CS336: Language Modeling from Scratch

Lecture 11: Scaling Laws - Part 2

Stanford University - Spring 2025

Abstract

Abstract: This lecture provides detailed case studies of how modern language model builders apply scaling laws in practice, examining real implementations from Cerebras-GPT, MiniCPM, and DeepSeek. We explore advanced techniques including the Maximal Update Parameterization (μP), Warm-up Stable Decay (WSD) learning rate schedules, and practical strategies for hyperparameter stability across scales. The second half provides a rigorous mathematical derivation of μP, demonstrating how initialization and learning rate scaling can achieve scale-invariant training dynamics.

Contents

1 Introduction and Motivation

1.1 The Practical Scaling Challenge

After the theoretical foundations from Lecture 9, we now examine how scaling laws are actually implemented in production language model training. Post-ChatGPT, most frontier labs became secretive about their scaling methodologies, making open research from specific organizations particularly valuable.

Core Questions for Practitioners:

- Does Chinchilla's approach actually work in practice?
- Can scaling laws reliably set optimal learning rates?
- How do we achieve hyperparameter stability across scales?
- What architectural choices scale predictably?

1.2 Scale-Invariant Training Objectives

Case Study Insight

The ideal scaling scenario: hyperparameters and design choices remain stable across all model scales, eliminating the need for expensive hyperparameter tuning at large scales.

Standard Problem: As models grow wider, optimal learning rates typically decrease, requiring expensive hyperparameter searches at each scale.

μP Solution: Reparameterize models so optimal learning rates remain constant across scales.

2 Case Study 1: Cerebras-GPT

2.1 Overview and Approach

Model Family: 0.1B to 13B parameters, trained with Chinchilla-optimal token ratios

Key Innovation: First major public validation of Maximal Update Parameterization (μP)

Core Finding: μP enables more stable and predictable scaling compared to standard parameterization

2.2 Empirical Results

Performance Comparison:

- Standard parameterization: Large oscillations around predicted scaling curves
- μP: Much closer adherence to scaling law predictions
- Competitive with Pythia and GPT-J baselines

2.3 Implementation Details

Methodology

Cerebras μP Implementation:

- **Initialization:** All non-embedding parameters scaled by 1/width
- **Learning Rates:** Per-layer learning rates scaled by 1/width
- **Key Difference:** Layer-specific learning rates (not global)

2.4 Aggressive Proxy Scaling

Strategy: Scale down experiments to 40M parameters for extensive hyperparameter search, then scale up using μP stability.

Process:

1. Train proxy models across hyperparameter grid
2. Identify optimal configurations at small scale
3. Apply μP to maintain stability during scale-up
4. Validate predictions at target scale

Risk: Unclear whether 40M $\rightarrow$ 13B scaling is too aggressive for real production systems.

3 Case Study 2: MiniCPM

3.1 Objectives and Strategy

Goal: Train exceptionally high-quality small models (1.2-2.4B parameters) using extensive compute

Performance Achievement: Matched 7B model quality with 2.4B parameters (2024 standards)

Scaling Strategy: Use μP for hyperparameter stability, focus on data scaling rather than model scaling

3.2 μP Implementation

Parameterization Scheme:

- Embeddings: Constant scaling
- Residual connections: Scale by $\sqrt{\text{layers}}$
- Weights: Initialize with 1/base_width
- Learning rates: Scale by model width

Similarity to Cerebras: Nearly identical scaling factors, suggesting convergence on optimal μP practices.

3.3 Warm-up Stable Decay (WSD) Learning Rates

Methodology

WSD Innovation: Enable Chinchilla analysis in approximately one training run by using reusable learning rate phases.

Standard Cosine Problem: Different target training lengths require different cosine schedules, necessitating separate training runs for each data point.

WSD Solution: Three-phase schedule:

1. **Warmup:** Standard linear increase to peak learning rate
2. **Stable:** Flat learning rate for majority of training
3. **Decay:** Rapid cooldown to minimum learning rate

Key Advantage: Can rewind to any point in stable phase and apply decay, simulating different training lengths without full retraining.

3.4 Critical Batch Size Analysis

Methodology: Replicate Kaplan's critical batch size analysis using μP-stabilized models.

Key Findings:

- Critical batch size scales predictably with target loss
- Lower target losses enable larger batch sizes
- Maintains log-linear relationship across scales

Learning Rate Stability Validation: Optimal learning rate remains constant at 10^{-2} across multiple orders of magnitude in model size.

