Pushing the Limits of Scaling Laws in the Age of Generative Models

Azalia Mirhoseini
Assistant Professor of Computer Science, Stanford
The Age of Generative AI!

Large Language Models (LLMs) and Generative Vision Models are disrupting how we live:

- Education
- Software Engineering
- Productivity
- Law
- Finance
- Healthcare
- Art, and more

Midjourney’s AI generates images

GPT4 writes an iOS app

Anthropic’s Claude says its goal is to be a helpful, harmless, and honest assistant

The New York Times
https://www.nytimes.com/2023/03/08/technology/c...]
The Chatbots Are Here, and the Internet Industry Is in a Tizzy

Forbes
Generative AI ChatGPT Still Winning Hearts And Minds Over
The Driving Force Behind Generative AI is **Scaling**!

Sevilla et al., “Compute Trends Across Three Eras of Machine Learning”, 2022
Scaling Laws for Language Models

Performance of language models can almost precisely be predicted as a function of:

- Amount of compute used
- Dataset size
- Number of parameters in the model

Kaplan et al., Scaling Laws for Neural Language Models, 2020
Scaling Laws Goes Beyond Language Models

Henighan et al., Scaling Laws for Autoregressive Generative Modeling, 2020
The Significance of Scaling Laws

- Empirically "guaranteed" continual progress with scaling
- Emergent Behavior: Capabilities that only emerge in larger models
- Systematic compute allocation
Emergent Behavior: Zero/Few Shot Learning

**Zero-shot**
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1. Translate English to French:
2. cheese => ........................................
```

**Few-shot**
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1. Translate English to French:
2. sea otter => loutre de mer
3. peppermint => menthe poivrée
4. plush giraffe => girafe peluche
5. cheese => ........................................
```
Few Shot Learning Emerges in Larger Models

Brown et al., GPT-3, 2020
Emergent Behavior: Chain of Thought

Nye et al., Show Your Work: Scratchpads for Intermediate Computation with Language Models, 2021
Wei et al., Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, 2022 (Figure is from this publication)
Chain of Thought Emerges in Large Models

Wei et al., Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, 2022
Emergent Behavior: Ability to Solve Entirely New Tasks

We refer to Wei et al., Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, 2022.
Scaling Laws Enable Systematic Compute Allocation

Capability prediction on 23 coding problems

Mean Log Pass Rate

- Observed
- Prediction
- gpt-4

OpenAI 2023, GPT4
How Can We Push the Limits of Scaling?
Deep Learning Systems

Different modalities of data and associated training mechanisms ...

Data

Model

Software

Hardware

Neural networks and methodologies such as ResNet, Transformers, graph neural networks, SL, RL

Compilers and software libraries such as XLA, MLIR, TF, Pytorch for accelerators from edge to cloud

Accelerators such as Google TPUs, NVIDIA GPUs, SambaNova, Cerebras, Graphcore, ...
This Talk: Pushing the Limits of Scaling Laws in the Age of Generative Models

- Reinforcement Learning from Human Feedback, Constitutional AI
- Co-design from Application to Hardware
- Sparse and Dynamic Neural Networks
- Automating Design Cycle
This Talk: Pushing the Limits of Scaling Laws in the Age of Generative Models

- Reinforcement Learning from Human Feedback, Constitutional AI
- Co-design from Application to Hardware
- Sparse and Dynamic Neural Networks
- Automating Design Cycle
Traditional LLM Training and Finetuning

Training data is text (the internet, books, transcripts) and optionally human augmented prompts.

