DCNs. This underscores the potential of BatchNorm in not just conventional neural network architectures but also in the specialized domain of DCNs, which are increasingly important in various domains.

3 Method

3.1 Coordinate Networks

Coordinate Networks are a class of neural networks that map low-dimensional coordinates to a high-dimensional space, typically used for tasks such as image synthesis or 3D shape representation. A common approach is to use a multi-layer perceptron (MLP) with L layers and ReLU activations. Given a coordinate (x, y), the MLP processes this input through a series of transformations to predict properties like color or density at that coordinate.

The mathematical formulation of this process can be described as follows:

$$\mathbf{h}_0 = (x, y) \tag{1}$$

$$\mathbf{h}_{l+1} = \text{ReLU}(\mathbf{W}_l \mathbf{h}_l + \mathbf{b}_l), \quad l = 0, \ldots, L-1 \tag{2}$$

$$\mathbf{c} = \mathbf{W}_L \mathbf{h}_L + \mathbf{b}_L \tag{3}$$

Here, $\mathbf{h}_l$ denotes the hidden representation after the l-th layer, $\mathbf{W}_l$ and $\mathbf{b}_l$ represent the weight matrix and bias vector for the l-th layer, respectively. The function ReLU is the rectified linear unit activation function, which introduces non-linearity into the model, enabling it to learn complex functions.

3.2 Adding Normalization Layers to Coordinate Networks

We now investigate an MLP with normalization layers (MLP+Norm) for coordinate networks. The architecture uses a normalization layer right after each linear layer, and before each ReLU activation function, as shown in Figure 1. The output is a low-dimensional target such as the pixel value at the input coordinate location.

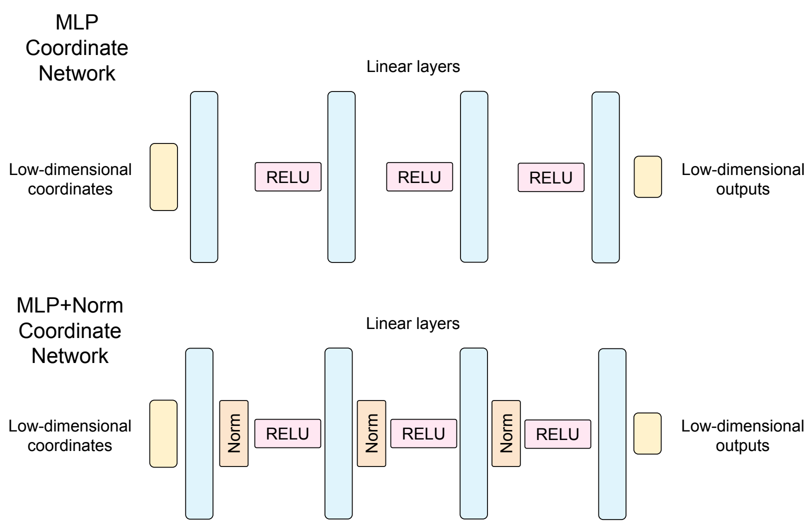

Figure 1: The architecture of our proposed MLP+Norm Coordinate Network.

The normalization techniques used are as follows:

- **Layer Normalization (LN)** [1]:
$$LN(\mathbf{x}_i) = \gamma \frac{\mathbf{x}_i - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{4}$$
where $\mu = \frac{1}{H} \sum_{j=1}^{H} \mathbf{x}_{ij}$ is the mean and $\sigma^2 = \frac{1}{H} \sum_{j=1}^{H} (\mathbf{x}_{ij} - \mu)^2$ is the variance computed across the features for a feature vector of size H.
- **RMS Normalization (RMSNorm)** [14]:
$$RMSNorm(\mathbf{x}_i) = \gamma \frac{\mathbf{x}_i}{\sqrt{\frac{1}{D} \sum_{k=1}^{D} \mathbf{x}_{ik}^2 + \epsilon}} \tag{5}$$
where D is the dimensionality of the input feature vector and the denominator is the root mean square of the feature vector components.
- **Batch Normalization (BN)** [4]:
$$BN(\mathbf{x}_i) = \gamma \frac{\mathbf{x}_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta \tag{6}$$
where $\mu_B = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i$ is the batch mean and $\sigma_B^2 = \frac{1}{m} \sum_{i=1}^{m} (\mathbf{x}_i - \mu_B)^2$ is the batch variance, computed across the m samples in the batch.

Note: In the above equations, γ and β are parameters learned during training, and ϵ is a small constant added for numerical stability.

