Weight Initializers in Neural Networks

This document provides an overview of common weight initialization strategies used in deep learning. Proper initialization helps mitigate vanishing/exploding gradients, accelerates training, and improves convergence stability.

Overview

Initial weights play a crucial role in how signals and gradients propagate through a network. Different initialization methods are tailored to specific activation functions and network depths.

Comparison of Initializers

Initializer Name	Description	Use Cases
Glorot Uniform	Xavier Uniform: balances forward/backward variance	tanh, sigmoid, shallow-to-mid networks
Glorot Normal	Xavier Normal: Gaussian counterpart of Glorot Uniform	tanh, sigmoid
He Uniform	He Uniform: preserves variance with ReLU activations	ReLU, Leaky ReLU, deep CNNs and MLPs
He Normal	He Normal: Gaussian counterpart for ReLU	ReLU, Leaky ReLU, deep networks
LeCun Normal	LeCun Normal: suited for SELU and self-normalizing networks	SELU activations, self-normalizing nets
Random Normal	Simple Gaussian sampling	quick experiments, baselines
Random Uniform	Simple uniform sampling	generic fallback
Zeros	All weights = 0 (breaks symmetry if used for hidden layers)	rarely used
Ones	All weights = 1	rarely used (e.g., bias)

Initialization Formulas

$$\textbf{Glorot Uniform}: \quad \left[-\sqrt{\frac{6}{n_{\text{in}} + n_{\text{out}}}},\ \sqrt{\frac{6}{n_{\text{in}} + n_{\text{out}}}}\right]$$

$$\textbf{Glorot Normal}: \quad \mathcal{N}\left(0,\ \sqrt{\frac{2}{n_{\text{in}} + n_{\text{out}}}}\right)$$

$$\textbf{He Uniform}: \quad \left[-\sqrt{\frac{6}{n_{\text{in}}}},\ \sqrt{\frac{6}{n_{\text{in}}}}\right]$$

$$\textbf{He Normal}: \quad \mathcal{N}\left(0,\ \sqrt{\frac{2}{n_{\text{in}}}}\right)$$

$$\textbf{LeCun Normal}: \quad \mathcal{N}\left(0,\ \frac{1}{n_{\text{in}}}\right)$$

- **Random Normal**: $\mathcal{N}(\mu,\ \sigma)$, defaults $\mu = 0$, $\sigma = 0.05$
- **Random Uniform**: $[\text{minval},\ \text{maxval}]$, defaults $[-0.05,\ 0.05]$
- **Zeros**: All weights = 0
- **Ones**: All weights = 1

Choosing the Right Initializer

1. **Activation-Aware**
   - **ReLU / Leaky ReLU** → He (Uniform or Normal)
   - **tanh / sigmoid** → Glorot / Xavier (Uniform or Normal)
   - **SELU** → LeCun Normal
2. **Network Depth**
   - Deep architectures often require variance-scaling (He or Glorot) to maintain stable gradients.
3. **Task Sensitivity**

- For specialized architectures (e.g., RNNs, GANs), custom initializers (e.g., orthogonal, custom scaling) can further stabilize training.