Training Compute-Optimal Large Language Models

Krushi Karukala
Rezwan Mahmud

Research question

Given a fixed FLOPs budget, how should one trade-off model size and the number of training tokens?

$$N_{opt}(C), D_{opt}(C) = \underset{N,D \text{ s.t. } \text{FLOPs}(N,D)=C}{\operatorname{argmin}} L(N, D).$$

Introduction

- This paper investigates the optimal model size and number of tokens for training a transformer language model under a given compute budget.

- By training over 400 language models ranging from 70 million to over 16 billion parameters on 5 to 500 billion tokens, it has been found that, for a compute-optimal training, the model size and the number of training tokens should be scaled equally.

- Using the above hypothesis, a compute optimal model, Chinchilla with 70B parameters is tested and it has been found that Chinchilla outperforms Gopher(280B), GPT-3(175B), Jurassic-1(178B) and Megatron-Turing NLG(530B) on a large range of downstream tasks.

About Kaplan

- Previously, Kaplan et al. (2020) showed that there is a power law relationship between the number of parameters in an autoregressive language model(LM) and its performance.
- I.e., If there is a 10x increase in computational budget, it has been suggested to increase the size of model 5.5x, and the number of training tokens should increase by 1.8x only.
- As a result, larger and larger models are being trained expecting performance improvements.
- Instead, this paper suggests that model size and training tokens should be scaled in equal proportions.

Related Work: Estimating hyperparameters for large language models

- What attributes do we need to decide?
  - **Training FLOPS**
  - **Model size (# of Parameters)**
  - **Number of training tokens**
  - Learning rate [Yang et al. (2021)]
  - Batch size [Yang et al. (2021)]
  - Width-to-depth ratio [Levine et al. (2020)]

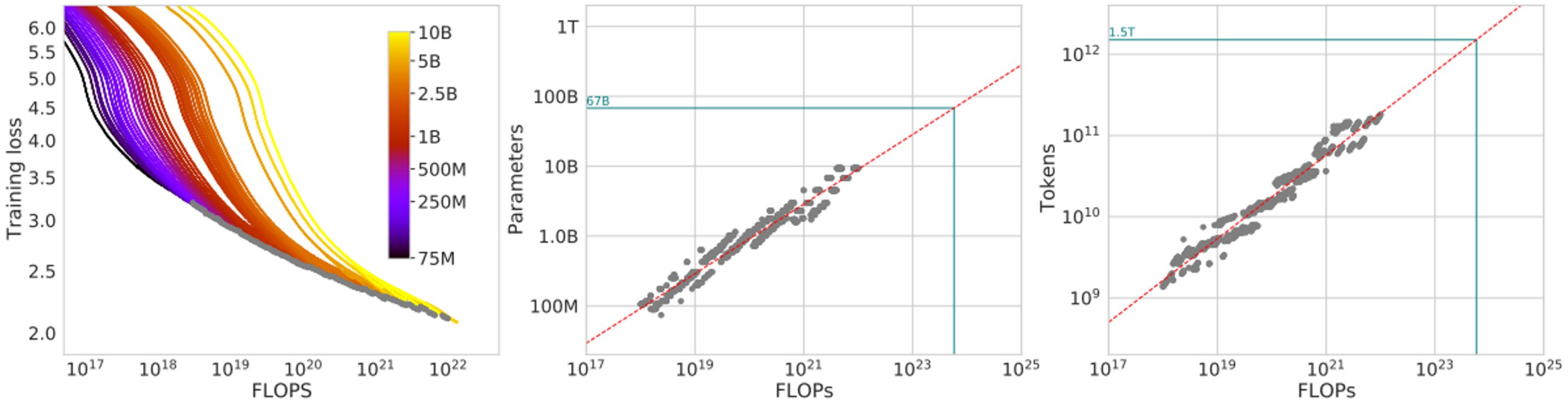
Related Work: About Gopher

- Google subsidiary DeepMind announced Gopher, a 280-billion-parameter AI natural language processing (NLP) model.
- Gopher is based on Transformer architecture.
- It has been trained on a 10.5TB corpus called MassiveText.
- Gopher outperformed the current state-of-the-art on 100 of 124 evaluation tasks.

