Training Compute-Optimal Large Language Models

Проблема

В индустрии прослеживается тенденция на большие LLM модели > 500 B параметров.

Power Law: чем больше модель, тем лучше её перфоманс.

Проблема

В индустрии прослеживается тенденция на большие LLM модели > 500 B параметров.

Power Law: чем больше модель, тем лучше её перфоманс.

Но и с ростом количества параметров - растет стоимость обучения таких моделей.

Контекст - размеры state-of-art LLM моделей

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

Постановка задачи

N - размер модели

D - количество токенов

C - бюджет вычислений

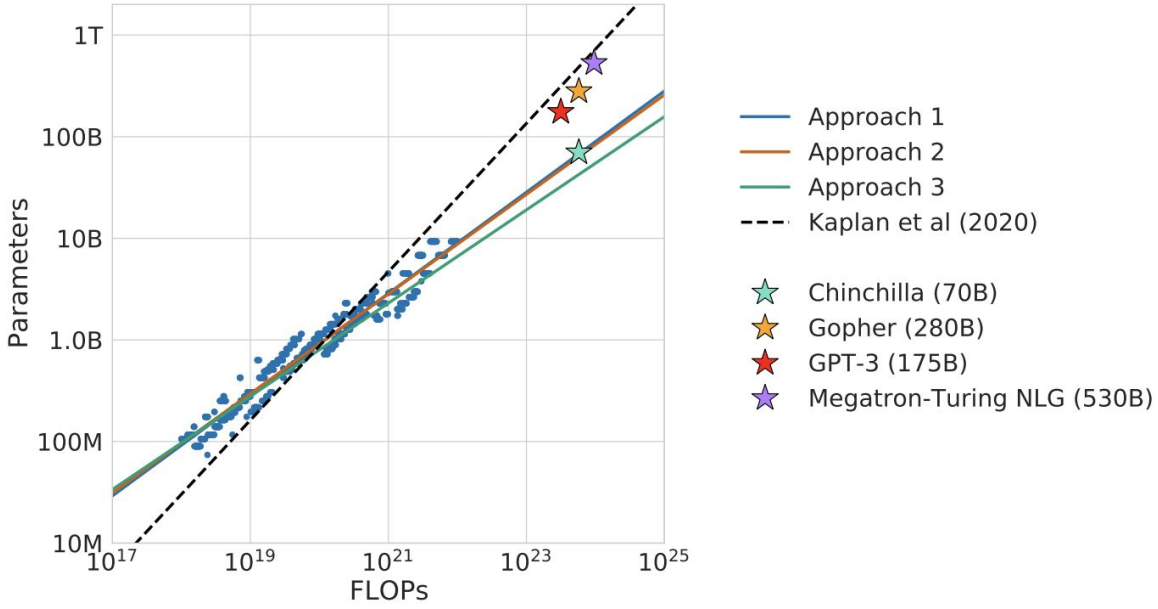
FLOPs(N, D) - детерминированная функция,

оценка вычислительной стоимости обучения такой модели

L(N, D) - loss функция

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

Опровержение статьи Kaplan



Правила скейлинга не подтвердились!

Chinchilla, которая меньше Gopher в 4 раза, при равном бюджете вычислений показывает результаты ничем не хуже.

Как так?

У Chinchilla больше объем обучающей выборки, и за счет малого размера - дообучали дольше.

