On the Effect of Normalization Layers in Deep Coordinate Networks

Lunjun Zhang
University of Toronto

Motivation

- In recent years, a new class of neural networks called **deep coordinate networks** has gained popularity. This type of neural nets use low-dimensional coordinates as inputs. They offer a highly flexible representation for domains such as computational imaging and graphics.
- Deep Coordinate Networks have a well-known issue: neural nets have **low-frequency bias** [1], so it is difficult for them to learn high-frequency signals directly from input coordinates. To bypass this, prior works either rely on Fourier Features [2] or periodic activation functions like sine [3].
- In this project we explore the following question: **can normalization layers provide an alternative path for addressing the low-frequency bias in coordinate networks?**

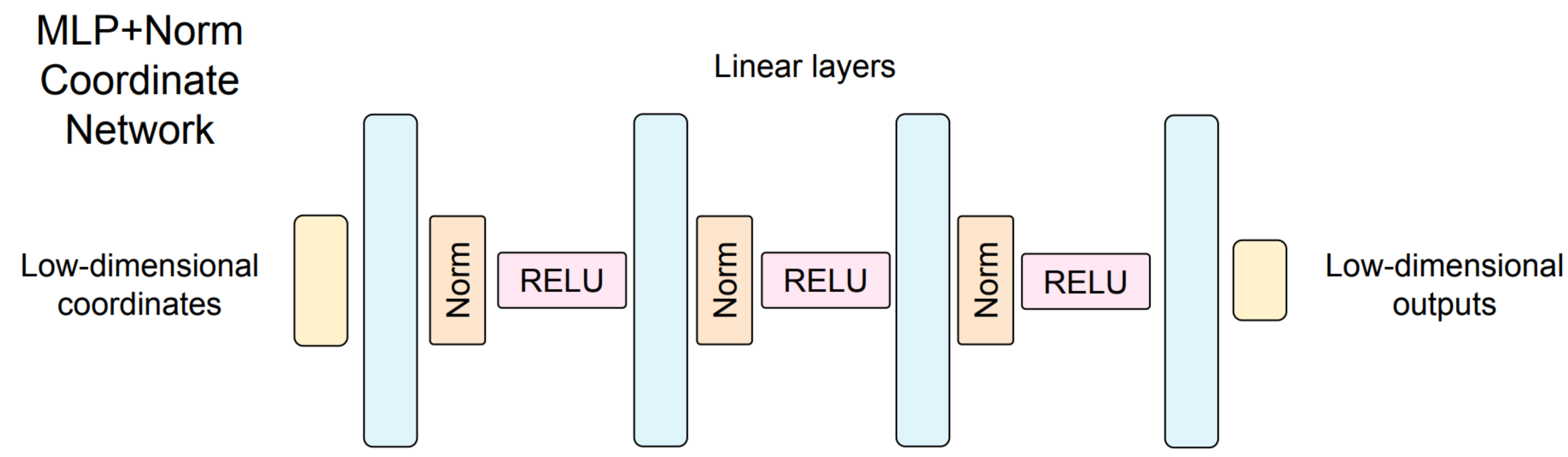
Related Work

- **Deep Coordinate Networks**: they provide a direct mapping from coordinates (x,y,z) to desired outputs such as pixel values. They tend to use Fourier Features (FFs) [2] that are loosely inspired by positional encodings in Transformer [4]; FFs have been shown to be crucial for ReLU-based MLP to function as coordinate networks. It has been shown that we can alternatively use sine activation (SIREN) [3].
- **Deep Normalization Layers**: normalization has become a standard tool in deep learning, but they have not yet been sufficiently explored in coordinate networks. Common techniques include BatchNorm (**BN**) [5], LayerNorm (**LN**) [6], and **RMSNorm** [7].

References

[1] Rahaman, Nasim, et al. "On the spectral bias of neural networks." ICML, 2019.
[2] Tancik, Matthew, et al. "Fourier features let networks learn high frequency functions in low dimensional domains." NeurIPS 2020.
[3] Sitzmann, Vincent, et al. "Implicit neural representations with periodic activation functions." NeurIPS 2020.
[4] Vaswani, Ashish, et al. "Attention is all you need." NeurIPS 2017.
[5] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." ICML, 2015.
[6] Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton. "Layer normalization." arXiv , 2016.
[7] Zhang, Biao, and Rico Sennrich. "Root mean square layer normalization." NeurIPS 2019.

Our Method



- Batch Normalization (**BN**) [5]:

$$\mathrm{BN}(x_i) = \gamma \left(\frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \right) + \beta$$

The batch here is across the input coordinates of the same sample.

- Layer Normalization (**LN**) [6]:

$$\mathrm{LN}(x_i) = \gamma \left(\frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \right) + \beta$$

- RMS Normalization (**RMSNorm**) [7]:

$$\mathrm{RMSNorm}(x_i) = \gamma \left(\frac{x_i}{\sqrt{\frac{1}{D}\sum_{k=1}^{D} x_{ik}^2 + \epsilon}} \right)$$

Experimental Results

Main Finding: BatchNorm allows RELU-based deep coordinate networks to learn high-frequency signals.