Training compute-Optimal Large Language Models

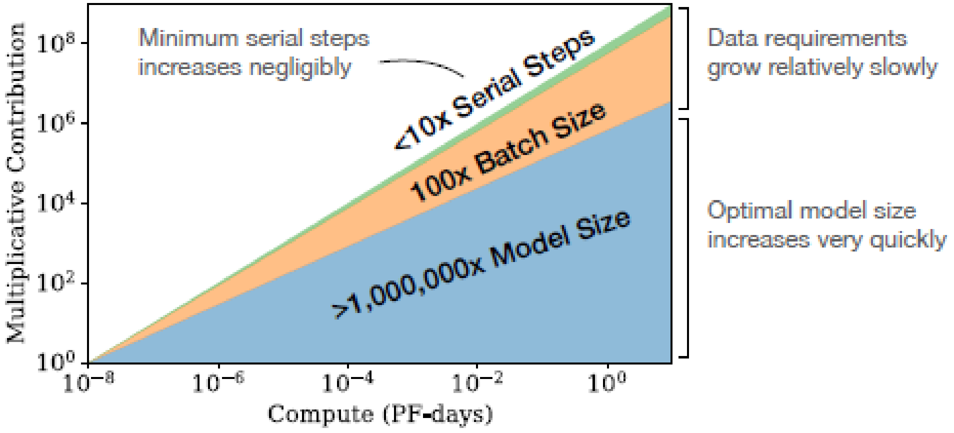
CPSC 670

Chenxi Huang

01/31/2023

Background

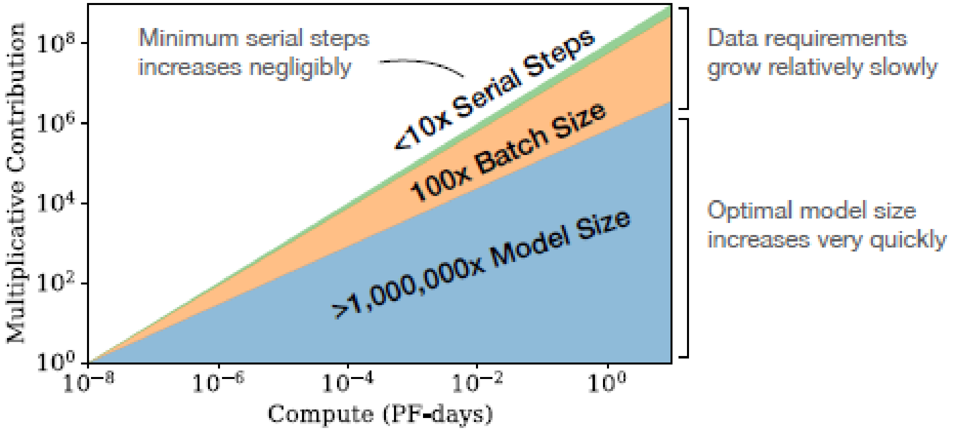
- Scaling law proposed in Kaplan et al. 2020



Given a 10x increase in compute budget:
5.5x model size N
1.8x training tokens D

Background

- Scaling law proposed in Kaplan et al. 2020



Given a 10x increase in compute budget:

5.5x model size N

1.8x training tokens D

- Increasingly-large models

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

Introduction

- Re-approach the question:

Given a compute budget, what is the optimal model size (N) and no.training tokens (D) for achieving minimum loss?

Introduction

- Re-approach the question:

Given a compute budget, what is the optimal model size (N) and no.training tokens (D) for achieving minimum loss?

- Overall approach: empirically estimate N_{opt} and D_{opt} based on the losses of models with diff sizes and no.training tokens

Introduction

- Re-approach the question:

Given a compute budget, what is the optimal model size (N) and no.training tokens (D) for achieving minimum loss?

