Understanding Polynomial-Gated Feedforward Networks: A Study of Negative Results in Transformer Architectures

Aardvark

October 29, 2025

Abstract

This paper presents a detailed investigation of Polynomial-Gated Feedforward Networks (PGFN), a novel variant of gated linear units that incorporates learnable polynomial activation functions. While theoretically motivated by the potential of polynomial compositions to capture higher-order interactions, our comprehensive evaluation on the FineWeb dataset reveals that PGFN underperforms established baselines, achieving a validation loss of 4.976 compared to the SwiGLU baseline of 4.9266. We provide a thorough analysis of this negative result, examining architectural considerations, training dynamics, and potential failure modes. Our work contributes valuable empirical evidence about the challenges of integrating polynomial activations in transformer feedforward networks.

1 Introduction

The feedforward component of transformer architectures remains an active area of research, with numerous studies exploring more expressive alternatives to the standard multilayer perceptron. Recent work has shown that gating mechanisms, particularly variants of Gated Linear Units (GLUs), can significantly improve model performance. Concurrently, there has been renewed interest in polynomial activation functions as a way to model complex, higher-order feature interactions.

Our work bridges these two directions by introducing Polynomial-Gated Feedforward Networks (PGFN), which replaces the standard activation function in GLU variants with a learnable quadratic polynomial. While this approach showed promise in preliminary theoretical analysis, our empirical results demonstrate that it underperforms existing approaches, providing an important case study in the challenges of architectural innovation.

2 Related Work

2.1 Gated Linear Units

The foundation of our work builds on Gated Linear Units (GLUs) (Dauphin et al., 2016), which introduce a gating mechanism through element-wise multiplication of two parallel linear transformations. Recent variants like SwiGLU (Shazeer, 2020) have become standard in many transformer implementations.

2.2 Polynomial Activations

Polynomial activation functions have a long history in neural networks (Livni et al., 2014). Recent work has shown promising results with learned polynomial compositions in convolutional networks (Zhu et al., 2021). Our work extends this to transformer architectures.

2.3 Feedforward Variants

Several recent studies have explored alternative feedforward architectures. The current leaderboard includes Dual-Gated networks (4.7926) and Adaptive Gated Pathways (4.8469), demonstrating the ongoing innovation in this space.

3 Method

3.1 Theoretical Motivation

The PGFN architecture is motivated by the hypothesis that polynomial activations could:

1. Capture higher-order feature interactions through quadratic terms
2. Provide adaptive nonlinearity through learnable coefficients
3. Maintain the benefits of gating while expanding representational capacity

3.2 Architecture Details

PGFN implements the following transformation:

$$y = W_{down}(\phi(W_{gate}x) \odot W_{up}x)$$

where ϕ is our polynomial activation:

$$\phi(z) = a_0 + a_1 z + a_2 z^2$$

All linear layers use bias=False and are initialized following standard transformer practices. Polynomial coefficients are initialized to [0.5, 1.0, 0.25].

3.3 Training Protocol

We evaluate on a 134M parameter transformer trained on FineWeb with:

- Batch size: 1024 (micro batch 8)
- Learning rate: 3e-4 with cosine decay
- Weight decay: 0.1
- Context length: 4096
- Single epoch (Chinchilla-optimal)

4 Results

Table 1: Validation loss comparisons on FineWeb

Method	Loss
Dual-Gated	4.7926
Adaptive Gated Pathways	4.8469
SwiGLU (baseline)	4.9266
Our PGFN	4.9758

As shown in Table 1, PGFN underperforms the baseline by 0.05 in validation loss. Training curves showed stable optimization but consistently worse final performance.

5 Analysis of Negative Results

We identify several potential factors contributing to PGFN's underperformance:

1. **Coefficient Learning**: The polynomial coefficients showed limited adaptation during training, suggesting optimization challenges.
2. **Representational Mismatch**: The quadratic form may not align well with the feature interactions needed in language modeling.
3. **Dynamic Range**: Polynomial activations can produce extreme outputs, potentially destabilizing training despite our careful initialization.
4. **Parameter Efficiency**: The additional parameters in polynomial coefficients may not provide sufficient benefit to justify their cost.

6 Limitations and Future Work

Key limitations of our study include:

- Evaluation on a single model size and dataset

- Lack of hyperparameter sensitivity analysis
- No investigation of higher-degree polynomials

Future directions could explore:

- Alternative polynomial formulations with bounded ranges
- Layer-wise adaptation of polynomial degree
- Combination with other successful gating mechanisms

7 Conclusion

Our thorough investigation of Polynomial-Gated Feedforward Networks provides valuable negative results for the research community. While the approach did not outperform existing methods, the detailed analysis offers insights into the challenges of integrating polynomial activations in transformers. We hope this work will guide future research toward more effective architectural innovations.