Title

Causal Inference Framework for Deriving Neural Scaling Laws through Targeted Interventions

Problem Statement

Current scaling laws for large language models are largely correlational, making it difficult to infer causal relationships between model/data scaling and performance improvements. This limits our ability to make robust predictions about future model performance and optimize scaling strategies.

Motivation

Existing approaches typically observe scaling behavior passively, without performing controlled interventions to isolate the causal effects of different scaling factors. By performing targeted causal interventions, we can develop more robust and interpretable scaling laws that capture the true drivers of performance improvements. This will enable more accurate predictions of future model capabilities and more efficient allocation of resources for model development.

Proposed Method

We propose a causal inference framework for deriving scaling laws through targeted interventions. Our method involves: 1) Identifying key factors hypothesized to drive scaling behavior (e.g., model width, depth, training data quantity, data quality, training time). 2) Designing a series of controlled experiments where we intervene on one factor while controlling for others. 3) Utilizing causal inference techniques (e.g., instrumental variables, difference-in-differences) to isolate the causal effect of each factor on performance. 4) Developing a causal model that predicts performance improvements based on interventions across multiple factors. 5) Analyzing interaction effects and potential confounders in the causal model. 6) Using the causal model to simulate and optimize hypothetical scaling strategies.

Step-by-Step Experiment Plan

Step 1: Identify Key Scaling Factors

Based on literature review and expert knowledge, identify 5-7 key factors hypothesized to drive scaling behavior. These may include: model size (number of parameters), model width, model depth, training data quantity, training data quality, training compute, and training time.

Step 2: Design Controlled Experiments

For each identified factor, design a series of experiments that systematically vary that factor while controlling for others. For example, to study the effect of model width, create a series of models with increasing width while keeping depth, data, and compute constant. Aim for 5-10 data points per factor, covering a range of at least 2 orders of magnitude.

Step 3: Implement and Train Models

Using a common architecture (e.g., GPT-style decoder-only transformer), implement and train the series of models designed in Step 2. Use a consistent training setup (optimizer, learning rate schedule, etc.) across all experiments. Train models on a large-scale language modeling task (e.g., C4 or The Pile) for a fixed number of tokens or time.

Step 4: Evaluate Performance

Evaluate all trained models on a consistent set of downstream tasks. Include tasks that test different capabilities: 1) Language understanding (e.g., GLUE benchmark) 2) Knowledge and reasoning (e.g., MMLU, TruthfulQA) 3) Code generation (e.g., HumanEval) 4) Math problem-solving (e.g., GSM8K) 5) Few-shot learning (e.g., BIG-bench) Record performance metrics for each task and model configuration.

Step 5: Apply Causal Inference Techniques

For each scaling factor, apply causal inference techniques to estimate its effect on performance: 1) Use instrumental variables analysis, treating the intentional variation in the factor as an instrument. 2) Apply difference-in-differences analysis, comparing performance changes across factor levels. 3) Employ causal forests or other machine learning-based causal inference methods to estimate heterogeneous treatment effects.

Step 6: Develop Causal Model

Based on the results of Step 5, develop a causal model that predicts performance improvements as a function of interventions on multiple factors. Start with a simple additive model and progressively add interaction terms and non-linear effects as supported by the data. Use techniques like regularization and cross-validation to avoid overfitting.

Step 7: Analyze Interactions and Confounders

Examine the causal model for significant interaction effects between factors. Investigate potential confounding variables not explicitly controlled for in the experiments. Conduct sensitivity analyses to assess the robustness of causal estimates to unobserved confounding.

Step 8: Simulate and Optimize Scaling Strategies

Use the causal model to simulate the effects of different scaling strategies (e.g., increasing width vs. depth vs. data). Develop an optimization framework to identify efficient scaling paths given constraints on compute, data, or other resources. Compare these optimized strategies to current scaling approaches in the field.

Step 9: Validate on Held-Out Configurations

Test the predictive power of the causal scaling laws on a set of held-out model configurations not used in the initial experiments. This may include interpolated configurations as well as extrapolations to larger scales.

Step 10: Analyze and Report Results

Synthesize findings into a comprehensive analysis of causal scaling laws. Compare to existing correlational scaling laws and discuss implications for future model development. Prepare visualizations of key results and scaling predictions.

Test Case Examples

Baseline Prompt Input (Correlational Scaling Law)

Given a language model with 10 billion parameters trained on 1 trillion tokens, predict its performance on the MMLU benchmark.

Baseline Prompt Expected Output (Correlational Scaling Law)

Based on the power law scaling observed in previous studies, we predict the model will achieve a score of 65% on MMLU.

Proposed Prompt Input (Causal Scaling Law)

Given a language model with 10 billion parameters, predict its performance on MMLU if we intervene to increase its training data from 1 trillion to 10 trillion tokens, while holding all other factors constant.

Proposed Prompt Expected Output (Causal Scaling Law)

Our causal model predicts that increasing training data from 1T to 10T tokens will cause a performance improvement of 7.5 percentage points on MMLU, from 65% to 72.5%. This estimate accounts for the diminishing returns observed in our controlled experiments and isolates the causal effect of data scaling from confounding factors like model size increases that often accompany data scaling in practice.

Explanation

The causal scaling law provides a more precise and actionable prediction by isolating the effect of a specific intervention (increasing training data) while controlling for other factors. It also offers uncertainty estimates and accounts for non-linear scaling effects, providing a more nuanced and reliable forecast compared to simple extrapolation of correlational trends.

Fallback Plan

If the proposed causal inference framework fails to produce more accurate or interpretable scaling laws compared to existing methods, we can pivot the project in several directions: 1) Conduct a detailed analysis of where and why the causal approach fails, which could yield insights into the limitations of current experimental designs or the complexity of scaling phenomena. 2) Focus on developing improved experimental protocols for isolating individual scaling factors, which could benefit future scaling law research even if our specific causal model is not successful. 3) Investigate the possibility of unobserved confounders or complex interaction effects that our initial framework failed to capture, potentially leading to a more sophisticated multi-factor scaling model. 4) Explore alternative statistical techniques, such as structural equation modeling or Bayesian networks, that might be better suited to capturing the complex causal relationships in neural scaling. 5) Shift focus to comparing and reconciling different scaling behaviors across model architectures or tasks, which could provide valuable insights even without establishing clear causal relationships.

Ranking Score: 6