3.3 Other Baselines

SIREN [9], or Sinusoidal Representation Networks, is a type of MLP coordinate network that utilizes sine functions as activation functions. Unlike traditional MLPs that often use ReLU activations, SIREN replaces these with the sine function to better capture the periodicities inherent in the data. The definition in a layer-wise manner is given by:

$$\mathbf{h}_{l+1} = \sin(\mathbf{W}_l \mathbf{h}_l + \mathbf{b}_l), \quad l = 0, \ldots, L-1 \tag{7}$$

where $\mathbf{h}_l$ is the output of the l-th layer, and $\mathbf{W}_l$ and $\mathbf{b}_l$ are the weights and biases, respectively.

Fourier Features [10], also known as positional encodings, are a technique used to encode the position information into a format that is more amenable to processing by MLPs. They involve transforming the input coordinates through a set of sine and cosine functions of different frequencies, effectively lifting the input into a higher-dimensional space where correlations can be more easily learned. Mathematically, this can be expressed as:

$$\mathbf{h}_0(\mathbf{x}) = [\sin(2\pi \mathbf{B}\mathbf{x}), \cos(2\pi \mathbf{B}\mathbf{x})] \tag{8}$$

where $\mathbf{x}$ is the input coordinate vector, $\mathbf{B}$ is a matrix with frequencies used for encoding, and $\mathbf{h}_0$ is the positional encoding feature vector (initial input to the network).

4 Experimental Results

In our comprehensive experimental framework, each model underwent a training regimen of 2000 gradient steps, employing the Adam Optimizer with a learning rate set to 1×10^{-4}. The architecture of each multi-layer perceptron (MLP) consisted of 3 layers, with each layer featuring a hidden dimensionality of 256. This consistent configuration across experiments ensured that the observed performance differences could be attributed to the normalization techniques rather than variations in network capacity. The robustness of different normalization techniques was rigorously evaluated within this controlled setting, providing clear insights into their respective impacts on the learning dynamics of deep coordinate networks.

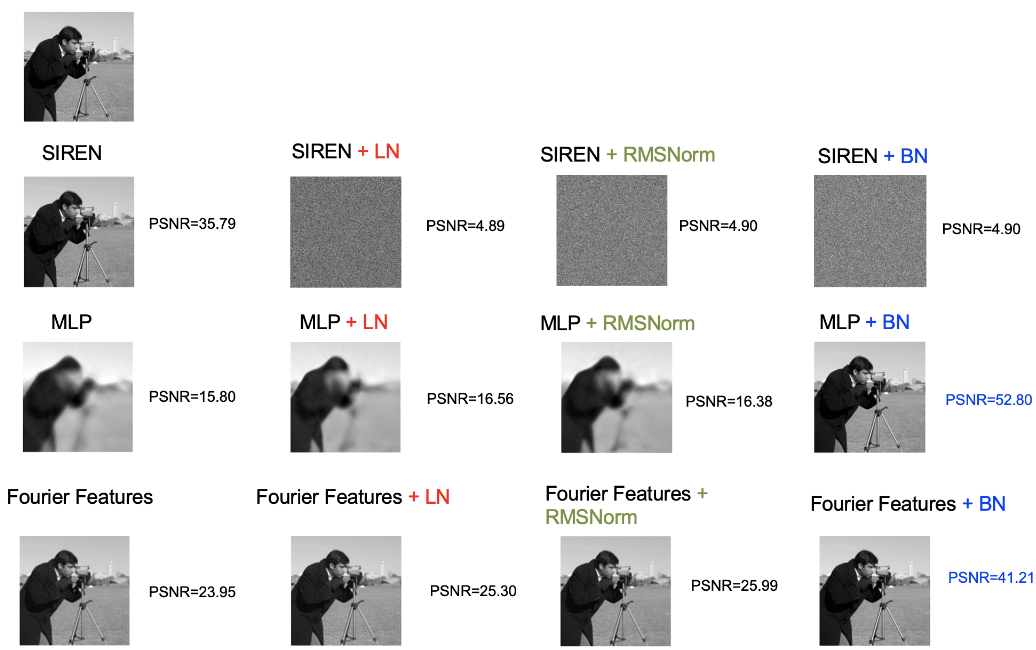

Figure 2: 2D Imaging Fitting Experiment.

The performance was measured using Peak Signal-to-Noise Ratio (PSNR). Peak Signal-to-Noise Ratio (PSNR) is a common measure used to assess the quality of reconstruction of lossy compression codecs. The metric is most commonly used in image processing to compare the output of an algorithm against a ground truth image. PSNR is defined as:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{\text{MAX}_I^2}{\text{MSE}}\right) \tag{9}$$

where MAX_I is the maximum possible pixel value of the image, and MSE is the mean squared error between the original and reconstructed image. The MSE is calculated as:

$$\text{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{10}$$

Here, m and n are the dimensions of the images, I is the original image, and K is the reconstructed image. Higher PSNR values typically indicate better quality reconstruction.

