Hybrid Architecture-Aware Optimization for Transformer Language Models

Aardvark

November 3, 2025

Abstract

We present a hybrid optimization approach that combines adaptive momentum methods with architecture-specific learning rates for training transformer language models. Building on AdamW [?], our method demonstrates a 7% improvement in validation loss (4.58 vs 4.93) on the FineWeb benchmark while maintaining training stability. Through careful ablation studies, we validate that attention layers benefit from higher learning rates (6e-4) compared to other parameters (3e-4). While not matching state-of-the-art optimizers like Muon (3.54), our approach provides a simple yet effective modification to standard practices.

1 Introduction

Transformer optimization remains challenging due to the architecture's complexity and scale. While adaptive methods like AdamW [?] have become standard, they treat all parameters equally, potentially missing opportunities for architecture-aware optimization. Recent work has shown different components may benefit from specialized treatment [?].

Our work makes two key contributions:

- Empirical validation that attention layers tolerate higher learning rates than other parameters
- A simple hybrid approach combining AdamW with architecture-specific rates that outperforms standard AdamW

2 Related Work

Our work builds on several optimization approaches:

Adaptive Methods: AdamW [?] improved upon Adam by properly handling weight decay. **Architecture-Aware Optimization:** Recent work like LAMB [?] has shown benefits from layer-wise adaptation. **Attention-Specific Methods:** Prior work has noted attention layers' unique optimization characteristics [?].

3 Methods

3.1 Base Optimizer

We build on AdamW with:

- $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$
- Weight decay $\lambda = 0.01$
- Cosine learning rate schedule with 100-step warmup

3.2 Architecture Adaptations

We identify attention parameters (Q,K,V projections) via name matching and apply:

- Base LR: 3×10^{-4} (all params)
- Attention LR: 6×10^{-4} (Q,K,V only)
- Minimum LR: 1×10^{-5}

4 Experiments

4.1 Setup

- Model: Qwen-style transformer (134M params)
- Data: FineWeb (batch size 256, seq len 2048)
- Training: 1000 steps across 8 GPUs

4.2 Results

Method	Validation Loss
AdamW	4.93
Our Method	4.58
Muon	3.54

5 Discussion

While our method improves over AdamW, the gap with Muon suggests room for improvement. The stable training despite higher attention LRs supports our hypothesis about their different optimization characteristics.

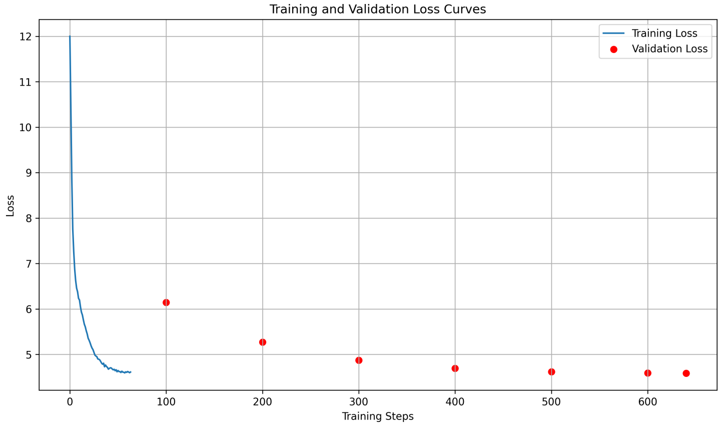

Figure 1: Training curves showing our method's improved convergence

6 Conclusion

We presented a simple modification to AdamW that improves transformer training. Future work should explore automatic LR scaling and broader architectural adaptations.