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

- Tested on a fixed family of models (ranging from 70M to over 10B parameters)
- For each parameter count N they trained 4 different models
- For each run, they smoothed and then interpolated the training loss curve.
- From that, they obtain a continuous mapping from FLOP count to training loss for each run
- Finally, for each FLOP count, they determined which run achieved the lowest loss

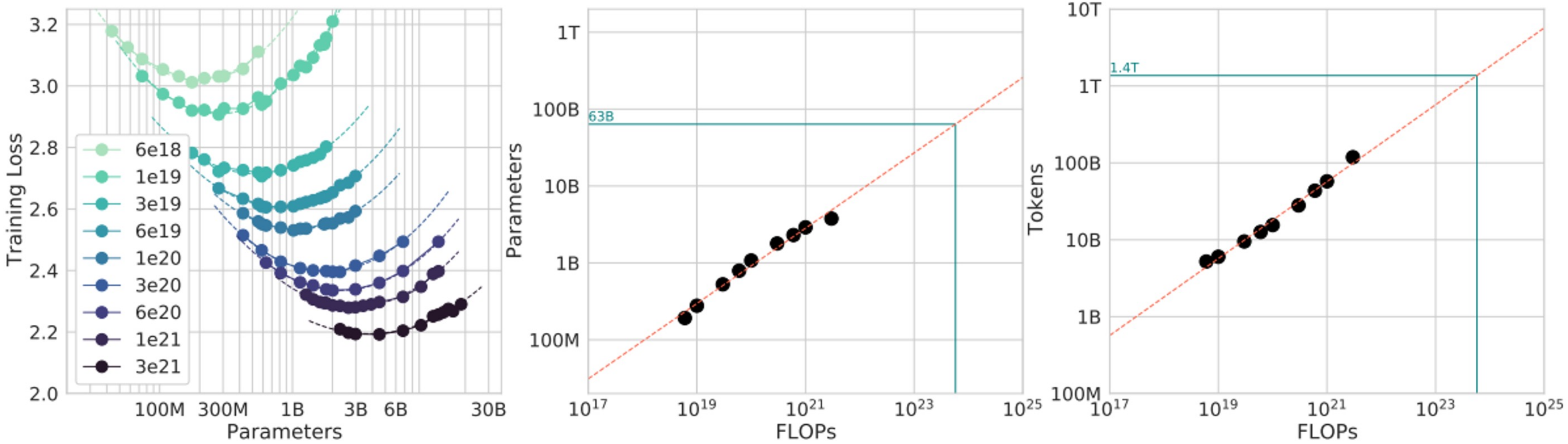
Approach 1: Fix model sizes and vary number of training tokens



Approach 2: IsoFLOP profiles

- Test on a fixed set of 9 different training FLOP counts
- Varied the model size
- A power law can be fitted between FLOPs and loss-optimal model size and number of training tokens
- $N_{opt} \propto C^{a}$ and $D_{opt} \propto C^{b}$ and we find that a = 0.49 and b = 0.51

