Scaling laws for large language models

What & Why?

Given a fixed compute budget (C), how should one trade-off model size (N) and the number of training tokens (D)?

Accounting for both *training* and *inference*, how does one minimise the cost required to produce and serve a high quality model?

An analogous formulation to Moore's Law for transistor density.

Kaplan & McCandlish et al., 2020

Performance has a power-law relationship with each of N, D, and C when not bottlenecked by the other two.

$$L(N, D) = \left[\left(\frac{N_c}{N} \right)^{\frac{\alpha_N}{\alpha_D}} + \left(\frac{D_c}{D} \right) \right]^{\alpha_D}$$

Kaplan, Jared, et al. "Scaling laws for neural language models." arXiv preprint arXiv:2001.08361 (2020)

Kaplan & McCandlish et al., 2020

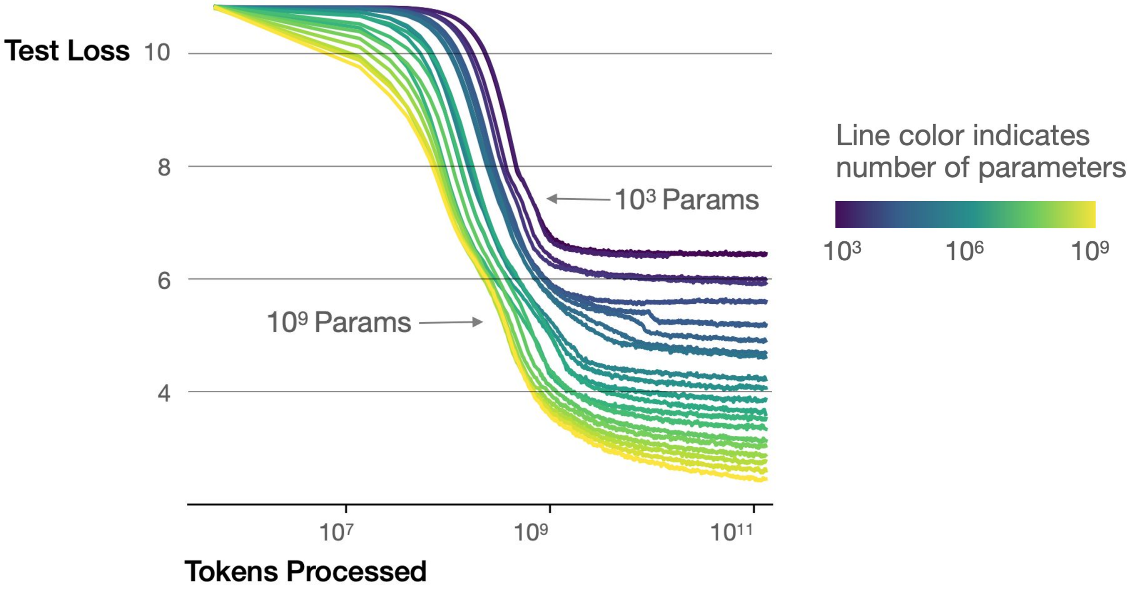
An estimate of the total non-embedding training compute C in terms of N, and D is given by,

$$C \approx \overbrace{(\underbrace{2N}_{\text{Forward Pass}} + \underbrace{2 \times 2N}_{\text{Backward Pass}^{1}})}^{\text{Compute per token}} \cdot D = 6ND$$

[1] https://sites.krieger.jhu.edu/jared-kaplan/files/2019/04/ContemporaryMLforPhysicists.pdf

