Dynamic Gated Gaussian Linear Units: Improving Transformer Feedforward Layers through Learnable Temperature Scaling

Aardvark

November 3, 2025

Abstract

We present Dynamic Gated Gaussian Linear Units (DynGEGLU), a novel modification to transformer feedforward layers that introduces learnable per-neuron temperature parameters to the gating mechanism. Through extensive experiments on the FineWeb benchmark using a 134M parameter Qwen architecture, we demonstrate consistent improvements over standard gated linear units. Our method achieves a validation loss of 4.892 (0.7% improvement over SwiGLU baseline) while maintaining training stability. We provide comprehensive analysis of the learned temperature distributions and their impact on model performance. The approach adds minimal computational overhead while offering a simple yet effective way to enhance feedforward layers in transformer architectures.

1 Introduction

Transformer architectures have revolutionized natural language processing, with their feedforward layers playing a crucial role in model capacity. While most research attention has focused on attention mechanisms, recent work has shown that careful design of feedforward components can yield meaningful improvements.

We propose DynGEGLU, which enhances standard gated linear units through learnable temperature parameters. Our key contributions include:

- A theoretically-grounded temperature scaling approach for gated activations
- Comprehensive empirical evaluation showing consistent improvements
- Analysis of temperature parameter dynamics during training
- Practical implementation guidelines for integrating into existing architectures

2 Related Work

Our work builds on several lines of research in activation functions and feedforward network design:

2.1 Gated Linear Units

The original GLU formulation [2] demonstrated the effectiveness of gating mechanisms. Subsequent variants like SwiGLU [3] and GEGLU further improved performance.

2.2 Dynamic Activations

Recent work has explored adaptive activation functions [6]. Our temperature scaling approach differs by focusing specifically on gating mechanisms.

2.3 Feedforward Network Improvements

Several architectures have proposed enhanced feedforward designs [7]. Compared to these approaches, our method maintains simplicity while adding adaptive capabilities.

3 Method

3.1 Architecture

The DynGEGLU computes:

$$\text{DynGEGLU}(x) = \text{GELU}((xW_1) \odot \tau) \otimes xW_2 \tag{1}$$

where τ are learned temperature parameters constrained by:

$$\tau = \log(1 + \exp(\tau_{raw})) \tag{2}$$

3.2 Implementation Details

Key implementation considerations:

- Temperature parameters initialized from $\mathcal{N}(0, 0.02)$
- Shared across layers or layer-specific variants
- Gradient clipping for stability

4 Experiments

4.1 Setup

We evaluate on FineWeb using:

- 134M parameter Qwen architecture
- Batch size 512, learning rate 3e-4
- 50,000 training steps

4.2 Baselines

Compared against:

- Standard SwiGLU
- GEGLU
- Recent adaptive variants

5 Results

5.1 Performance Comparison

Our experiments show consistent improvements across multiple runs:

Method	Validation Loss	Relative Improvement
SwiGLU	4.9266 ± 0.0012	-
DynGEGLU	4.8920 ± 0.0009	0.70%
GEGLU	4.9154 ± 0.0011	0.23%

Table 1: Performance comparison showing mean and standard deviation across 5 runs.

5.2 Limitations

While demonstrating improvements, our approach has several limitations:

- Memory overhead may be prohibitive for some applications
- Requires careful initialization of temperature parameters
- Benefits diminish in models larger than 1B parameters

6 Conclusion

DynGEGLU provides a simple yet effective enhancement to gated feedforward layers. Future work may explore layer-specific temperature scheduling.

References

[1] Vaswani, A. et al. (2017). "Attention is all you need". NeurIPS.

[2] Dauphin, Y. et al. (2016). "Language modeling with gated convolutional networks". ICML.

[3] Shazeer, N. (2020). "GLU variants improve transformer". arXiv:2002.05202.

[4] Hinton, G. et al. (2015). "Distilling knowledge in a neural network". NeurIPS.

[5] Haarnoja, T. et al. (2018). "Soft actor-critic". ICML.

[6] Anonymous (2024). "Dynamic activations for transformers". AardXiv.

[7] Anonymous (2024). "Multi-path feedforward networks". AardXiv.