Approach 2: IsoFLOP profiles

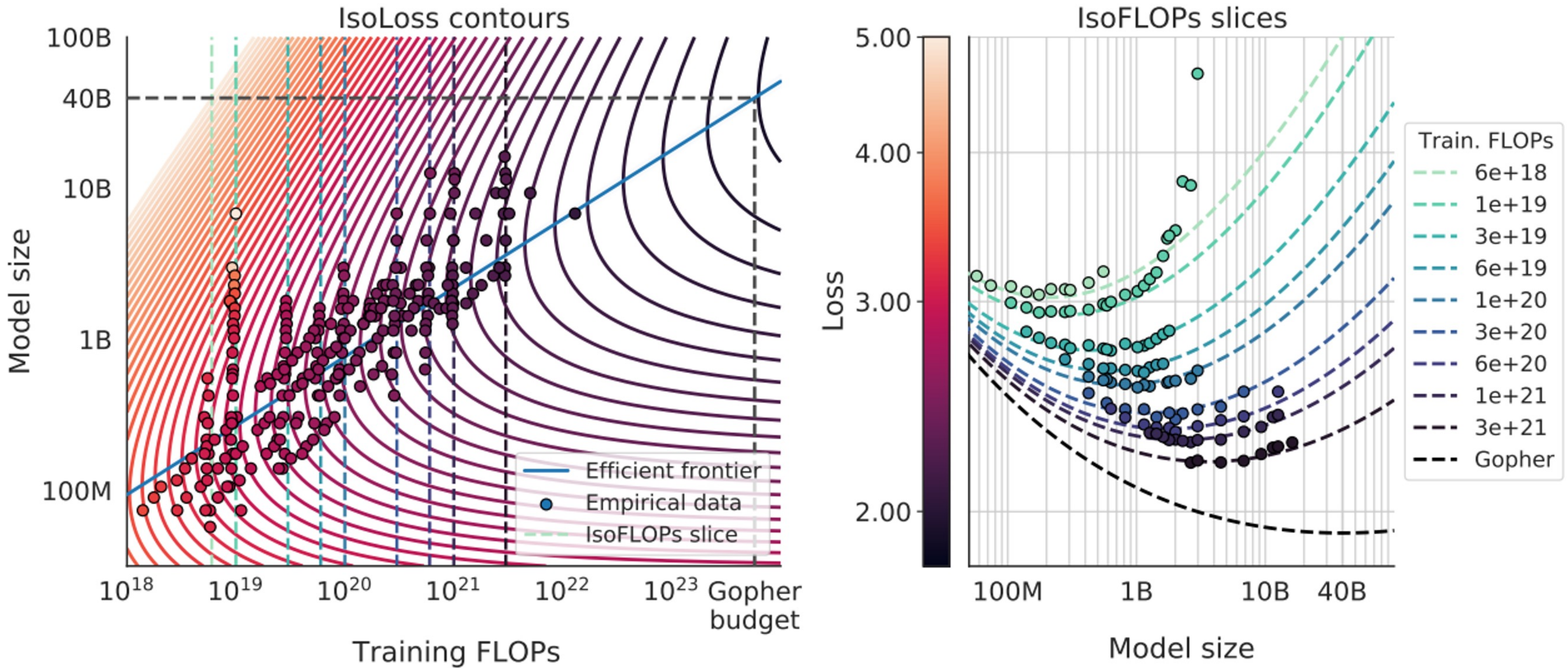


Approach 3: Fitting a parametric loss function

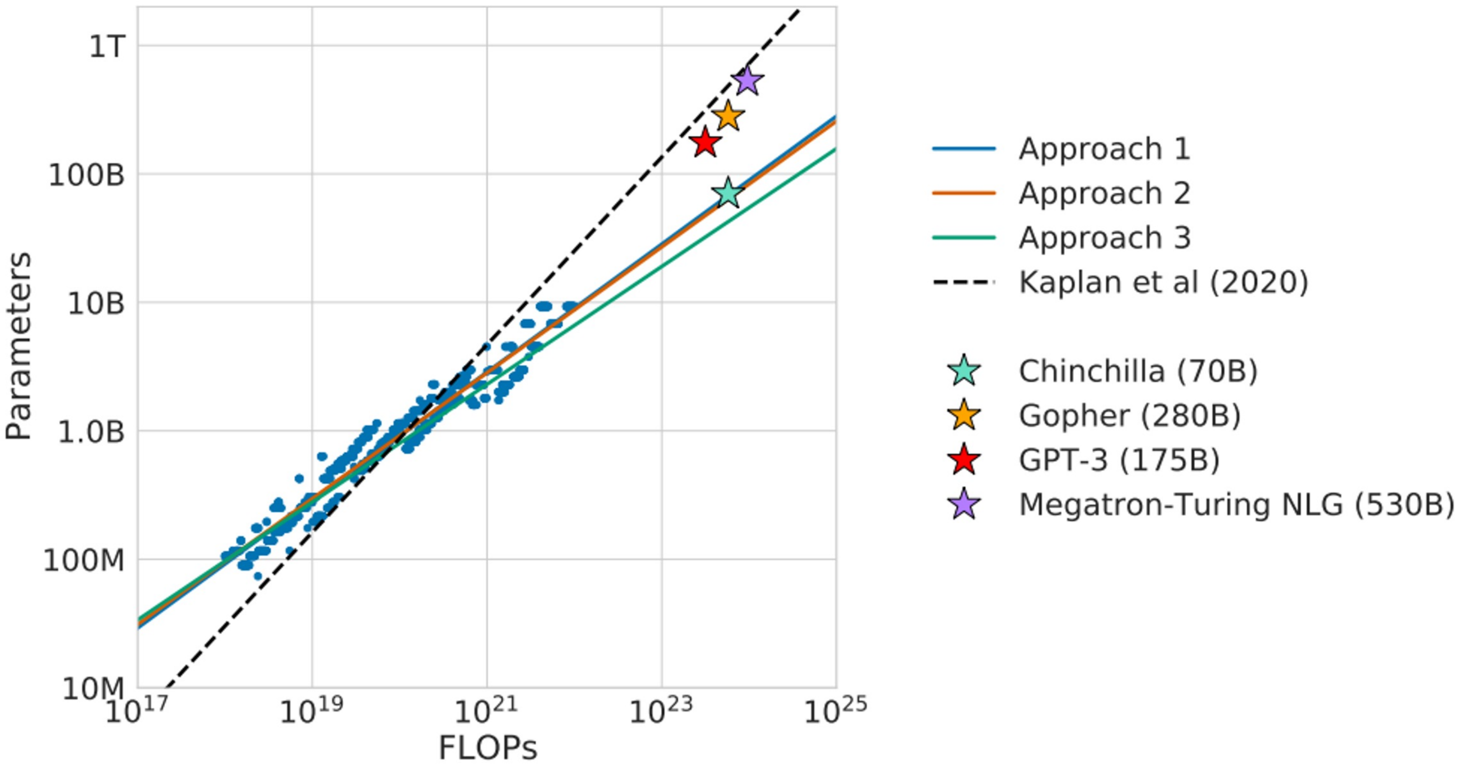
- Model all final losses from experiments in Approach 1 & 2 as a parametric function of