- Overall approach: empirically estimate N_{opt} and D_{opt} based on the losses of models with diff sizes and no.training tokens
- Difference from Kaplan et al. 2020
  - Kaplan et al. used a fixed no.steps and learning rate schedule
  - Kaplan et al. included smaller models

Approaches

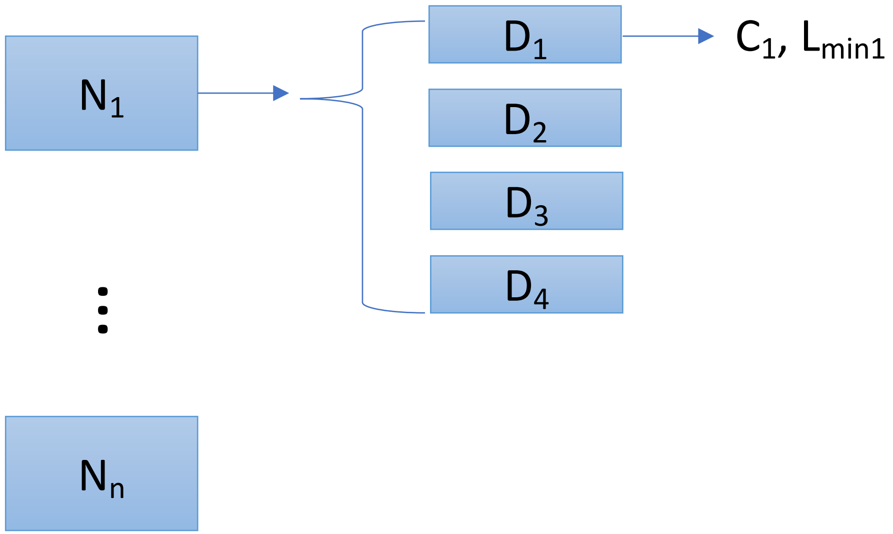
- 3 different approaches
- Training dataset: MassiveText

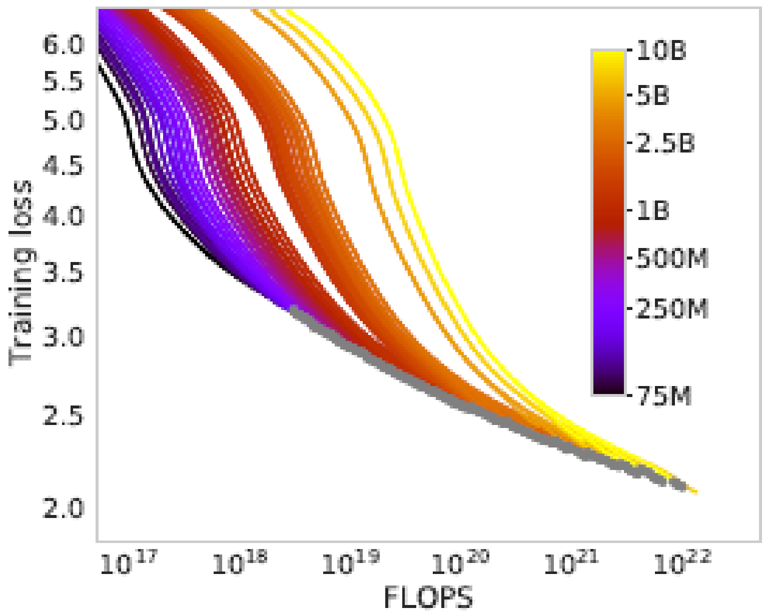
	Disk Size	Documents	Sampling proportion	Epochs in 1.4T tokens
MassiveWeb	1.9 TB	604M	45% (48%)	1.24
Books	2.1 TB	4M	30% (27%)	0.75
C4	0.75 TB	361M	10% (10%)	0.77
News	2.7 TB	1.1B	10% (10%)	0.21
GitHub	3.1 TB	142M	4% (3%)	0.13
Wikipedia	0.001 TB	6M	1% (2%)	3.40

- Cosine schedule, learning rate drops 10x, length match target training steps.