Training objective is to predict the next word.
We Can Use Human Preferences to Finetune LLMs

- Data quality greatly impacts learning efficiency and scaling performance
- We can determine data quality through human ranking
- Humans are asked to rank the outputs of the model based on various criteria, for example:
  - Usefulness
  - Harmfulness
  - Truthfulness

Stiennon et al., 2020, Learning to summarize from human feedback
Reinforcement Learning from Human Feedback (RLHF)

We Can Use Reinforcement Learning to Bring Human Preferences to Training

1. Collect Human Feedback (e.g. Do you prefer summary A or B?)
2. Train a Reward Model based on Human Feedback (e.g. Given summaries A and B, the neural net model predicts human label)
3. Use RL to Finetune the LLMs, Maximizing the Reward Model (e.g. LLM generates summaries that maximize the reward model)

Bai et al., 2022, Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback
RLHF Can Drastically Improve Scaling

1. Stiennon et al., 2020, Learning to summarize from human feedback
2. Bai et al., 2022, Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback
Constitutional AI: Self-Improving AI Using AI Feedback

In “Constitutional AI”, the LLM follows a Constitution (set of principles written by a human) to generate feedback for self-improvement.
Constitutional AI: Self-Improving AI Using AI Feedback

In “Constitutional AI”, the LLM follows a Constitution (set of principles written by a human) to generate feedback for self-improvement

1. **Supervised learning (SL):** Finetune LLM on data generated by self-critique and revisions
Constitutional AI: Self-Improving AI using AI Feedback

In “Constitutional AI”, the LLM follows a Constitution (set of principles written by a human) to generate feedback for self-improvement

1. **Supervised learning (SL):** Finetune LLM on data generated by self-critique and revisions

   **Example:**

   **Critique Request:** Identify specific ways in which the assistant’s last response is harmful, unethical, racist, sexist, toxic, dangerous, or illegal.

   **Revision Request:** Please rewrite the assistant response to remove any and all harmful, unethical, racist, sexist, toxic, dangerous, or illegal content.
Constitutional AI: Self-Improving AI using AI Feedback

In “Constitutional AI”, the LLM follows a Constitution (set of principles written by a human) to generate feedback for self-improvement

1. **Supervised learning (SL):** Finetune LLM on data generated by self-critique and revisions

1. **Reinforcement learning from AI feedback (RLAIF):**
   a. Train a preference model based on LLM (from Step 1) responses and the constitution
   b. Finetune the LLM to maximize the preference model

Bai et al., 2022, Constitutional AI: Harmlessness from AI Feedback (Used in Anthropic’s Claude)
Constitutional AI: Self-Improving AI using AI Feedback

In “Constitutional AI”, the LLM follows a Constitution (set of principles written by a human) to generate feedback for self-improvement

1. **Supervised learning (SL):** Finetune LLM on data generated by self-critique and revisions

1. **Reinforcement learning from AI feedback (RLAIF):**
   a. Train a preference model based on LLM (from Step 1) responses and the constitution
   b. Finetune the LLM to maximize the preference model

*Example:* Choose the response that answers the human in the most thoughtful, respectful and cordial manner.

Bai et al., 2022, Constitutional AI: Harmlessness from AI Feedback (Used in Anthropic’s Claude)
Constitutional AI Improves Scaling over RLHF

In “Constitutional AI”, the LLM follows a Constitution (set of principles written by a human) to generate feedback for self-improvement.

Bai et al., 2022, Constitutional AI: Harmlessness from AI Feedback
This Talk: Pushing the Limits of Scaling Laws in the Age of Generative Models

- Human Feedback and AI Feedback with Reinforcement Learning
- Sparse and Dynamic Neural Networks
- Co-design from Application to Hardware
- Automating Design Cycle
Can We Train More Efficient Models by Introducing Sparsity

- Traditionally, neural networks are dense:
  - Each input is processed by the entire model

- Mixture-of-Experts (MoE) is a dynamic model
  - Large weight matrices are replaced with a mixture of smaller weight matrices (“experts”)
  - A gating function routes inputs to only a small number of experts

MoEs for Transformer based Language Models

Fedus et al., Switch Transformers, 2021
Deep Mixture of Experts

- **2017**: Introduced sparsely gated Mixture of Experts (MoEs), and trained the first 100B parameter language model.
- **2019**: MoEs shown to be effective for both language and vision tasks.
- **2022**: Used in GLaM: an MoE-based 1T+ parameter LLM by Google.
  - ~2x more efficient training and inference than GPT-3
- **2023**: GPT4 is reportedly an MoE based model!

