Kalafus Initialization

A Refinement of Xavier Initialization for Improved Weight Scaling in Neural Networks

Draft Version: This paper is preliminary and has not yet been peer-reviewed.

James J Kalafus

August 27, 2024

$$\sigma^2 \propto \frac{1}{\max(n_{\text{in}}, n_{\text{out}})}$$

Weight Initialization

Weight initialization is critical to the stability and convergence speed of deep neural networks. Current approaches, such as **Lecun**, **He**, and **Xavier** initialization, are designed to mitigate the vanishing and exploding gradient problems by carefully scaling weights based on the input and output dimensions of each layer. However, **Xavier initialization**—which stabilizes the forward pass and back-propagation by scaling weights according to the sum of the input and output sizes — can result in over-shrinking of weights, especially in layers with similar input and output sizes. This can slow down convergence and negatively affect learning efficiency.

To address this, I propose **Kalafus Initialization**, a novel weight initialization method that refines Xavier by scaling weights based on the **maximum** of the input or output sizes, rather than their sum. This modification retains the goal of stabilizing both the forward and backward passes but avoids the issue of excessively small weight values that can impede training. By focusing on the larger of the two dimensions, Kalafus Initialization provides a more robust foundation, ensuring that activations and gradients remain well-scaled throughout the network.

The value of Kalafus Initialization is compelling on conceptual grounds alone. Kalafus Initialization reduces the risk of over-shrinking weights. This change is simple yet impactful: by replacing the sum of input and output sizes with their maximum, **Kalafus Initialization** directly addresses potential under-scaling without overcomplicating the model's initialization process. Given the success of similar methods, Kalafus Initialization is built on a strong theoretical foundation as a conservative yet effective improvement.

- Lecun Initialization —
  - Gaussian $W \sim \mathcal{N}\left(0, \frac{1}{n_{\text{in}}}\right)$
  - Flat $W \sim U\left[-\sqrt{\frac{3}{n_{\text{in}}}}, \sqrt{\frac{3}{n_{\text{in}}}}\right]$
- He Initialization —
  - Gaussian $W \sim \mathcal{N}\left(0, \frac{2}{n_{\text{in}}}\right)$
  - Flat $W \sim U\left[-\sqrt{\frac{6}{n_{\text{in}}}}, \sqrt{\frac{6}{n_{\text{in}}}}\right]$
- Xavier Initialization —
  - Gaussian $W \sim \mathcal{N}\left(0, \frac{2}{n_{\text{in}} + n_{\text{out}}}\right)$
  - Flat $W \sim U\left[-\sqrt{\frac{6}{n_{\text{in}} + n_{\text{out}}}}, \sqrt{\frac{6}{n_{\text{in}} + n_{\text{out}}}}\right]$
- **Kalafus Initialization — e.g. ReLU activations**
  - **Gaussian** $W \sim \mathcal{N}\left(0, \frac{2}{\max(n_{\text{in}}, n_{\text{out}})}\right)$
  - **Flat** $W \sim U\left[-\sqrt{\frac{6}{\max(n_{\text{in}}, n_{\text{out}})}}, \sqrt{\frac{6}{\max(n_{\text{in}}, n_{\text{out}})}}\right]$
- **Kalafus Initialization — e.g. tanh activations**
  - **Gaussian** $W \sim \mathcal{N}\left(0, \frac{1}{\max(n_{\text{in}}, n_{\text{out}})}\right)$
  - **Flat** $W \sim U\left[-\sqrt{\frac{3}{\max(n_{\text{in}}, n_{\text{out}})}}, \sqrt{\frac{3}{\max(n_{\text{in}}, n_{\text{out}})}}\right]$

References

1. **Xavier Initialization (Glorot & Bengio)**:
   - Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics* (AISTATS), 249-256.
   - Available at: http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf
2. **He Initialization (He et al.)**:
   - He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1026-1034.
   - Available at: https://arxiv.org/abs/1502.01852
3. **LeCun Initialization (LeCun et al.)**:
   - LeCun, Y., Bottou, L., Orr, G. B., & Müller, K.-R. (1998). Efficient backprop. In G. B. Orr & K.-R. Müller (Eds.), *Neural Networks: Tricks of the Trade* (pp. 9-50). Springer.
   - Available at: https://link.springer.com/chapter/10.1007/3-540-49430-8_2