Approach-1

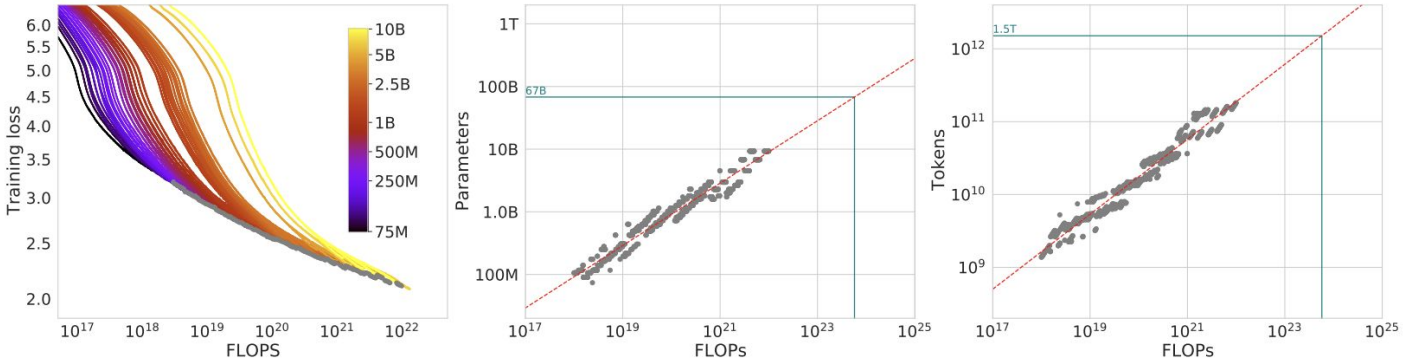


Figure 2 | **Training curve envelope.** On the **left** we show all of our different runs. We launched a range of model sizes going from 70M to 10B, each for four different cosine cycle lengths. From these curves, we extracted the envelope of minimal loss per FLOP, and we used these points to estimate the optimal model size (**center**) for a given compute budget and the optimal number of training tokens (**right**). In green, we show projections of optimal model size and training token count based on the number of FLOPs used to train *Gopher* (5.76×10^{23}).

Approach-2

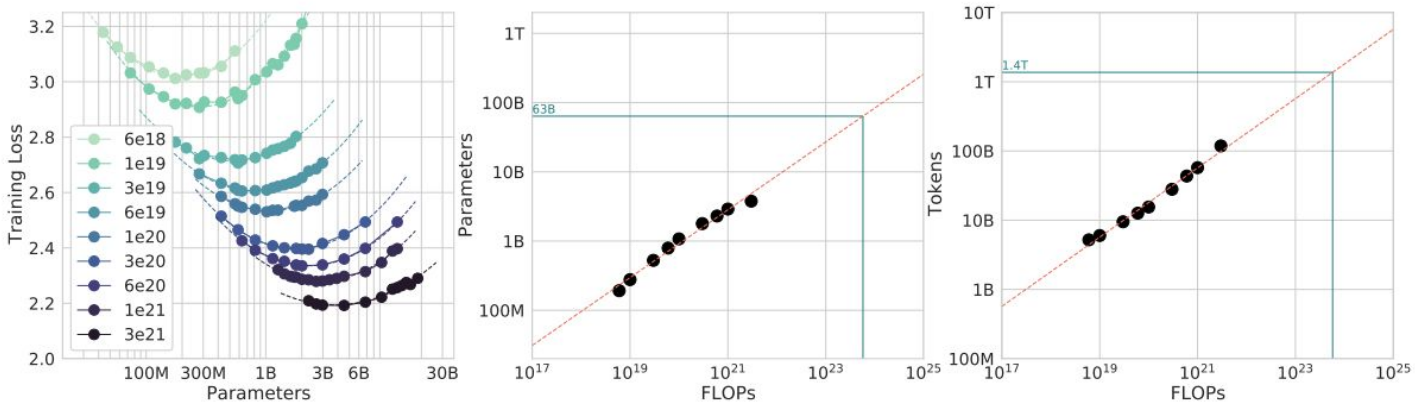


Figure 3 | **IsoFLOP curves.** For various model sizes, we choose the number of training tokens such that the final FLOPs is a constant. The cosine cycle length is set to match the target FLOP count. We find a clear valley in loss, meaning that for a given FLOP budget there is an optimal model to train (**left**). Using the location of these valleys, we project optimal model size and number of tokens for larger models (**center** and **right**). In green, we show the estimated number of parameters and tokens for an *optimal* model trained with the compute budget of *Gopher*.