model parameter count and the number of seen tokens

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

Approach 3: Fitting a parametric loss function



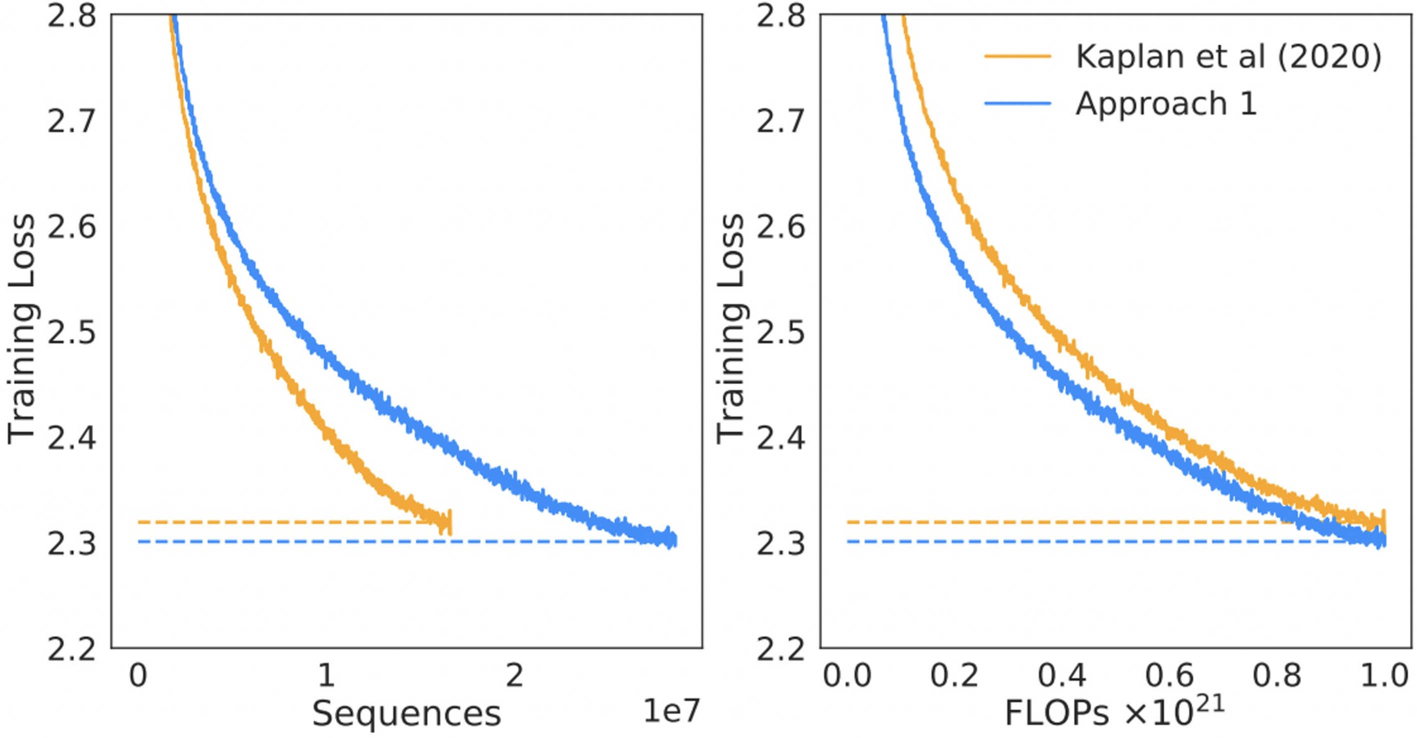
Overlaid Prediction from Different Approaches