Noam Shazeer*, Azalia Mirhoseini*, Krzys Maziarz*, Andy Davis, Quoc Le, Geoffrey Hinton, Jeff Dean, Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer, ICLR 2017


Nan Du et al, GLaM: Efficient Scaling of Language Models with Mixture-of-Experts, ICML 2022
Pushing the Limits of Scaling from a Software and Hardware Perspective

- **Extreme HW/SW co-design for generative models**
  - If generative models serve billions of users, the economy of scale calls for further customization

- **Automated and fast design cycle**
  - Currently it takes 2-3 years to design a new generation of accelerators, slowing down customization and adaptation to new models
Why Co-Design Matters for Deep Learning?
New Models May Break Software and Hardware Assumptions

- Example: EfficientNets vision models
- Despite its low FLOP count, runtime is high
- TPUv3 was not designed for EfficientNet!

<table>
<thead>
<tr>
<th>Operation</th>
<th>FLOP %</th>
<th>Runtime %</th>
</tr>
</thead>
<tbody>
<tr>
<td>DepthwiseConv2D</td>
<td>5.00%</td>
<td>65.30%</td>
</tr>
<tr>
<td>Conv2D</td>
<td>94.67%</td>
<td>34.20%</td>
</tr>
<tr>
<td>Other</td>
<td>0.33%</td>
<td>0.50%</td>
</tr>
</tbody>
</table>

FLOP: Floating Point Operation

Tan and Le, EfficientNet, 2019
FAST: A Full-Stack Custom Accelerator Search Framework

- Designs custom accelerators for a single workload, or a mixture of workloads
- Addresses compute and memory bottlenecks
- Searches a space of $O(10^{2300})$
  - Datapath: $\sim 10^{11}$ search space
  - Compiler: $\sim 10^{300}$ search space
  - Scheduler: $\sim 10^{2000}$ search space

Dan Zhang, Safeen Huda, Ebrahim Songhori, Quoc Le, Anna Goldie, Azalia Mirhoseini,
A Full Stack Search Technique for Domain Optimized Deep Learning Accelerators, ASPLOS 2022
FAST’s Comprehensive Datapath Design Space

A superset template capable of describing scalar, vector, and matrix processors with a flexible memory hierarchy

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Potential Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEs_x_dim</td>
<td>int</td>
<td>1 to 256, powers of 2</td>
</tr>
<tr>
<td>PEs_y_dim</td>
<td>int</td>
<td>1 to 256, powers of 2</td>
</tr>
<tr>
<td>Systolic_array_x</td>
<td>int</td>
<td>1 to 256, powers of 2</td>
</tr>
<tr>
<td>Systolic_array_y</td>
<td>int</td>
<td>1 to 256, powers of 2</td>
</tr>
<tr>
<td>Vector_Unit_Multiplier</td>
<td>int</td>
<td>1 to 16, powers of 2</td>
</tr>
<tr>
<td>L1_buffer_config</td>
<td>enum</td>
<td>Private, Shared</td>
</tr>
<tr>
<td>L1_input_buffer_size</td>
<td>int</td>
<td>1KB to 1MB, powers of 2</td>
</tr>
<tr>
<td>L1_weight_buffer_size</td>
<td>int</td>
<td>1KB to 1MB, powers of 2</td>
</tr>
<tr>
<td>L1_output_buffer_size</td>
<td>int</td>
<td>1KB to 1MB, powers of 2</td>
</tr>
<tr>
<td>L2_buffer_config</td>
<td>enum</td>
<td>Disabled, Private, Shared</td>
</tr>
<tr>
<td>L2_input_buffer_multiplier</td>
<td>int</td>
<td>1x to 128x, powers of 2</td>
</tr>
<tr>
<td>L2_weight_buffer_multiplier</td>
<td>int</td>
<td>1x to 128x, powers of 2</td>
</tr>
<tr>
<td>L2_output_buffer_multiplier</td>
<td>int</td>
<td>1x to 128x, powers of 2</td>
</tr>
<tr>
<td>L3_global_buffer_size</td>
<td>int</td>
<td>0MB to 256MB, powers of 2</td>
</tr>
<tr>
<td>GDDR6_channels</td>
<td>int</td>
<td>1 to 8, powers of 2</td>
</tr>
<tr>
<td>Native_batch_size</td>
<td>int</td>
<td>1 to 256, powers of 2</td>
</tr>
</tbody>
</table>
Efficient Fusion is Key for Properly Evaluating Datapaths