Each model was tasked with fitting a 2D image, which is a common benchmark for assessing the ability of a network to capture and reproduce complex patterns, particularly high-frequency details. We systematically applied Batch Normalization (BN), Layer Normalization (LN), and RMS Normalization (RMSNorm) across different network architectures, including a basic multi-layer perceptron (MLP), SIREN with sine activations, and networks utilizing Fourier Features for positional encoding. Special attention was given to the comparison of the same normalization technique's performance across the different setups, as our primary hypothesis centered on its potential to enhance learning of high-frequency signals.

The results were strikingly clear, with BN combined with the MLP framework yielding the highest PSNR values, significantly outperforming other normalization techniques. This finding underscores the importance of BN in enabling deep coordinate networks to effectively learn and replicate high-frequency details in images. The efficacy of BN in facilitating the neural network's ability to discern and model intricate patterns was evident when compared to the SIREN model, which, despite its sine activation functions that are theoretically beneficial for such tasks, did not achieve comparable PSNR values without BN. Similarly, models leveraging Fourier Features for positional encoding displayed improved performance with BN, suggesting that BN's role in stabilizing and speeding up the learning process is crucial across different network designs. Our experiments not only affirm the critical role of BN in enhancing MLPs but also pave the way for future research into the optimization of deep learning models for complex signal processing tasks.

Method	**PSNR**
Ground Truth	-
SIREN	35.79
SIREN + LN	4.89
SIREN + BN	4.90
MLP	15.80
MLP + LN	16.56
MLP + BN	**52.80**
MLP + RMSNorm	16.38
Fourier Features	23.95
Fourier Features + LN	25.30
Fourier Features + BN	41.21
Fourier Features + RMSNorm	25.99

Table 1: PSNR comparison of different normalization techniques applied to deep coordinate networks.

The results clearly indicate that incorporating BN with an MLP results in the highest PSNR, suggesting that this configuration is the most effective at capturing high-frequency details compared to other normalization techniques.

Figure 3: Another example of the 2D Imaging Fitting Experiment.

5 Discussion

Our main finding is that Batch Normalization (BN) significantly enhances the network's ability to learn high-frequency signals. Two potential reasons why Batch Normalization (BN) is effective for deep coordinate networks in learning high-frequency signals are:

1. **Internal Covariate Shift Reduction**: BN standardizes the inputs to each layer *across the batch*, which can reduce the internal covariate shift—where the distribution of network activations changes during training. This stabilization allows higher learning rates and more aggressive training, enabling the network to adapt more quickly and effectively to the high-frequency components of the data.
2. **Smoothing the Optimization Landscape**: By normalizing the inputs to layers, BN may help smooth the optimization landscape, making it easier for the gradient descent algorithm to find lower loss regions. This smoothing effect can be particularly beneficial in deep coordinate networks, which might have complex loss surfaces due to the high-frequency nature of the signals they attempt to model.

6 Conclusion

In conclusion, our exploration into the integration of Batch Normalization (BN) within deep coordinate networks has yielded compelling evidence of its effectiveness in enhancing the learning of high-frequency signals. This improvement is crucial for tasks that demand a high level of detail and precision. The findings from our experiments suggest that BN not only accelerates the training process but also facilitates a more robust convergence, potentially transforming the approach to designing and training deep coordinate networks for complex, high-dimensional signals.

We plan to further extend this work in two ways:

1. **Extending the experiments to NeRF models**. Due to the limitation of time, we have not yet been able to apply this technique to NeRF models. However, if the conclusion drawn from 2D image fitting experiments holds true in general, then we expect this technique to speed up NeRF training.
2. **Theoretical explanation of why this technique works so well**. We have not yet been able to fully mathematically explain why this technique works so well for coordinate works. Perhaps tools from Fourier analysis and further visualizations of the learned scales and biases inside each BN layer can offer more insights.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Elaina Chai, Mert Pilanci, and Boris Murmann. Separating the effects of batch normalization on cnn training speed and stability using classical adaptive filter theory. In *2020 54th Asilomar Conference on Signals, Systems, and Computers*, pages 1214–1221. IEEE, 2020.

[3] Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. Multiplicative filter networks. In *International Conference on Learning Representations*, 2020.

[4] 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.

[5] David B Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited coordinate networks for multiscale scene representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16252–16262, 2022.

[6] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[7] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

[8] Shayan Shekarforoush, David Lindell, David J Fleet, and Marcus A Brubaker. Residual multiplicative filter networks for multiscale reconstruction. *Advances in Neural Information Processing Systems*, 35:8550–8563, 2022.

[9] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

[10] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.

[11] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[13] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[14] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.