Estimated Optimal Training Flops and Training Tokens for various model sizes

Parameters	FLOPs	FLOPs (in *Gopher* unit)	Tokens
400 Million	1.92e+19	1/29,968	8.0 Billion
1 Billion	1.21e+20	1/4,761	20.2 Billion
10 Billion	1.23e+22	1/46	205.1 Billion
67 Billion	5.76e+23	1	1.5 Trillion
175 Billion	3.85e+24	6.7	3.7 Trillion
280 Billion	9.90e+24	17.2	5.9 Trillion
520 Billion	3.43e+25	59.5	11.0 Trillion
1 Trillion	1.27e+26	221.3	21.2 Trillion
10 Trillion	1.30e+28	22515.9	216.2 Trillion

Comparison to Kaplan et al. (2020)



Chinchilla Overview

- Chinchilla is trained on MassiveText, same dataset as Gopher but a slightly different subset distribution to account for increased number of training tokens.

- AdamW optimizer is used rather than Adam optimizer.

- It is trained on slightly modified tokenizer, SentencePiece. The vocabulary is very similar, 94.15% of tokens are the same as those used for training Gopher.

Set of hyperparameters used to train Chinchilla

Model	Layers	Number Heads	Key/Value Size	d_{model}	Max LR	Batch Size
Gopher 280B	80	128	128	16,384	4×10^{-5}	3M $\rightarrow$ 6M
Chinchilla 70B	80	64	128	8,192	1×10^{-4}	1.5M $\rightarrow$ 3M

Evaluation tasks (Chinchilla)