- Accelerator performance is a function of its hardware datapath and how workloads are mapped onto that datapath

- Designed a new ILP-based fusion technique to address memory bandwidth:
  - A new fusion technique capable of fusing the entire model to reduce access to off-chip DRAM
  - Inter-layer activations and weights stay in on-chip SRAM

![Diagram of Conv2D and Element-wise Op fusion with ILP notation](image)

ILP: Integer Linear Programming
FAST Design for EfficientNet-B7

An example architecture found using FAST with a Perf/TDP objective

<table>
<thead>
<tr>
<th></th>
<th>TPUv3 (die-shrunk)</th>
<th>FAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>MXU Dimensions</td>
<td>128x128</td>
<td>32x32</td>
</tr>
<tr>
<td>Num MXUs</td>
<td>2x2</td>
<td>64</td>
</tr>
<tr>
<td>Global Buffer Size</td>
<td>2x16MiB</td>
<td>128MiB</td>
</tr>
<tr>
<td>Compute Utilization</td>
<td>0.14</td>
<td>0.61</td>
</tr>
<tr>
<td>Fusion Efficiency</td>
<td>0%</td>
<td>85%</td>
</tr>
<tr>
<td>QPS</td>
<td>210</td>
<td>733</td>
</tr>
<tr>
<td>Perf/TDP</td>
<td>1</td>
<td>3.91</td>
</tr>
</tbody>
</table>

QPS: Queries Per Second
TDP: Thermal Design Power
Co-Design is Key to Unlock Multiplier Gains

Datapath, scheduling, and fusion impact vary across workloads
FAST Search Results: Single Model

- Perf/TDP improvements of ~1.8X to ~6X vs TPUv3
- For reference, a ~2-3x increase between two generations of an accelerator is considered a success
FAST Search Results: Mixture Models

- Yellow bars: customize for a mixture of five models
  - EfficientNet-B7, ResNet50, OCR-RPN, OCR-Rec, BERT-1024
  - 2.4X geometric mean improvement in Perf/TDP
FAST: An Automated Full-Stack HW/SW Co-Design Framework

- FAST: Synthesizing accelerators by searching an $O(10^{2300})$ datapath, compiler, and scheduler space
- Customizing for one or a family of workloads can lead to significant performance improvements
- ROI analysis demonstrated that custom accelerators can be ROI-positive for moderate-size deployments

Dan Zhang, Safeen Huda, Ebrahim Songhori, Quoc Le, Anna Goldie, Azalia Mirhoseini,
A Full Stack Search Technique for Domain Optimized Deep Learning Accelerators, ASPLOS 2022
Chip Placement Problem

- A chip typically has dozens of blocks
- Each block is a netlist with thousands of memory and millions of logic nodes
- Placement problem:
  - Place nodes of a netlist while optimizing for design constraints, e.g., power, timing, area
Chip Placement is Challenging and Important

- An NP-hard problem
- Takes months to design production placements
- Each day incurs $XM in labor and opportunity cost
Chip Placement with Reinforcement Learning

