Pre-training: Scaling

CSC2541H1 Topics in Machine Learning, Winter 2025, UToronto

Chris J. Maddison

Announcements

- If you are assigned to **present on Feb 7, come talk** about presentations after class.
- Launched a **team-finding form for the course project**. Please fill it in now, if you want help finding a team for the project. I will contact interested folks this weekend.
- **Project and proposal handouts are up.**

Questions?

Recap

- We started building a Transformer last week.
  - **Tokenization** – converting text to sequences of integers
  - **Batching** – input / output matrices that capture batches of next token prediction problems
  - **Named arrays** – syntactic sugar for JAX arrays that allows us to track axes
  - **Bigram model** – predict the next token conditioned just on the current token
- Want to practice?
  - Implement an encoder-decoder Transformer with named arrays
  - Implement Llama 3 architecture

This week

- Continue building up our Transformer from scratch
- **Pre-training: Scaling**; Llama 3 tech report, sec 1-3.2

Llama 3: Take-homes from the team

Key levers for high-quality foundation models

- **Data:**
  - “careful pre-processing and curation pipelines for pre-training data and the development of more rigorous quality assurance and filtering approaches for post-training data”
- **Scale:**
  - “a flagship model with 405B trainable parameters on 15.6T text tokens”
- **Managing complexity:**
  - “We make design choices that seek to maximize our ability to scale the model development process”

Llama 3

Architecture

- Standard dense Transformer with particular choices of
  - normalization
  - activation
  - attention
  - positional embedding
- **But basically the same architecture we've built!**

	8B	70B	405B
Layers	32	80	126
Model Dimension	4,096	8192	16,384
FFN Dimension	14,336	28,672	53,248
Attention Heads	32	64	128
Key/Value Heads	8	8	8
Peak Learning Rate	3×10^{-4}	1.5×10^{-4}	8×10^{-5}
Activation Function	SwiGLU		
Vocabulary Size	128,000		
Positional Embeddings	RoPE ($\theta = 500,000$)		

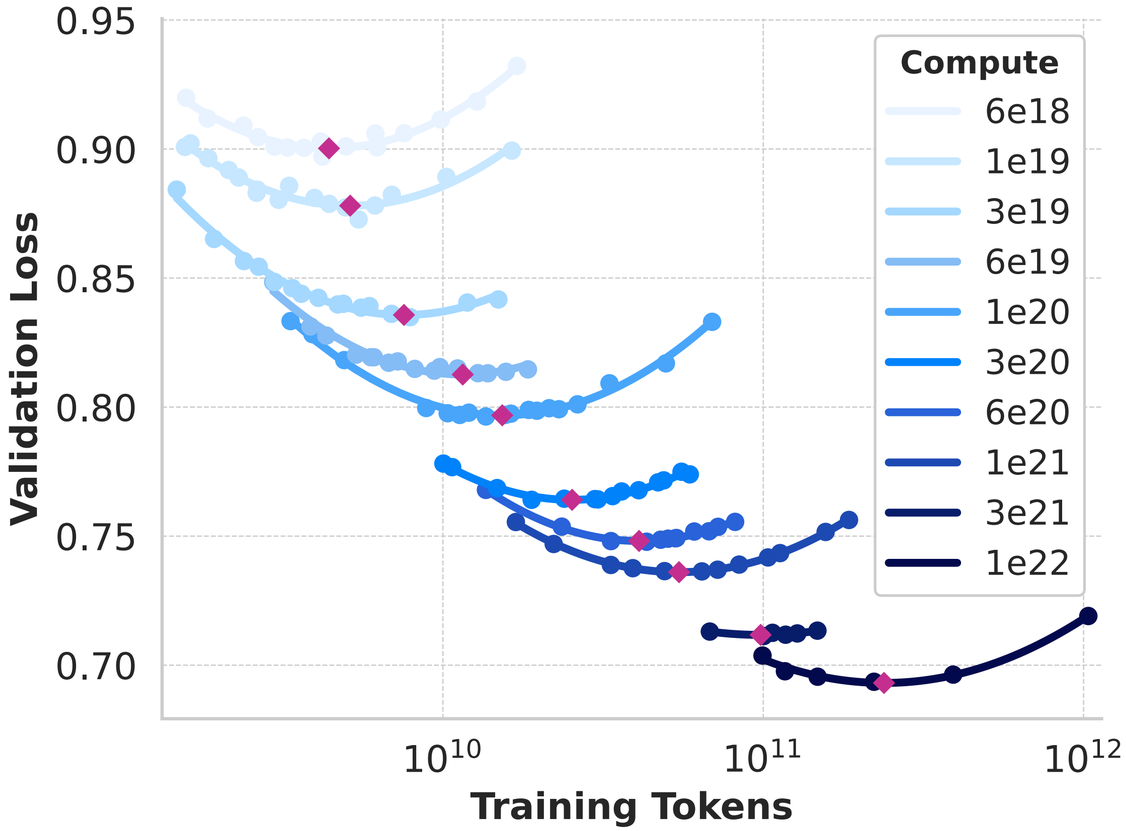
Llama 3

Compute optimal scaling

- The FLOPs needed to train a Transformer with single-pass SGD scale like $6ND$ where D = parameter count and N = number of tokens in the data.
  - Let $\hat{w}^*_{N,D} \in \mathbb{R}^D$ be the result of SGD on N tokens.
- What is the best we can do under a compute constraint?

$$N^*(c), D^*(c) = \arg\min_{N,D} \mathbb{E}\left[R(\hat{w}^*_{N,D})\right] \text{ s.t. } 6ND \leq c$$

