Weight Initialization and Training Stability

Mohammed Luqmaan Akhtar

December 2025

Introduction

Proper weight initialization plays a critical role in the successful training of MLPs. Poor initialization can lead to unstable gradients, slow convergence, or failure to train.

Importance of Initialization

Weights determine how signals propagate forward and gradients propagate backward. If weights are poorly initialized, gradients may vanish or explode, preventing learning.

Xavier Initialization

Xavier (Glorot) initialization aims to preserve variance across layers:

$$W \sim \mathcal{N}\left(0, \frac{1}{n}\right)$$

where n is the number of input neurons. It is suitable for sigmoid and tanh activations.

He Initialization

He initialization is designed for ReLU-based networks:

$$W \sim \mathcal{N}\left(0, \frac{2}{n}\right)$$

This compensates for the zeroing effect of ReLU activations.

Choice of Activation Functions

Activation functions introduce non-linearity but influence gradient flow:

- Sigmoid and tanh may cause vanishing gradients.
- ReLU mitigates vanishing gradients but introduces dead neurons.

ReLU and Dying ReLU

The ReLU activation is defined as:

$$g(z) = \max(0, z)$$

If a neuron consistently outputs zero, its gradient becomes zero permanently, resulting in the dying ReLU problem.

Vanishing and Exploding Gradients

During backpropagation, gradients are multiplied through layers:

$$\frac{\partial L}{\partial W^{(1)}} = \prod_{l=1}^{L} g'(z^{(l)})$$

If derivatives are less than one, gradients vanish; if greater than one, gradients explode.

Gradient Clipping

Gradient clipping limits gradient magnitude:

$$g := \frac{g}{\max\left(1, \frac{||g||}{c}\right)}$$

This stabilizes training in deep networks.

Softmax Function

The softmax function converts logits into probabilities:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

It is commonly used in the output layer for multi-class classification.