- **RL agent iteratively optimizes node placements**
- **Action:** Placing the current node on a grid cell
- **Reward:** A weighted average of total wirelength, density, and congestion
- **State:** Embeddings of chip netlist and canvas
Chip Placement with RL is Extremely Challenging!
Complexity of Chip Placement Problem

Chess
Number of states $\sim 10^{123}$

Go
Number of states $\sim 10^{360}$

Chip Placement
Number of states $\sim 10^{9000}$
Chip Placement with Reinforcement Learning is Even Harder

- **Long episode lengths**: There are millions of nodes to place
- **Complex rewards**: EDA tools are slow and expensive
- **Limited access to prior data**: Most chip designs are confidential
- **Hard to generalize**: Unlike Go and Chess, the board, pieces, rules, and win conditions of the “game” change from chip to chip
Reducing the Complexity of RL Optimization Space

- Shortened RL episode length:
  - Policy places the macros (up to thousands)
  - Analytical solver places millions of standard cells: leveraging their negligible area
Reducing the Complexity of RL Optimization Space

- **Shortened RL episode length:**
  - Policy places the macros (up to thousands)
  - Analytical solver places millions of standard cells: leveraging their negligible area
- **Sped up evaluation time:**
  - Designed fast congestion, wirelength, and density costs that correlate with EDA tools
Reducing the Complexity of RL Optimization Space

- **Shortened RL episode length:**
  - Policy places the macros (up to thousands)
  - Analytical solver places millions of standard cells: leveraging their negligible area

- **Sped up evaluation time:**
  - Designed fast congestion, wirelength, and density costs that correlate with EDA tools

- **Sped up data collection and training through parallel computing**
Edge-GNN: A New Edge-Based Graph Neural Network for Learning from Chips

```plaintext
while Not converged do
    Update edge: \( e_{ij} = f_1(\text{concat}[f_0(v_i)|f_0(v_j)|w_{ij}]) \)
    Update node: \( v_i = \text{mean}_{j \in N(v_i)}(e_{ij}) \)
end
```

*Node type: One-hot category \{Hard macro, soft macro\}
Circuit Training Optimization Cost Function

\[ J(\theta, G) = \frac{1}{K} \sum_{g \sim G} E_{g, p \sim \pi_\theta} [R_{p,g}] \]

- **J**: Cost function
- **\( R_{p,g} \)**: Reward for Placement \( p \) on Chip \( g \)
- **G**: Set of training chips
- **K**: Number of chips in G
- **\( \theta \)**: RL policy’s parameters
Results on a TPU-v4 Block

Human Expert

Time taken: ~6-8 weeks
Total wirelength: 57.07m
Route DRC violations: 1766

Circuit Training

Time taken: 24 hours
Total wirelength: 55.42m (-2.9% shorter)
Route DRC violations: 1789 (+23, negligible difference)

DRC: Design Rule Checking
New Insights From Circuit Training

Circuit Training broke conventional wisdom: e.g., alignment, macro hierarchy, while producing superhuman results.
Circuit Training Improves as More Chip Netlists are Used In Training

Huge Opportunity: Policy is “Gaining Experience”

Placement cost is a function of wirelength, density, and congestion (lower is better)
Real-World Impact on Accelerator Design

- One of the earliest real-world productionizations of a deep RL method
- Used to design 4 generations of TPUs, saving thousands of engineering hours

Open-sourced: github.com/google-research/circuit_training
This Talk: Pushing the Limits of Scaling Laws in the Age of Generative Models

- Human Feedback and AI Feedback with Reinforcement Learning
- Sparse and Dynamic Neural Networks
- Co-design, from Application to Hardware
- Automating Design Cycle
Summary

- Large generative models are changing the way we work and live!
- Scaling of data, model size, and compute consistently leads to new AI capabilities
- There are many opportunities to improve scaling across the deep learning stack, from data, all the way to hardware design
- AI itself will play a big role in accelerating this scaling!
Thank You!