Sample Efficiency of Large Models

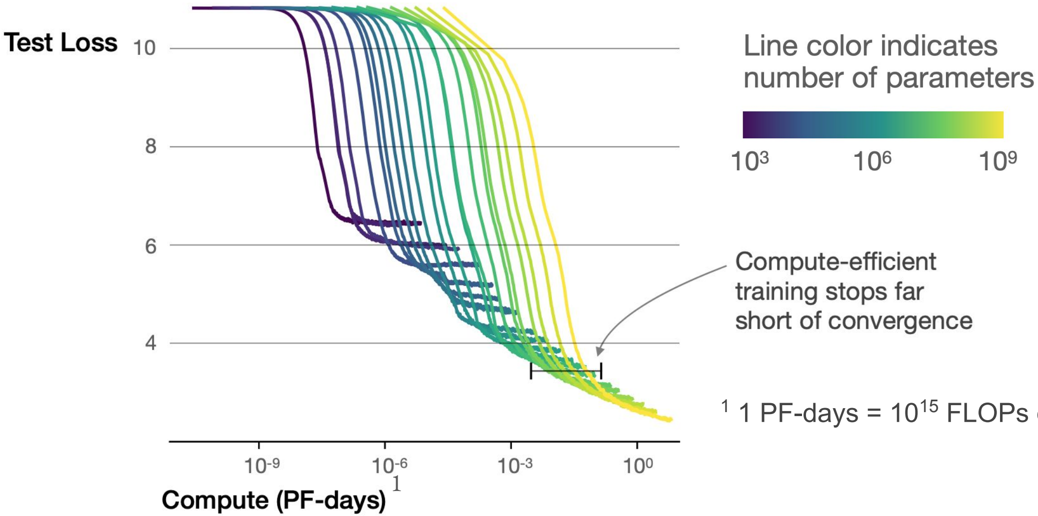
Larger models require fewer samples to reach the same performance.



Kaplan, Jared, et al. "Scaling laws for neural language models." arXiv preprint arXiv:2001.08361 (2020)

Convergence is Inefficient

Optimal performance can be attained by training *very large models* and stopping *significantly short of convergence*.



[1] 1 PF-days = 10^{15} FLOPs every second for 24 hours

Kaplan, Jared, et al. "Scaling laws for neural language models." arXiv preprint arXiv:2001.08361 (2020)

Optimal Allocation of Compute

As compute budget increases, optimal allocation is *larger models*, not more data or longer training.

For example, a 10x increase in compute should be allocated as:

- a 5x increase in model size
- a 2x increase in data size
    - ~ a 1.86x increase in batch size
    - ~ a 1.07x increase in number of steps

Kaplan, Jared, et al. "Scaling laws for neural language models." arXiv preprint arXiv:2001.08361 (2020)

Hoffmann et al., 2022

$$L(N, D) = E + \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}}$$

Irreducible error   Model error   Data error

Hoffmann, Jordan, et al. "Training compute-optimal large language models." arXiv preprint arXiv:2203.15556 (2022)

Hoffmann et al., 2022

Optimising the loss function L under the compute budget C = 6ND, the compute-optimal model size and dataset size are obtained as,

$$N_{\text{opt}}(C) = G\left(\frac{C}{6}\right)^{\frac{\beta}{\alpha+\beta}} \qquad D_{\text{opt}}(C) = G^{-1}\left(\frac{C}{6}\right)^{\frac{\alpha}{\alpha+\beta}}$$

$$\text{where } G = \left(\frac{\alpha A}{\beta B}\right)^{\frac{1}{\alpha+\beta}}$$

Hoffmann, Jordan, et al. "Training compute-optimal large language models." arXiv preprint arXiv:2203.15556 (2022)

Optimal Allocation of Compute

Unlike Kaplan et al., 2020, Hoffman et al., 2022 recommend that given a 10x increase in compute,
the model size and number of training tokens should be scaled in *equal proportions*.

Approach	Coeff. a where $N_{opt} \propto C^a$	Coeff. b where $D_{opt} \propto C^b$
1. Minimum over training curves	0.50 (0.488, 0.502)	0.50 (0.501, 0.512)
2. IsoFLOP profiles	0.49 (0.462, 0.534)	0.51 (0.483, 0.529)
3. Parametric modelling of the loss	0.46 (0.454, 0.455)	0.54 (0.542, 0.543)
Kaplan et al. (2020)	0.73	0.27

Hoffmann, Jordan, et al. "Training compute-optimal large language models." arXiv preprint arXiv:2203.15556 (2022)

Chinchilla Optimal

"For the compute budget used to train *Gopher*, the optimal model [*Chinchilla*] should be 4 times smaller, while being trained on 4 times more tokens."

Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
Chinchilla	70 Billion	1.4 Trillion

MMLU 5-shot accuracy

Random	25.0%
Average human rater	34.5%
GPT-3 5-shot	43.9%
Gopher 5-shot	60.0%
***Chinchilla* 5-shot**	**67.6%**
Average human expert performance	*89.8%*
June 2022 Forecast	57.1%
June 2023 Forecast	63.4%

Hoffmann et al. "Training compute-optimal large language models." arXiv preprint arXiv:2203.15556 (2022)