3.5 Extended Chinchilla Analysis

Novel Finding: 192 tokens per parameter ratio (significantly higher than standard 20:1)

Justification Arguments:

- Improved data quality vs. original Chinchilla datasets
- Enhanced model efficiency from architectural improvements
- Better optimization techniques

Validation: Modern models (Llama 3) do train with higher ratios without severe diminishing returns, supporting the feasibility of exceeding Chinchilla ratios.

4 Case Study 3: DeepSeek LLM

4.1 Approach and Philosophy

Model Sizes: 7B and 67B parameters, competitive with Llama 2 and Mistral at release

Methodological Difference: Direct scaling law fitting for hyperparameters instead of μP

Core Philosophy: Strong belief in scaling law extrapolation combined with extensive empirical validation

4.2 Direct Hyperparameter Scaling

Batch Size Scaling:

1. Train models at two different scales with hyperparameter grids
2. Identify optimal batch sizes at each scale
3. Fit scaling law to optimal points
4. Extrapolate to target model size

Learning Rate Scaling:

- Similar grid search approach
- Fit scaling law to optimal learning rates
- Extrapolate to large-scale training

Scaling Law Quality: Batch size relationships appear more robust than learning rate relationships.

4.3 WSD Implementation

DeepSeek Variant: Warmup → Stable → Two-phase decay (10

Performance: Matches cosine learning rate performance while enabling efficient Chinchilla analysis.

4.4 Chinchilla Replication

High-Quality Results: Clean isoFLOP analysis with well-fitted quadratics and clear optimal points.

Case Study Insight

Chinchilla-style isoFLOP analysis consistently produces clean, reliable results across different research groups, while hyperparameter scaling laws often appear noisier and less reliable.

Validation Strategy: Rather than copying existing ratios, DeepSeek performed complete scaling analysis to derive their own optimal token-to-parameter ratios.

Predictive Success: Successfully extrapolated from 10^{20} to 10^{24} FLOPs, accurately predicting 7B and 67B model performance.

5 Modern Scaling Developments (2024-2025)

5.1 Llama 3 Scaling Results

Updated Chinchilla Ratio: Approximately 39:1 tokens per parameter (vs. original 20:1)

Trend Analysis: Consistent upward trend in optimal ratios across multiple research groups, suggesting:

- Improved architectural efficiency

- Higher quality training data
- Better optimization algorithms

Downstream Performance Correlation: Attempts to correlate log-likelihood improvements with downstream task accuracy using sigmoid fits.

5.2 Hunyuan-Large (MoE Scaling)

MoE-Specific Analysis: 96:1 data-to-active-parameter ratio for mixture of experts models

Architectural Considerations: Different optimal ratios expected due to sparse activation patterns in MoE architectures.

5.3 MiniMax-01 (Linear Attention Validation)

Architectural Validation: Use scaling laws to justify linear attention choices over standard softmax attention

Method: Lower envelope analysis (Chinchilla Method 1) comparing:

- Softmax attention (quadratic)
- Lightning attention (linear)
- Hybrid models

Result: Linear and hybrid models scale equivalently to softmax attention, justifying efficiency gains.

6 Common Scaling Patterns

6.1 Methodological Convergence

Hyperparameter Stability:

- Cerebras + MiniCPM: Use μP for stability
- DeepSeek: Direct scaling law extrapolation
- Both approaches achieve similar goals

Chinchilla Replication:

- Nearly universal adoption of isoFLOP analysis
- WSD learning rates increasingly common
- Consistent methodology across research groups

Proxy Model Usage: Aggressive scaling down for hyperparameter search, then stable scaling up.

6.2 Reliability Hierarchy

Most Reliable: Chinchilla isoFLOP analysis - consistently clean results **Moderately Reliable:** Batch size scaling laws - generally predictable **Least Reliable:** Learning rate scaling laws - often noisy, require careful validation

7 Maximal Update Parameterization (μP) Theory

7.1 Theoretical Foundations

Theoretical Foundation

Two Core Principles for Scale-Invariant Training:

1. **Activation Stability:** Activations at initialization remain $\Theta(1)$ as width increases
2. **Update Stability:** Activation changes after one gradient step remain $\Theta(1)$

Mathematical Framework: Consider deep linear networks to derive scaling requirements, then extend to nonlinear cases.

7.2 Activation Stability Derivation

Setup: Deep linear network with layers $h^{(l)} = W^{(l)}h^{(l-1)}$

Initialization: Gaussian weights $W^{(l)} \sim \mathcal{N}(0, \sigma_l^2 I)$

Matrix Concentration: For large width, operator norm concentrates:

$$\|W^{(l)}\|_{\text{op}} \approx \sigma_l \sqrt{n_l + n_{l-1}}$$

Stability Condition: For $\|h^{(l)}\| = \sqrt{n_l}$ (desired scaling), we need:

$$\sigma_l = \frac{1}{\sqrt{n_{l-1}}} \min\left(1, \sqrt{\frac{n_l}{n_{l-1}}}\right)$$

Simplified Form: $\sigma_l = \frac{1}{\sqrt{\text{fanin}}}$ plus aspect ratio correction.