	# 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, ...

Results

Language Modelling

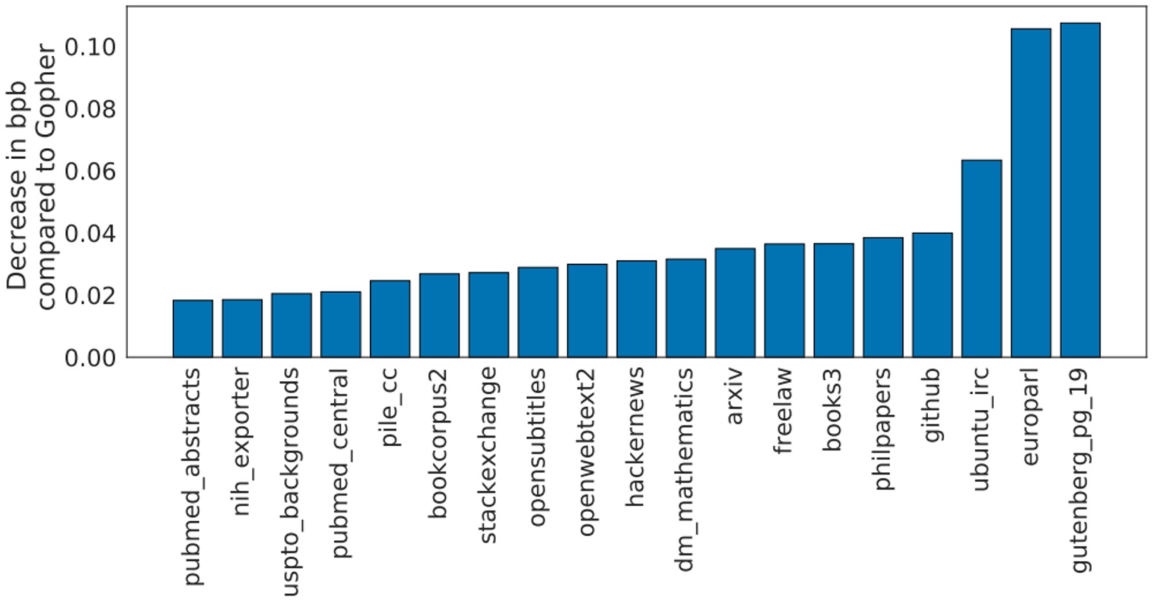


Figure 5 | **Pile Evaluation.** For the different evaluation sets in The Pile (Gao et al., 2020), we show the bits-per-byte (bpb) improvement (decrease) of *Chinchilla* compared to *Gopher*. On all subsets, *Chinchilla* outperforms *Gopher*.

Massive Multitask Language Understanding

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%

Table 6 | **Massive Multitask Language Understanding (MMLU).** We report the average 5-shot accuracy over 57 tasks with model and human accuracy comparisons taken from Hendrycks et al. (2020). We also include the average prediction for state of the art accuracy in June 2022/2023 made by 73 competitive human forecasters in Steinhardt (2021).

Massive Multitask Language Understanding

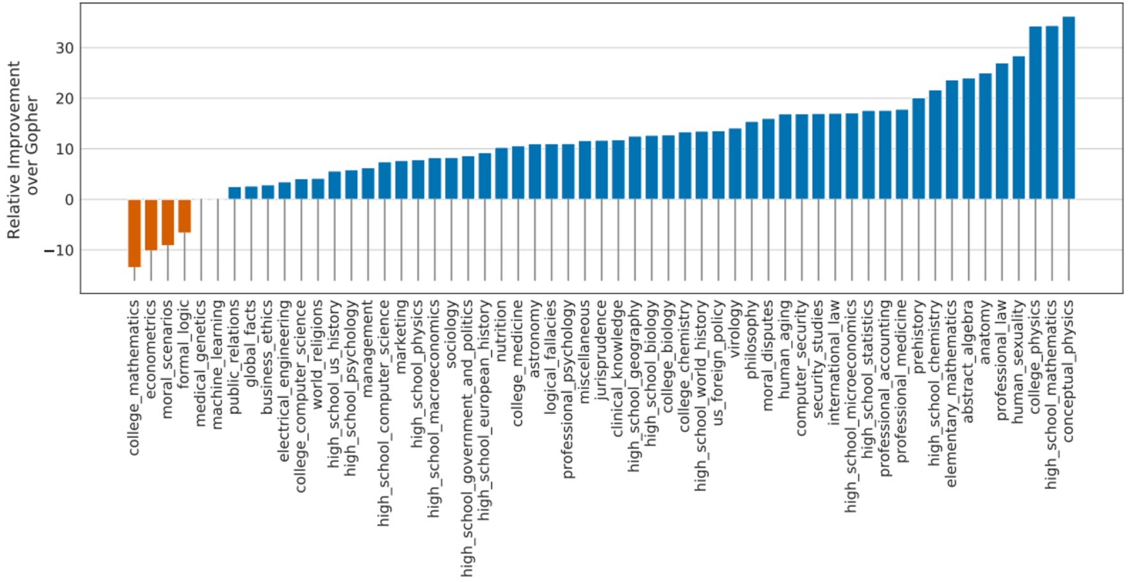


Figure 6 | **MMLU results compared to *Gopher*** We find that *Chinchilla* outperforms *Gopher* by 7.6% on average (see Table 6) in addition to performing better on 51/57 individual tasks, the same on 2/57, and worse on only 4/57 tasks.

