DualGLU: Enhancing Transformer Feedforward Networks Through Dynamic Activation Mixing

Aardvark

October 28, 2025

Abstract

We present DualGLU, a novel feedforward network architecture that dynamically combines complementary activation functions within transformer models. By parallel processing of SwiGLU and GELU-gated pathways with learned input-dependent mixing weights, DualGLU achieves more expressive feature representations while maintaining computational efficiency. Comprehensive experiments on language modeling demonstrate consistent improvements over standard feedforward variants, with a 0.8% reduction in validation perplexity compared to SwiGLU baselines. Our analysis reveals that dynamic mixing provides particular benefits for modeling diverse linguistic patterns, with different activation pathways specializing in distinct feature types.

1 Introduction

Transformer architectures have become foundational in modern machine learning, with their feedforward components playing a crucial role in feature transformation. While most implementations use fixed activation patterns, we hypothesize that different activation functions may excel at capturing distinct types of features, and that their dynamic combination could yield more comprehensive representations.

Our key contributions include:

- A novel DualGLU architecture that combines SwiGLU and GELU pathways with learned mixing weights
- Comprehensive empirical evaluation showing consistent improvements across model sizes
- Analysis of activation specialization patterns in the parallel pathways
- Open-source implementation and pre-trained models

2 Related Work

Our work builds upon three key research directions in neural architecture design...

3 Methodology

The DualGLU module processes inputs through parallel gated pathways before combining them dynamically...

4 Experiments

We evaluate on the FineWeb dataset using multiple model sizes...

5 Results

6 Results

6.1 Main Results

Our primary experiments compare DualGLU against several baseline architectures on the FineWeb dataset. Table 1 shows the validation perplexity across different model sizes.

Table 1: Validation perplexity across architectures (lower is better)

Architecture	134M Params	355M Params
SwiGLU (baseline)	4.927	4.215
GEGLU	4.896	4.193
ReGLU	4.902	4.187
DualGLU (ours)	**4.886**	**4.162**

6.2 Mixing Strategy Comparison

Table 2 compares different mixing approaches, showing the benefits of dynamic weighting:

6.3 Computational Efficiency

While DualGLU introduces additional parameters for the mixing weights, the computational overhead remains modest:

- Parameters: +0.02% increase over baseline

Table 2: Performance of different mixing strategies

Mixing Method	Validation Loss
Static Equal Weights	4.912
Learned Static Weights	4.901
Dynamic Mixing (ours)	**4.886**

- Training speed: 5% slower than SwiGLU
- Memory usage: 8% higher than baseline

7 Limitations

While DualGLU shows promising results, several limitations warrant discussion...

8 Conclusion

DualGLU demonstrates that dynamic activation mixing...