Inductive Proof: If $\|h^{(l-1)}\| = \sqrt{n_{l-1}}$, then:

$$\|h^{(l)}\| = \|W^{(l)}\|_{\text{op}} \|h^{(l-1)}\| \approx \sqrt{n_l}$$

7.3 Update Stability Derivation

Gradient Update: $\Delta W^{(l)} = -\eta \frac{\partial L}{\partial W^{(l)}} = -\eta \frac{\partial L}{\partial h^{(l)}} (h^{(l-1)})^T$

Update Size Analysis: For rank-one updates:

$$\|\Delta W^{(l)} h^{(l-1)}\| = \|\Delta W^{(l)}\|_{\text{op}} \|h^{(l-1)}\|$$

Learning Rate Scaling: To maintain $\Theta(1)$ updates as width increases:

$$\eta^{(l)} = \frac{\eta_0}{n_l}$$

where η_0 is a base learning rate independent of width.

7.4 Complete μP Recipe

Initialization Scaling:

- Output layer: $\sigma = \frac{1}{n}$
- Hidden layers: $\sigma = \frac{1}{\sqrt{n}}$
- Embeddings: $\sigma = 1$ (constant)

Learning Rate Scaling:

- Output layer: $\eta = \frac{\eta_0}{n}$
- Hidden layers: $\eta = \frac{\eta_0}{\sqrt{n}}$
- Embeddings: $\eta = \eta_0$ (constant)

Key Insight: Different layer types require different scaling relationships based on their role in the forward and backward pass.

8 μP Validation and Limitations

8.1 Third-Party Validation

Recent independent research has validated μP effectiveness while identifying limitations:

Robustness Testing:

- Works well for transformer language models
- Requires careful hyperparameter tuning for base scale
- Sensitive to architectural modifications
- Performance gains diminish with very large scale differences

Practical Considerations:

- Base model selection affects transferability
- Layer-specific learning rates add implementation complexity
- Not all architectures benefit equally

8.2 Implementation Challenges

Software Requirements: Need framework support for per-layer learning rates

Hyperparameter Sensitivity: Base learning rate selection still critical

Architecture Dependence: May require re-derivation for novel architectures

9 Practical Guidelines for Scaling

9.1 Hyperparameter Strategy Selection

Choose μP when:

- Planning multiple scale experiments
- Limited compute for large-scale hyperparameter search
- Using standard transformer architectures

Choose Direct Scaling when:

- Training single target scale
- Novel architectures without established μP derivations
- Sufficient compute for empirical validation

9.2 Learning Rate Schedule Design

WSD Advantages:

- Enables efficient Chinchilla analysis
- Flexible termination points
- Generally comparable performance to cosine

Cosine Advantages:

- Well-established performance
- Simpler implementation
- Extensive empirical validation

9.3 Scaling Law Validation

Priority Order for Validation:

1. **Chinchilla Analysis:** Most reliable, consistently clean results
2. **Batch Size Scaling:** Generally predictable, important for efficiency
3. **Learning Rate Scaling:** Use with caution, validate carefully

Minimum Validation Requirements:

- Span at least 2 orders of magnitude in compute
- Include multiple data points per scale
- Validate extrapolation accuracy on held-out scales

10 Future Directions and Open Questions

10.1 Emerging Challenges

Post-Training Scaling: How do scaling laws apply to RLHF, fine-tuning, and alignment?
Multimodal Scaling: Extension of principles to vision-language models
Mixture of Experts: Optimal scaling for sparse architectures

10.2 Research Opportunities

Alternative Parameterizations: Meta-P and other variants of μP
Downstream Task Scaling: Better correlation between perplexity and task performance
Efficiency Scaling: Scaling laws for inference cost, memory usage, and environmental impact