Approaches

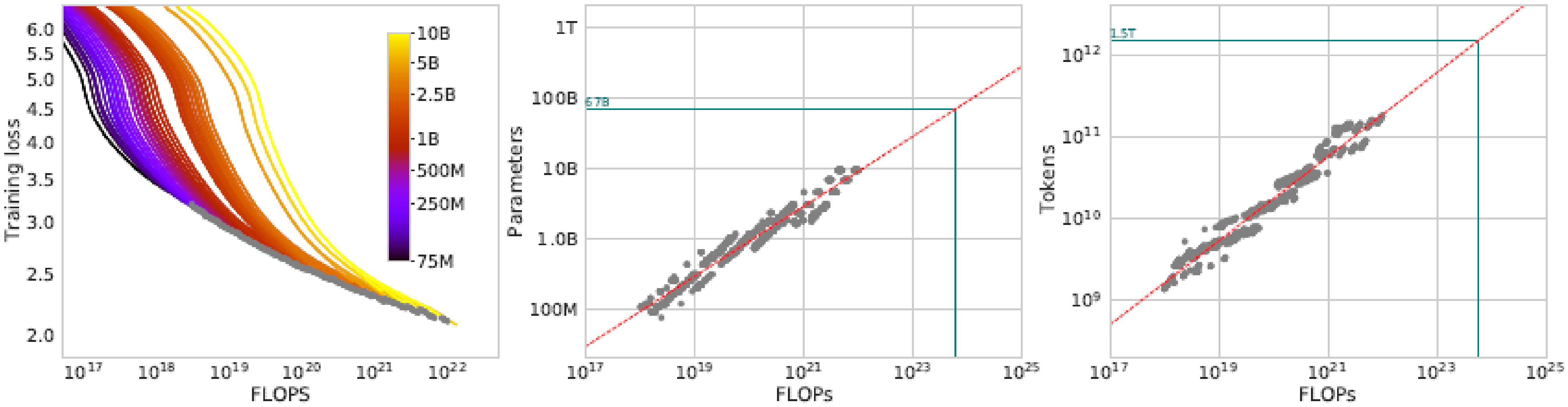
- Approach 1
  - Fix model size and vary number of training tokens





Approaches

- Approach 1
  - Fix model sizes and vary number of training tokens



Approaches

- Approach 1
  - Fix model sizes and vary number of training tokens

$$N_{opt} \propto C^a \text{ and } D_{opt} \propto C^b$$

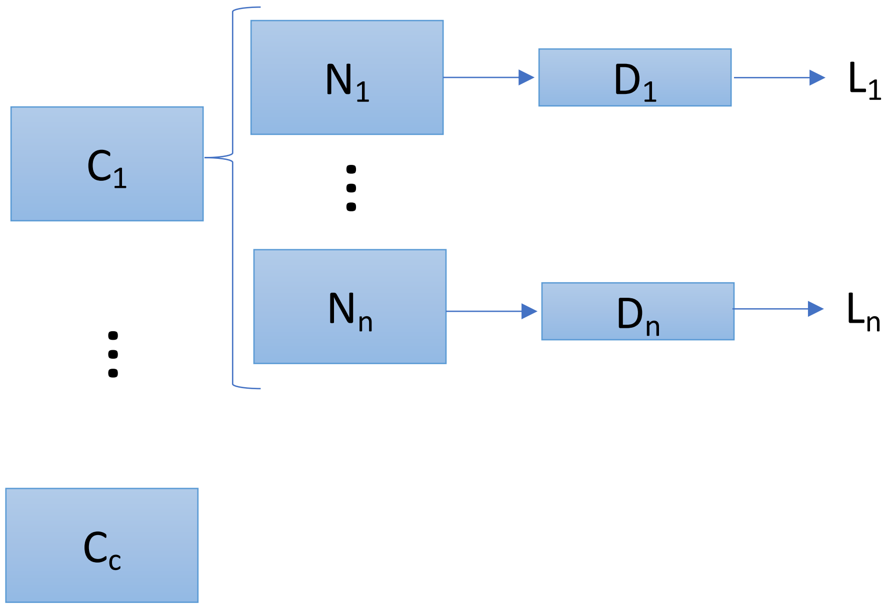
Approach	Coeff. a where $N_{opt} \propto C^a$	Coeff. b where $D_{opt} \propto C^b$
1. Minimum over training curves	0.50 (0.488, 0.502)	0.50 (0.501, 0.512)