Compute Overhead

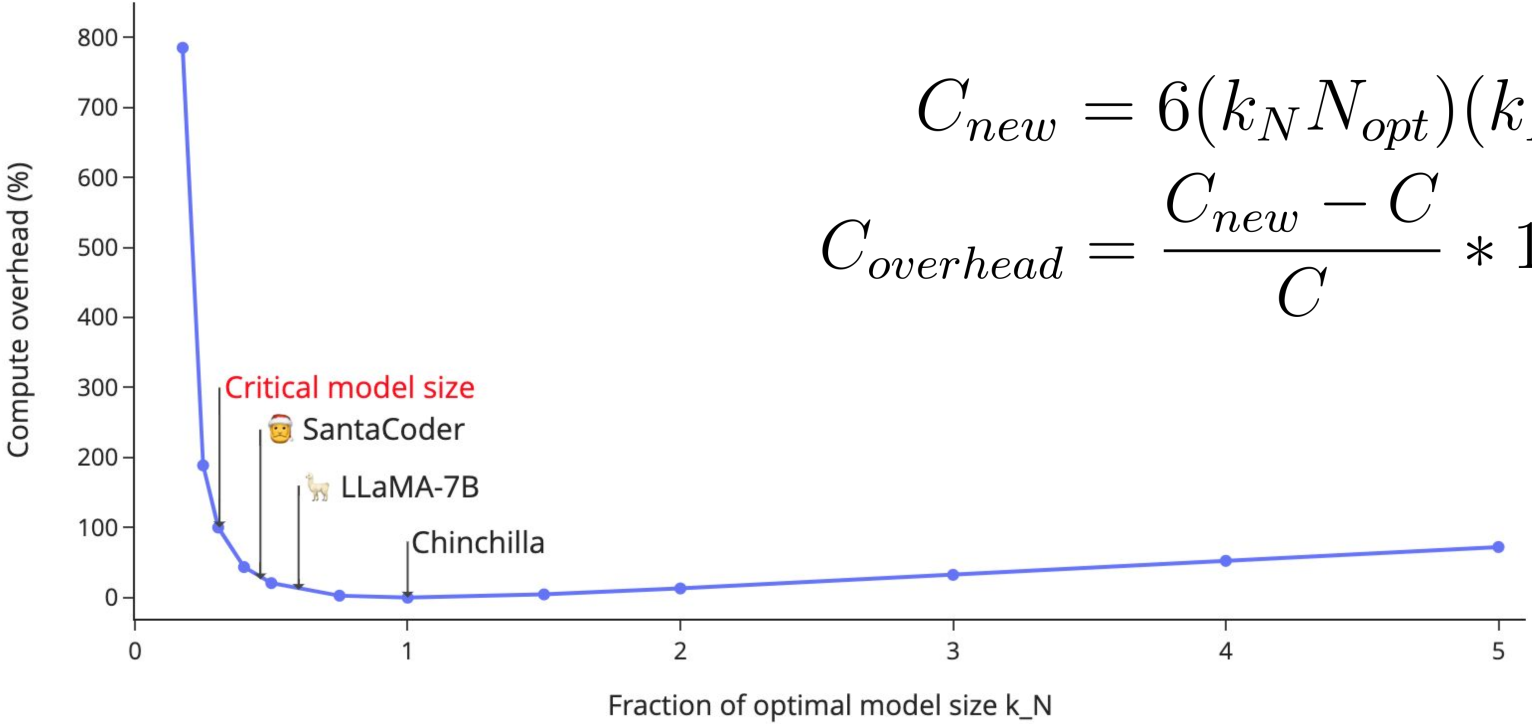
If we reduce the model size by k_N, by how much do we need to increase the number of tokens and compute to obtain the same loss?

$$k_D = \left(1 - (k_N^{-\alpha} - 1)\frac{AN_{opt}^{-\alpha}}{BD_{opt}^{-\beta}}\right)^{\frac{1}{-\beta}}$$

k_D turns out to be independent of the compute budget!

De Vries, Harm, "Go smol or go home". https://www.harmdevries.com/post/model-size-vs-compute-overhead

Compute Overhead



$$C_{new} = 6(k_N N_{opt})(k_D D_{opt})$$

$$C_{overhead} = \frac{C_{new} - C}{C} * 100.$$

De Vries, Harm, "Go smol or go home". https://www.harmdevries.com/post/model-size-vs-compute-overhead