11 Mind Map: CS336 Lecture 11 - Scaling Laws Part 2

11.1 Visual Learning Framework

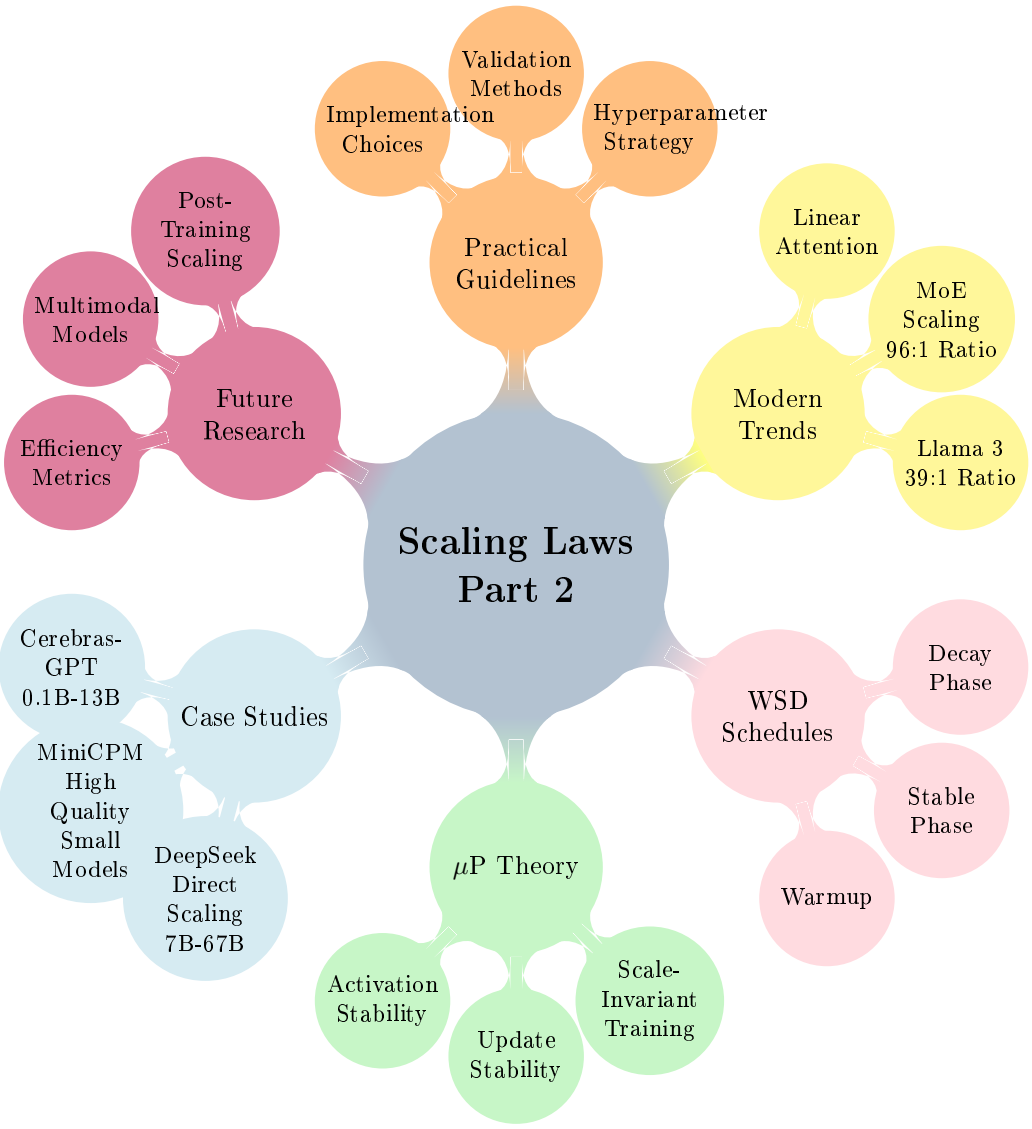

Figure 1: Comprehensive Mind Map: Scaling Laws in Practice

11.2 Mind Map Description and Navigation Guide

Branch Descriptions:

- **Case Studies (Light Blue):** Examines three major real-world implementations of scaling laws, showcasing how different organizations approach the practical challenges of training large language models at scale.
- **μP Theory (Light Green):** Covers the mathematical foundations of Maximal Update Parameterization, including the theoretical derivations for achieving scale-invariant training dynamics through proper initialization and learning rate scaling.

- **WSD Schedules (Light Red):** Details the Warm-up Stable Decay learning rate methodology that enables efficient Chinchilla analysis and flexible training termination points across different compute budgets.
- **Modern Trends (Yellow):** Highlights recent developments in scaling research, including updated optimal token-to-parameter ratios, mixture of experts scaling patterns, and architectural validation through scaling laws.
- **Practical Guidelines (Orange):** Provides actionable frameworks for practitioners to choose between different scaling methodologies, validate scaling laws, and implement stable hyperparameter transfer across scales.
- **Future Research (Purple):** Identifies emerging challenges and opportunities in scaling research, including post-training scaling, multimodal extensions, and efficiency-focused scaling metrics.

Learning Strategy: Use this mind map as a reference guide while studying. Start from any branch that aligns with your current interests or needs, then follow connections to related concepts. The visual structure reinforces the interconnected nature of scaling law theory and practice.