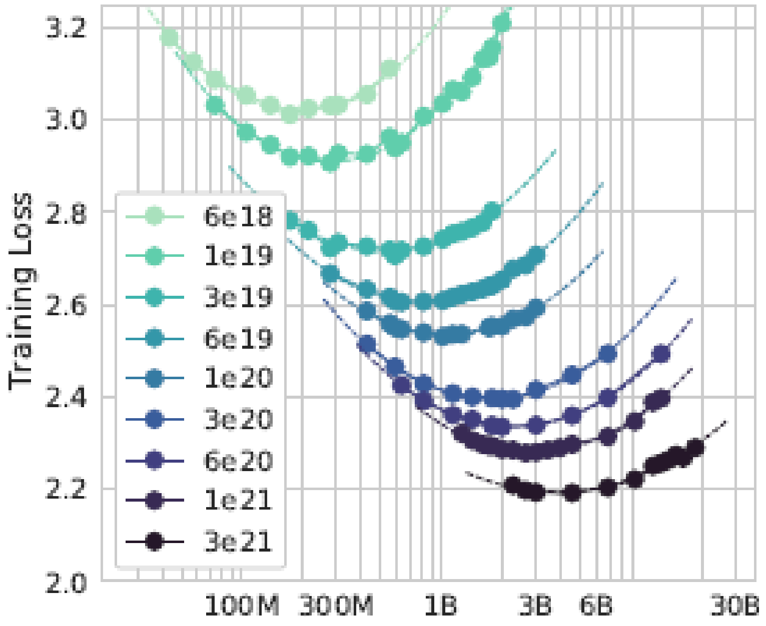
Approaches

- Approach 2
  - Fix training compute and vary model size

Approaches

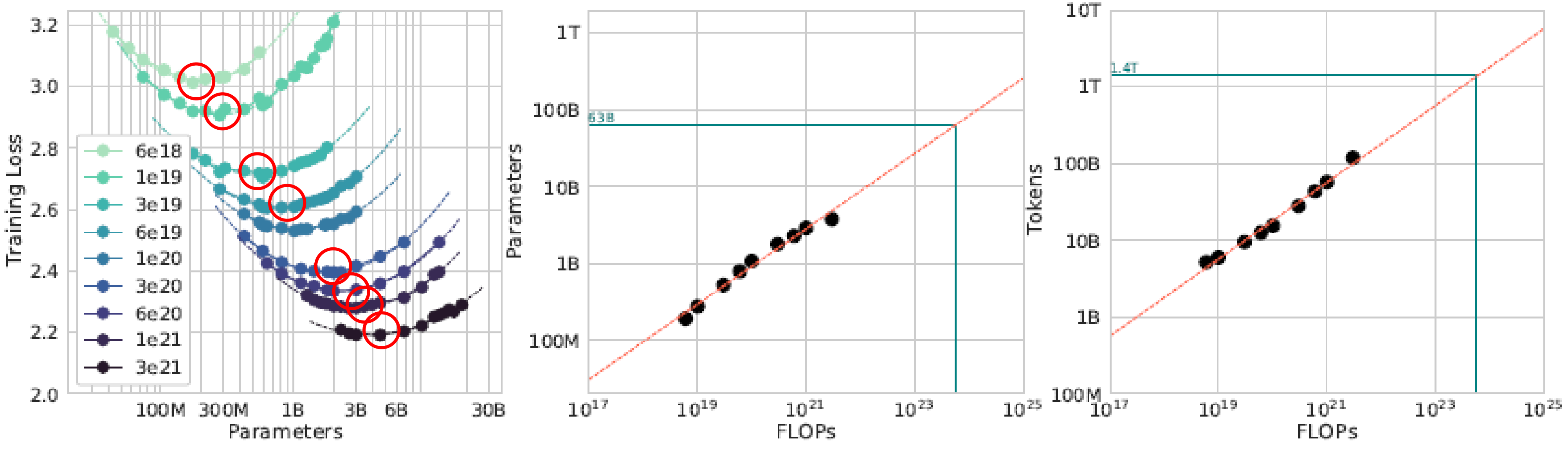
- Approach 2
  - Fix training compute and vary model size





Approaches

- Approach 2
    - Fix training compute and vary model size



Approaches

- Approach 2
  - Fix training compute and vary model size

$$N_{opt} \propto C^a \text{ and } D_{opt} \propto C^b$$

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)

Approaches

- Approach 3
  - Fit a parametric loss function
  - Take all final losses from Approach 1 and 2