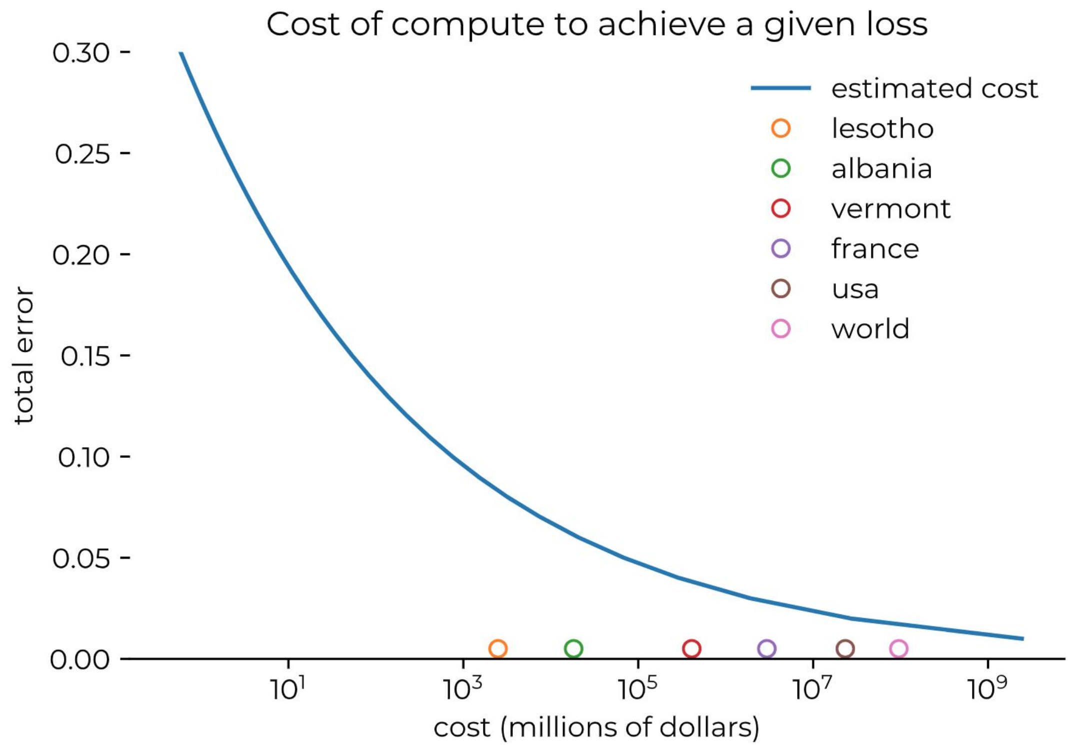
Training-Inference Compute Trade-off

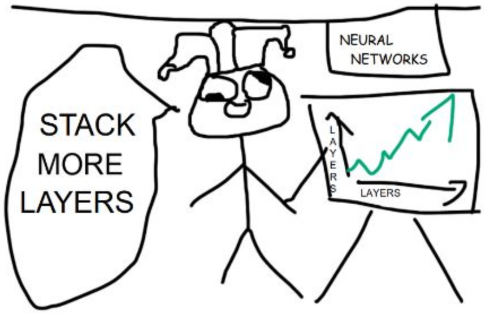
Chinchilla scaling laws only account for the computational cost of *training*.[1]

It may be beneficial to train far beyond Chinchilla optimal to reduce inference costs.

[1] Sardana, Nikhil, and Jonathan Frankle. "Beyond Chinchilla-Optimal: Accounting for Inference in Language Model Scaling Laws." arXiv preprint arXiv:2401.00448 (2023).

Do we have enough compute?





https://dynomight.net/scaling

Do we have enough tokens?

We are projected to run out of language data between 2030 & 2050.[1]

Model	Stock of data (#words)	Growth rate
Recorded speech	1.46e17 [3.41e16; 4.28e17]	5.2% [4.95%; 5.2%]
Internet users	2.01e15 [6.47e14; 6.28e15]	8.14% [7.89%; 8.14%]
Popular platforms	4.41e14 [1.21e14; 1.46e15]	8.14% [7.89%; 8.14%]
CommonCrawl	9.62e13 [4.45e13; 2.84e14]	16.68% [16.41%; 16.68%]
Indexed websites	2.21e14 [5.16e13; 6.53e15]	NA
Aggregated model	**7.41e14 [6.85e13; 7.13e16]**	**7.15% [6.41%; 17.49%]**

[1] Villalobos et al. "Will we run out of data? An analysis of the limits of scaling datasets in Machine Learning." arXiv preprint arXiv:2211.04325 (2022)

Future

Quality of data

Multi-modal training

Translation of loss into real-world performance

Sparsely activated gated Mixture-of-Expert models