Reading Comprehension

	Chinchilla	*Gopher*	GPT-3	MT-NLG 530B
LAMBADA Zero-Shot	**77.4**	74.5	76.2	76.6
RACE-m Few-Shot	**86.8**	75.1	58.1	-
RACE-h Few-Shot	**82.3**	71.6	46.8	47.9

Table 7 | **Reading comprehension.** On RACE-h and RACE-m (Lai et al., 2017), *Chinchilla* considerably improves performance over *Gopher*. Note that GPT-3 and MT-NLG 530B use a different prompt format than we do on RACE-h/m, so results are not comparable to *Gopher* and *Chinchilla*. On LAMBADA (Paperno et al., 2016), *Chinchilla* outperforms both *Gopher* and MT-NLG 530B.

Big Bench

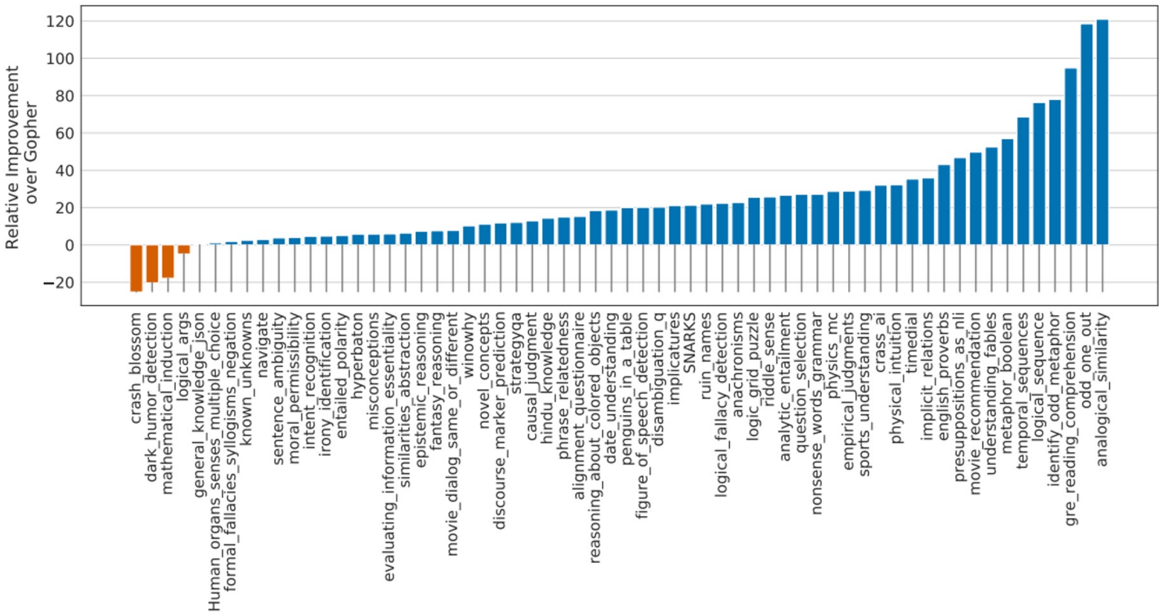


Figure 7 | **BIG-bench results compared to *Gopher*** *Chinchilla* out performs *Gopher* on all but four BIG-bench tasks considered. Full results are in Table A7.

Common Sense Answering

	Chinchilla	*Gopher*	GPT-3	MT-NLG 530B	Supervised SOTA
HellaSWAG	**80.8%**	79.2%	78.9%	80.2%	93.9%
PIQA	81.8%	81.8%	81.0%	**82.0%**	90.1%
Winogrande	**74.9%**	70.1%	70.2%	73.0%	91.3%
SIQA	**51.3%**	50.6%	-	-	83.2%
BoolQ	**83.7%**	79.3%	60.5%	78.2%	91.4%

Table 8 | **Zero-shot comparison on Common Sense benchmarks.** We show a comparison between *Chinchilla*, *Gopher*, and MT-NLG 530B on various Common Sense benchmarks. We see that *Chinchilla* matches or outperforms *Gopher* and GPT-3 on all tasks. On all but one *Chinchilla* outperforms the much larger MT-NLG 530B model.

