Title

Curriculum-Based Scaling Laws: Unveiling the Interplay Between Model Capacity and Training Complexity

Problem Statement

Current scaling laws for large language models assume a static training regime, neglecting the potential impact of evolving training curricula on model scaling behavior. This oversimplification may lead to suboptimal training strategies and incomplete understanding of the relationship between model capacity and learning dynamics.

Motivation

Existing approaches typically derive scaling laws based on models trained on a fixed dataset distribution, without considering how changes in the training curriculum might affect scaling dynamics. By analyzing how scaling behavior changes under different training curricula, we can potentially uncover more efficient scaling strategies and gain deeper insights into the relationship between data complexity and model capacity. This approach could lead to more comprehensive scaling laws that account for the interplay between model size, data complexity, and learning dynamics, potentially revolutionizing our understanding of large language model training.

Proposed Method

We propose a curriculum-based scaling law analysis framework. Our method involves: 1) Defining a spectrum of curricula, ranging from simple to complex tasks. 2) Training models of various sizes on different points along this curriculum spectrum. 3) Analyzing how scaling laws change as the curriculum complexity increases. 4) Investigating the existence of 'critical points' where increased model capacity suddenly enables learning of more complex patterns. 5) Developing a predictive model for optimal curriculum progression as a function of model size. 6) Exploring the interaction between curriculum progression and other training hyperparameters (e.g., learning rate schedules).

Step-by-Step Experiment Plan

Step 1: Define Curriculum Spectrum

Create a curriculum spectrum ranging from simple to complex tasks. For language models, this could include: a) Token prediction on simple sentences. b) Next sentence prediction. c) Cloze tasks. d) Question answering on simple facts. e) Multi-hop reasoning tasks. f) Complex problem-solving tasks.

Step 2: Prepare Datasets

For each point on the curriculum spectrum, prepare a dataset of appropriate difficulty. Ensure datasets are large enough to support training of models up to 100B parameters.

Step 3: Model Architecture

Use a standard Transformer architecture. Define a range of model sizes from 100M to 100B parameters, with logarithmically spaced intervals (e.g., 100M, 300M, 1B, 3B, 10B, 30B, 100B).

Step 4: Training Setup

For each model size and curriculum point, train the model using standard optimization techniques (e.g., Adam optimizer, learning rate warmup and decay). Use the same compute budget (in terms of FLOPS) for each model size to ensure fair comparison.

Step 5: Evaluation

Define a consistent evaluation set that covers the entire spectrum of task complexity. This set should remain constant across all experiments to allow for fair comparison.

Step 6: Scaling Law Analysis

For each curriculum point and model size, plot the evaluation performance against model size. Fit power law curves to these plots to derive scaling laws for each curriculum stage.

Step 7: Critical Point Analysis

Analyze the scaling curves to identify any 'critical points' where the scaling behavior changes significantly. This could indicate a sudden increase in the model's ability to learn more complex patterns.

Step 8: Curriculum Progression Model

Develop a predictive model that suggests the optimal curriculum stage for a given model size. This could be based on the critical points identified in Step 7.

Step 9: Hyperparameter Interaction

Investigate how the scaling behavior changes when varying other hyperparameters, particularly the learning rate schedule. Repeat Steps 4-7 with different learning rate schedules.

Step 10: Comparative Analysis

Compare the performance and efficiency of models trained with the curriculum-based approach to those trained on a static, mixed dataset. Analyze both final performance and training dynamics.

Step 11: Generalization Study

Test the models trained on different curriculum stages on tasks outside their training distribution to assess how curriculum-based training affects generalization ability.

Test Case Examples

Baseline Input

Train a 1B parameter model on a mixed dataset containing tasks from all complexity levels.

Baseline Expected Output

The 1B parameter model achieves X% accuracy on the evaluation set after T training steps.

Proposed Method Input

Train a 1B parameter model using the curriculum-based approach, starting from simple tasks and progressively moving to more complex ones.

Proposed Method Expected Output

The 1B parameter model achieves Y% accuracy on the evaluation set after T training steps, where Y > X. The model shows better performance on complex tasks compared to the baseline, indicating more efficient learning of complex patterns.

Explanation

The curriculum-based approach allows the model to build a strong foundation on simple tasks before tackling more complex ones, leading to better overall performance and potentially more efficient training.

Fallback Plan

If the proposed curriculum-based method doesn't show significant improvements over the baseline, we can pivot the project to an in-depth analysis of why the curriculum approach didn't work as expected. This could involve: 1) Analyzing the learning dynamics at different curriculum stages to understand where the method falls short. 2) Investigating whether certain types of tasks or patterns are more amenable to curriculum learning than others. 3) Exploring whether the benefits of curriculum learning are more pronounced for certain model sizes or architectures. 4) Examining the interaction between curriculum learning and other training techniques like pre-training or fine-tuning. This analysis could provide valuable insights into the nature of learning in large language models and potentially inspire new approaches to improving training efficiency.

Ranking Score: 6