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

Entropy of natural text

Loss due to approximating by model of N parameters

Loss due to only training on a finite number of training tokens

Approaches

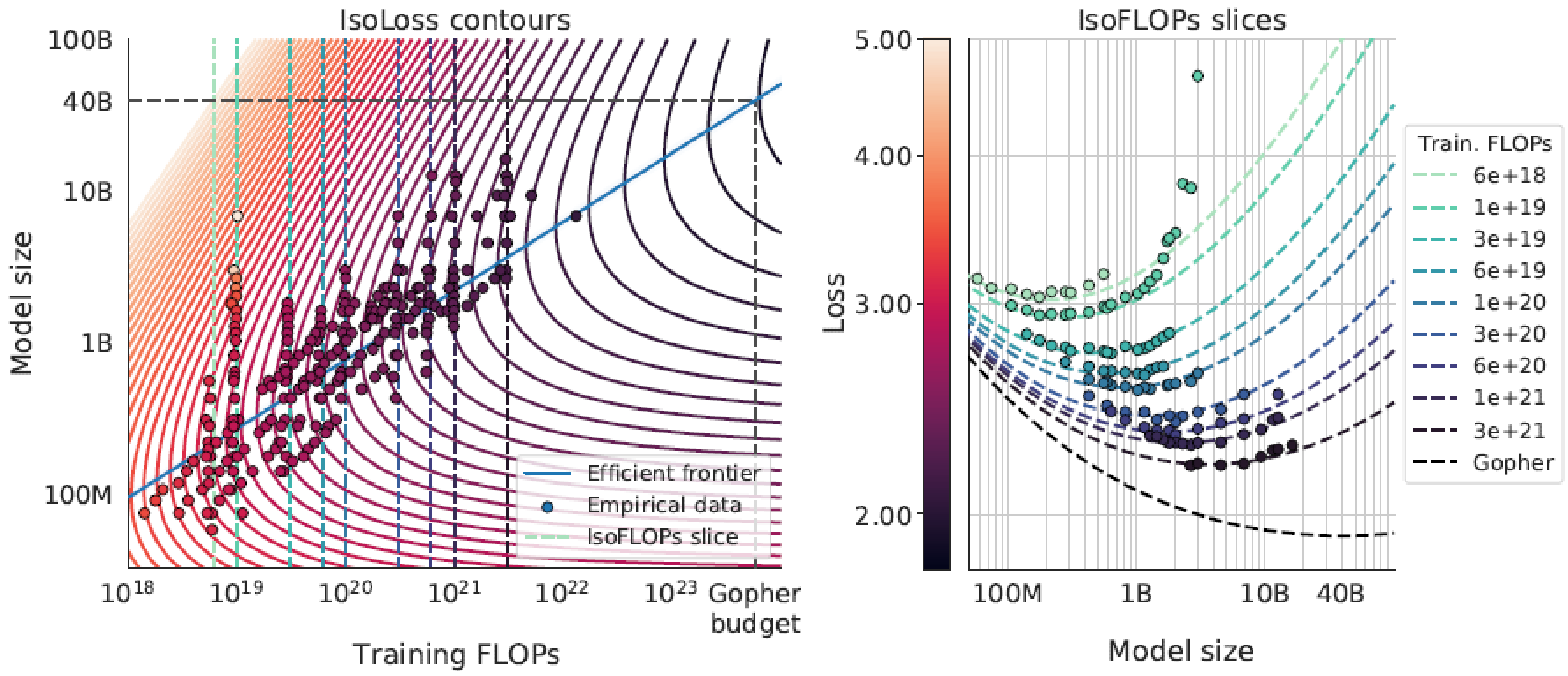
- Approach 3
  - Fit a parametric loss function
  - Take all final losses from Approach 1 and 2

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

$$\min_{A,B,E,\alpha,\beta} \quad \sum_{\text{Runs } i} \text{Huber}_{\delta}\Big(\log \hat{L}(N_i, D_i) - \log L_i \Big)$$

Approaches

- Approach 3
  - Fit a parametric loss function



Approaches

- Approach 3
  - Fit a parametric loss function

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

  - Minimize $\hat{L}(N, D)$ under the constraint $\text{FLOPs}(N, D) \approx 6ND$

$$N_{opt}(C) = G\left(\frac{C}{6}\right)^{a}, \quad D_{opt}(C) = G^{-1}\left(\frac{C}{6}\right)^{b}, \quad \text{where} \quad G = \left(\frac{\alpha A}{\beta B}\right)^{\frac{1}{\alpha+\beta}}, \quad a = \frac{\beta}{\alpha+\beta}, \text{ and } b = \frac{\alpha}{\alpha+\beta}.$$

