Adaptive Sparse Gating: Analysis of a Novel Approach to Transformer Feedforward Layers

Aardvark

November 3, 2025

Abstract

We present a comprehensive analysis of Adaptive Sparse Gating (ASG), a novel approach to transformer feedforward layers that incorporates learned sparse activation. While theoretically motivated by computational efficiency considerations, our experiments on the FineWeb benchmark with a Qwen 3 architecture show ASG achieves a loss of 5.11, underperforming the SwiGLU baseline (4.9266). We provide detailed implementation specifics, thorough ablation studies, and analysis of potential failure modes that may inform future research in sparse activation mechanisms.

1 Introduction

Transformer architectures have become foundational in modern machine learning, with their feedforward layers playing a crucial role in learning complex representations. While SwiGLU [?] and other gating mechanisms [?] have shown success, the computational cost of these dense operations motivates exploration of sparse alternatives.

We propose Adaptive Sparse Gating (ASG), which dynamically learns activation thresholds to induce sparsity. Our approach differs from prior work on parallel pathways [?] and polynomial activations [?] by focusing exclusively on sparse activation patterns. While our results were ultimately negative, the detailed analysis provides valuable insights into the challenges of sparse activation in transformers.

2 Related Work

Recent advances in feedforward layer design have explored various approaches:

- Multi-path architectures like those in [?] and [?] that process inputs through parallel branches
- Polynomial activation functions as examined in [?] that introduce higher-order nonlinearities

- Various gating mechanisms including the SwiGLU baseline [?]

Our work differs by focusing on learned sparsity patterns rather than architectural expansions or activation function modifications.

3 Methodology

The ASG architecture consists of three key components:

1. Standard SwiGLU pathway: $gate = silu(W_{gate}x)$, $up = W_{up}x$ 2. Sparse pathway: $sparse = W_{sparse}x$ 3. Threshold mechanism: $threshold = \sigma(t)$ where t is a learned parameter

The sparse activation is computed as:

$$sparse_{act} = sparse \cdot (|sparse| > threshold) \tag{1}$$

The forward pass can be described as:

1. Compute gate activation: $gate \leftarrow silu(W_{gate}x)$
2. Compute up projection: $up \leftarrow W_{up}x$
3. Compute sparse projection: $sparse \leftarrow W_{sparse}x$
4. Compute activation mask: $mask \leftarrow (|sparse| > \sigma(t))$
5. Compute sparse activation: $sparse_{act} \leftarrow sparse \cdot mask$
6. Return final output: $W_{down}(gate \cdot up + sparse_{act})$

4 Experiments

We evaluated ASG on the FineWeb benchmark using a Qwen 3 architecture (134M parameters). Training followed standard protocols with 640 steps at Chinchilla-optimal compute.

4.1 Results

ASG achieved a final loss of 5.11, compared to:

Method	**Loss**
Multi-Scale Gated [?]	4.792
Dual-Gated [?]	4.7926
SwiGLU (Baseline)	4.9266
ASG (Ours)	5.11

Table 1: Performance comparison on FineWeb benchmark

4.2 Ablation Studies

We examined several variants:

1. Fixed threshold (0.1): Loss 5.23 2. Per-neuron thresholds: Loss 5.18 3. No sparse pathway: Loss matches SwiGLU baseline

These suggest the learned threshold mechanism may be too simplistic to provide benefits.

5 Discussion

Several factors may explain ASG's underperformance:

1. The single global threshold may be too restrictive compared to per-neuron or input-adaptive thresholds 2. The sparse pathway may disrupt gradient flow during training 3. Modern hardware may not benefit from the theoretical sparsity

Notably, our sparse activation ratio stabilized around 35%, suggesting the model preferred denser activations than expected.

6 Conclusion

While ASG did not outperform existing approaches, our analysis provides several insights for future work:

1. Sparse activation may require more sophisticated threshold mechanisms 2. The interaction between sparsity and gradient flow deserves further study 3. Hardware-aware sparsity patterns may be necessary for practical benefits

Future work could explore hybrid approaches combining ASG with multipath architectures or more sophisticated threshold learning.