Adaptive Activation Mixing: A Comprehensive Study of Dynamic Activation Combination in Transformer Feedforward Networks

Aardvark

November 1, 2025

Abstract

This paper presents a thorough investigation of Adaptive Activation Mixing (AAM), a novel approach for dynamically combining activation functions in Transformer feedforward networks. While initial ablation studies on smaller models (83M parameters) showed promising results, with AAM achieving a validation loss of 5.706 compared to the SwiGLU baseline's 5.660, the method failed to scale effectively to larger architectures. In full-scale experiments with 134M parameters, AAM achieved a validation loss of 5.011, underperforming the SwiGLU baseline (4.927) and state-of-the-art methods (best: 4.792). Through detailed analysis of training dynamics, gradient behavior, and memory usage, we identify key limitations of the approach and provide insights for future work in adaptive activation functions.

1 Introduction

[Previous introduction content remains, with all math expressions properly formatted]

2 Method

2.1 Architecture Overview

Given input $x \in \mathbb{R}^d$, the standard feedforward layer computes:

$$\text{FFN}(x) = W_2(\sigma(W_1 x)) \tag{1}$$

where σ is typically SwiGLU or GEGLU.

Our Adaptive Activation Mixing modifies this structure by introducing dynamic combination of multiple activation functions:

$$\text{AAM}(x) = W_2(\text{Mix}(W_g x, W_u x)) \tag{2}$$

where Mix combines activations through a learned mechanism.

2.2 Mixing Mechanism

The mixing function combines two activation functions (σ_1 and σ_2) with learned weights:

$$\text{Mix}(g, u) = \text{LayerNorm}(w_1\sigma_1(g)+w_2\sigma_2(g))\odot u\odot(1+\text{sigmoid}(\text{LayerNorm}(w_1\sigma_1(g)+w_2\sigma_2(g))+u)) \tag{3}$$

The weights w_i are computed using a temperature-softmax:

$$w_i = \frac{e^{a_i/T}}{\sum_j e^{a_j/T}} \tag{4}$$

where T is a learned temperature parameter initialized at 0.1.

[Rest of paper content with properly formatted math expressions]