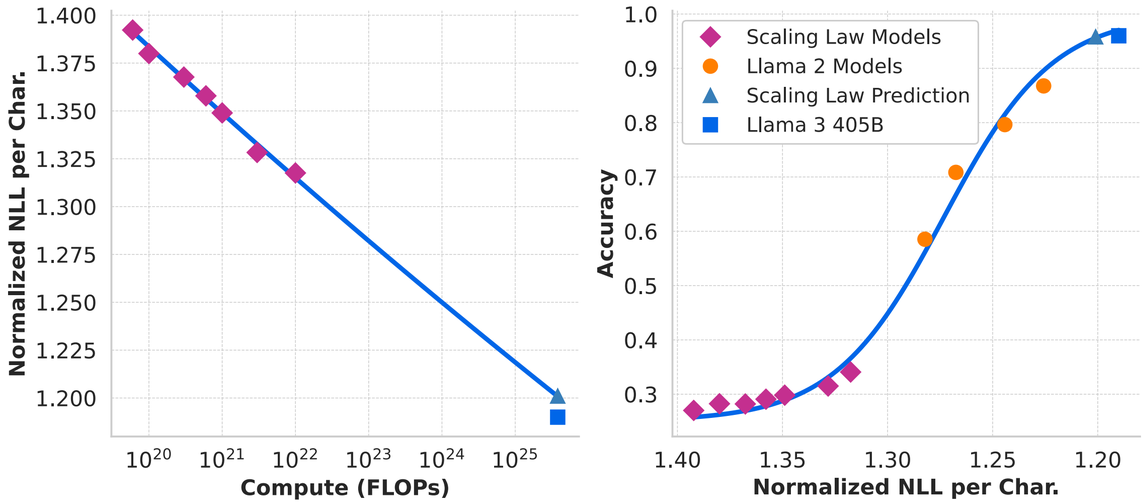
- **Chinchilla estimation**:
  - train models at different combinations of N and D
  - estimate $N^*(c)$ by optimizing over a compute level sets (IsoFLOP curves fitted to loss outcomes)
  - $6D^*(c) = c/N^*(c)$



Llama 3

Emergence of capabilities

- Capabilities are sometimes measured as accuracy on benchmarks.
- To forecast capabilities, we want to predict accuracy from compute. Llama team:
  - correlated train compute with normalized, negative log-likelihood per char. (similar to log-loss) on benchmark
  - then fitted a sigmoid relating NLL to accuracy
- Warning: this works on ARC Challenge, but other benchmarks are less predictable!

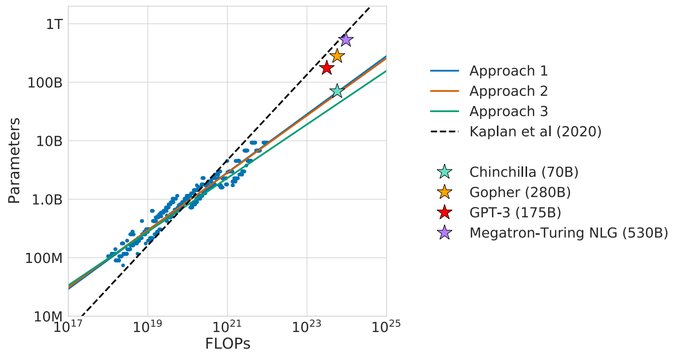


Notice: log-likelihood improves linearly with compute, capabilities follow an s-curve, i.e., **benchmarks have dynamic ranges**.

Today’s presentations

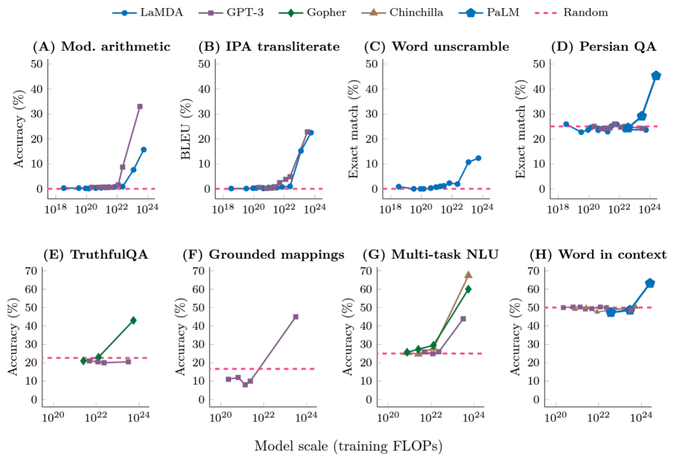
Topics

Compute-optimal scaling



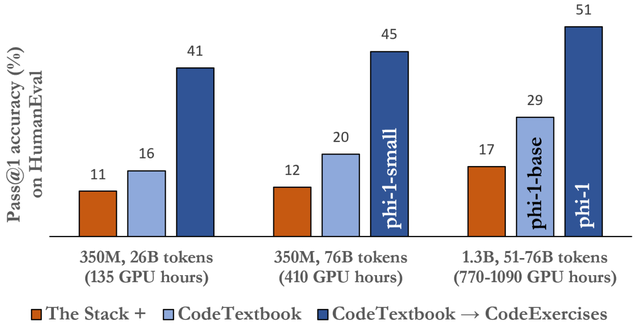
Hoffman et al, 2022

Emergent capabilities



Wei et al, 2022

Data quality



Gunasekar et al, 2023