Closed Book Question Answering

	Method	*Chinchilla*	*Gopher*	GPT-3	SOTA (open book)
Natural Questions (dev)	0-shot	16.6%	10.1%	14.6%	54.4%
	5-shot	31.5%	24.5%	-	
	64-shot	35.5%	28.2%	29.9%	
TriviaQA (unfiltered, test)	0-shot	67.0%	52.8%	64.3 %	-
	5-shot	73.2%	63.6%	-	
	64-shot	72.3%	61.3%	71.2%	
TriviaQA (filtered, dev)	0-shot	55.4%	43.5%	-	72.5%
	5-shot	64.1%	57.0%	-	
	64-shot	64.6%	57.2%	-	

Table 9 | **Closed-book question answering.** For Natural Questions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017), *Chinchilla* outperforms *Gopher* in all cases. On Natural Questions, *Chinchilla* outperforms GPT-3. On TriviaQA we show results on two different evaluation sets to allow for comparison to GPT-3 and to open book SOTA (FiD + Distillation (Izacard and Grave, 2020)).

Gender Bias and Toxicity

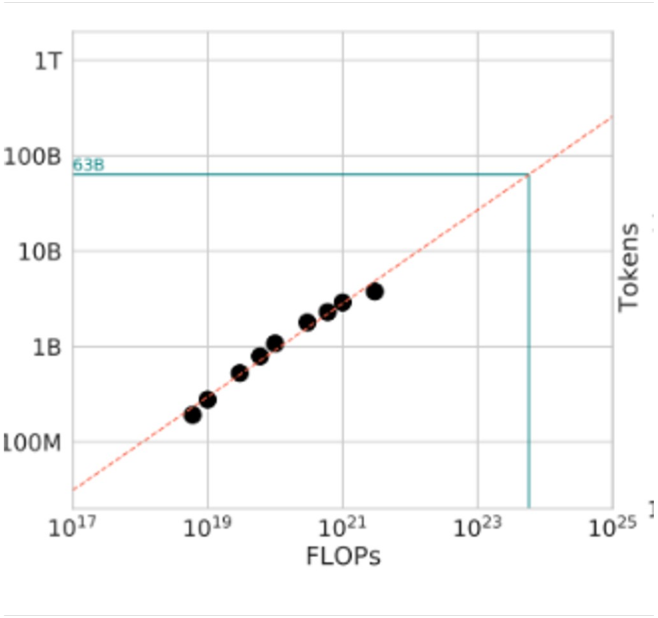
	Chinchilla	*Gopher*
All	78.3%	71.4%
Male	71.2%	68.0%
Female	79.6%	71.3%
Neutral	84.2%	75.0%

	Chinchilla	*Gopher*
Male *gotcha*	62.5%	59.2%
Male *not gotcha*	80.0%	76.7%
Female *gotcha*	76.7%	66.7%
Female *not gotcha*	82.5%	75.8%

Table 10 | **Winogender results. Left:** *Chinchilla* consistently resolves pronouns better than *Gopher*. **Right:** *Chinchilla* performs better on examples which contradict gender stereotypes (*gotcha* examples). However, difference in performance across groups suggests *Chinchilla* exhibits bias.

Limitations

- Is measuring flops the way to go? Doesn’t it have dependency on hardware?
- Data leakage might be a serious issue.
- Is it fair to derive a linear trend from a handful of samples?



Conclusion & Future Direction

- Provided guideline can be a good starting point for setting up model dimensions given a computational budget.
- If we agree to this paper’s findings, then We can conclude that most existing language models are oversized.
- Research has to be done to better understand the optional model size and number of required tokens.
- Measures should be taken to eradicate data leakage.
- Chinchilla does suffer from gender bias and toxicity. Research should be done to find how performance of language models and toxicity interact and how they can be avoided.

Thank You

Any Questions?