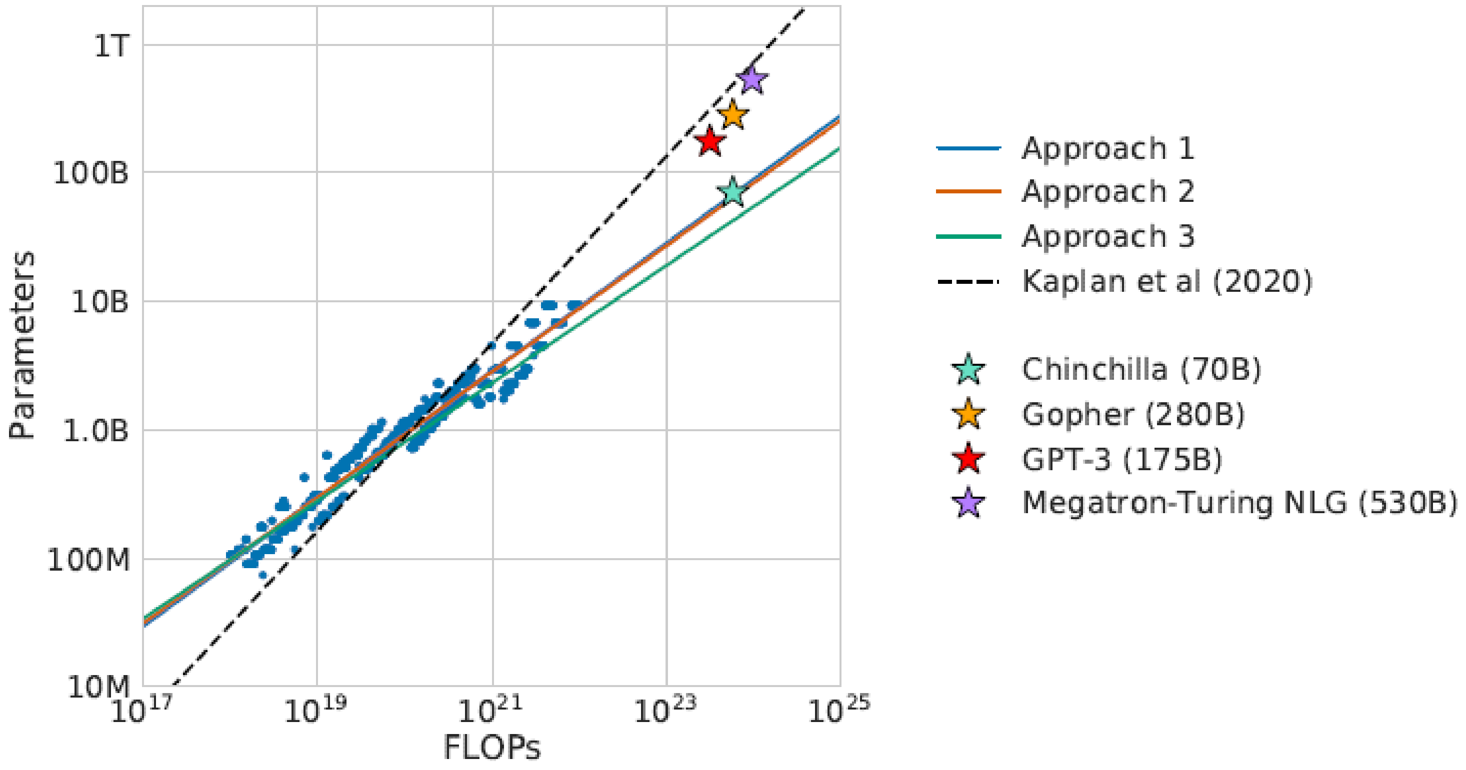
Approaches

- Approach 3
  - Fit a parametric loss function

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

Approaches

- Summary



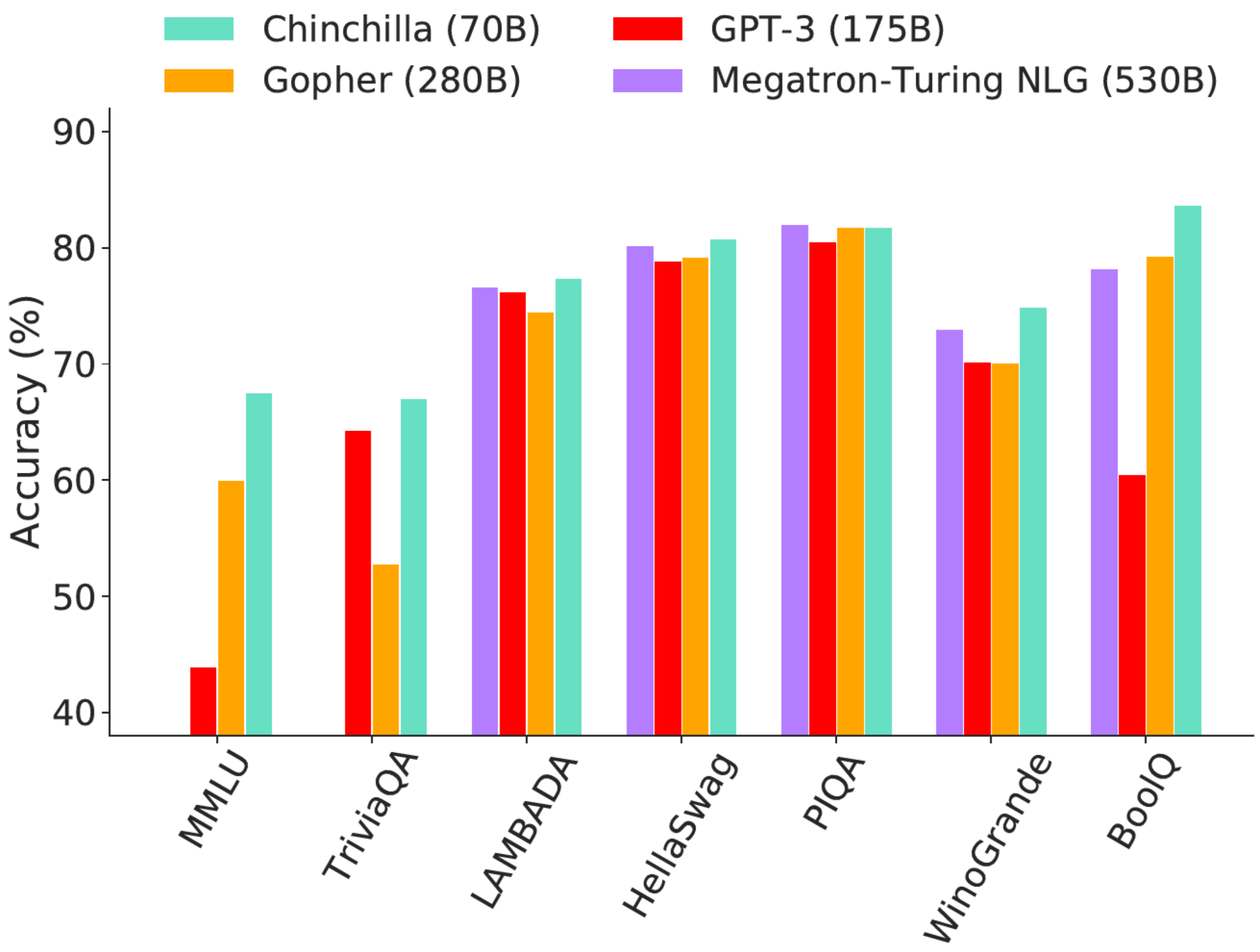
Approaches

- Chinchilla
  - Same compute budget as Gopher
  - N=70B, D=1.4T
  - Same architecture and training setup as Gopher with some difference
  - Evaluation:

	# Tasks	Examples
Language Modelling	20	WikiText-103, The Pile: PG-19, arXiv, FreeLaw, ...
Reading Comprehension	3	RACE-m, RACE-h, LAMBADA
Question Answering	3	Natural Questions, TriviaQA, TruthfulQA
Common Sense	5	HellaSwag, Winogrande, PIQA, SIQA, BoolQ
MMLU	57	High School Chemistry, Astronomy, Clinical Knowledge, ...
BIG-bench	62	Causal Judgement, Epistemic Reasoning, Temporal Sequences, ...

Approaches

- Chinchilla
  - Results



Implications

- Establish an optimal training paradigm for auto-regressive language models on a given compute budget
- Current large models are undertrained and underperforming
- Chinchilla
  - is smaller and performs better
  - has smaller memory footprint and less computation for fine-tuning and inference
- Increased focus on data instead of model size
- …