Approach-3

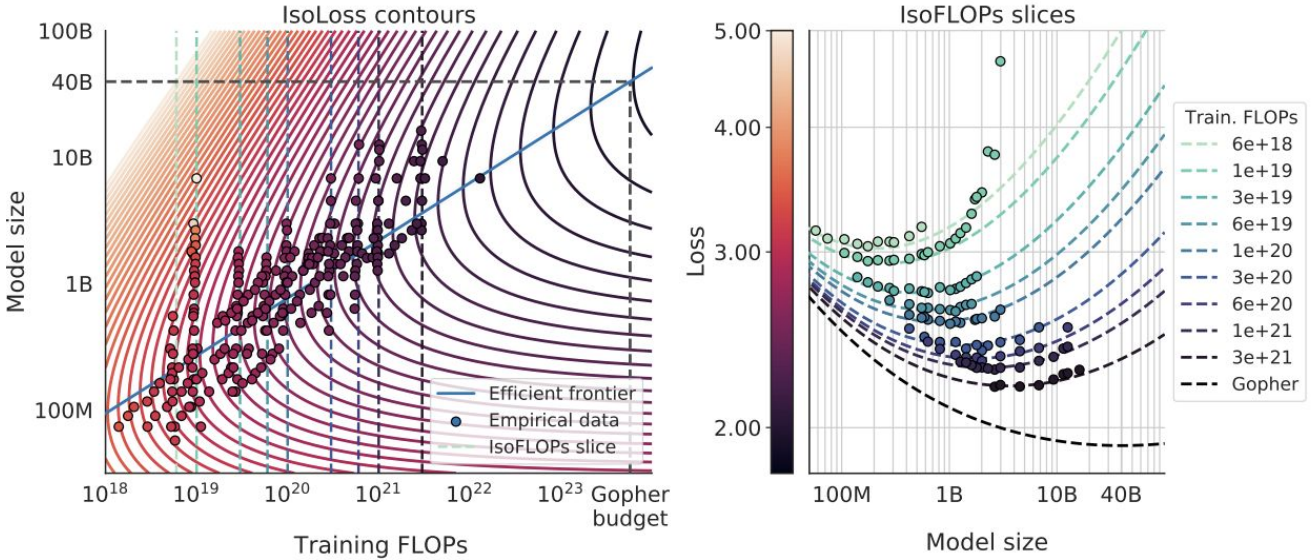
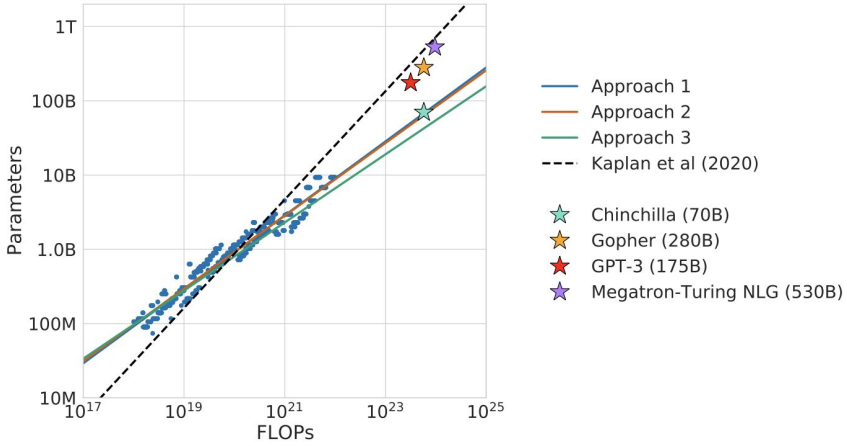


Figure 4 | **Parametric fit.** We fit a parametric modelling of the loss $\hat{L}(N, D)$ and display contour (**left**) and isoFLOP slices (**right**). For each isoFLOP slice, we include a corresponding dashed line in the left plot. In the left plot, we show the efficient frontier in blue, which is a line in log-log space. Specifically, the curve goes through each iso-loss contour at the point with the fewest FLOPs. We project the optimal model size given the *Gopher* FLOP budget to be 40B parameters.

Итоги



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

С ростом размера модели так же важно увеличивать размер обучающей выборки.

DeepMind N:D = 1:1

Kaplan N:D = 7:3