Scaling Laws

Training large language model is costly

- We want to train extremely large language models, but still, we can not train models that are arbitrarily large.
- We need to predict:
  - How large the language model need to be?
  - What is the training compute we need to use?
- This is where the scaling law is used, where we predict the behavior of large language models by doing experiment on much smaller ones.

We will talk about two scaling laws

- Capacity Scaling Law (by Meta and MBZUAI)
  - How large the language model needs to be.
- Computation Scaling Law (by DeepMind)
  - How long we need to train the language model.

Capacity Scaling Law

- Language Model Capacity:
  - We focus on how many “factual knowledge” can a language model memorize.

- (Factual) Knowledge is the foundation of human intelligence.
- Harvard University is a private Ivy League research university in Cambridge, Massachusetts. Founded in 1636 as Harvard College and named for its first benefactor, the Puritan clergyman John Harvard, it is the oldest institution of higher learning in the United States. Its influence, wealth, and rankings have made it one of the most prestigious universities in the world. (Wikipedia)

In this lecture, we consider one piece of knowledge defined as a (Name, Attribute, Value) tuple.

(Harvard, Found Year, 1636)
(Harvard, Location, Cambridge MA)

How many piece of such knowledge can a language model of size M memorize?

This is the scaling law of model capacity.

Framework

We consider a (semi) synthetic dataset of biographies

Example: Anya Briar Forger was born on October 2, 1996. She spent her early years in Princeton, NJ. She received mentorship and guidance from faculty members at Massachusetts Institute of Technology. She completed her education with a focus on Communications. She had a professional role at Meta Platforms. She was employed in Menlo Park, CA.

6 Attributes: Birthday, Birth City, University, Field, Company, Company City.

We generate synthetic biography for N people, the value of the attributes are randomly assigned. Each attribute has ~ 200 different values (first, middle, and last names each having 1000 values).

- Each sentence is either randomly chosen from 200 prescribed templates (fully synthetic data, BioS), or generated by Llama (semi-synthetic data, BioR).
- We see the results are the same for both cases. So we mainly focus on BioS.

Scaling Law

- How many piece of biography knowledge can a language model of size M store?

Piece of Biography knowledge

- Roughly speaking, we say that the language model stores one piece of biography knowledge, if after training, the model can answer questions like:
  - What is the birth date of Anya Briar Forger?
  - What is the birth city of Anya Briar Forger?
  - Where did Anya Briar Forger study?

Information-theoretic lower-bound

- On the biography dataset, we can compute an information-theoretic bound on
  - How many bits is necessary to store those biography information
- The bound looks like

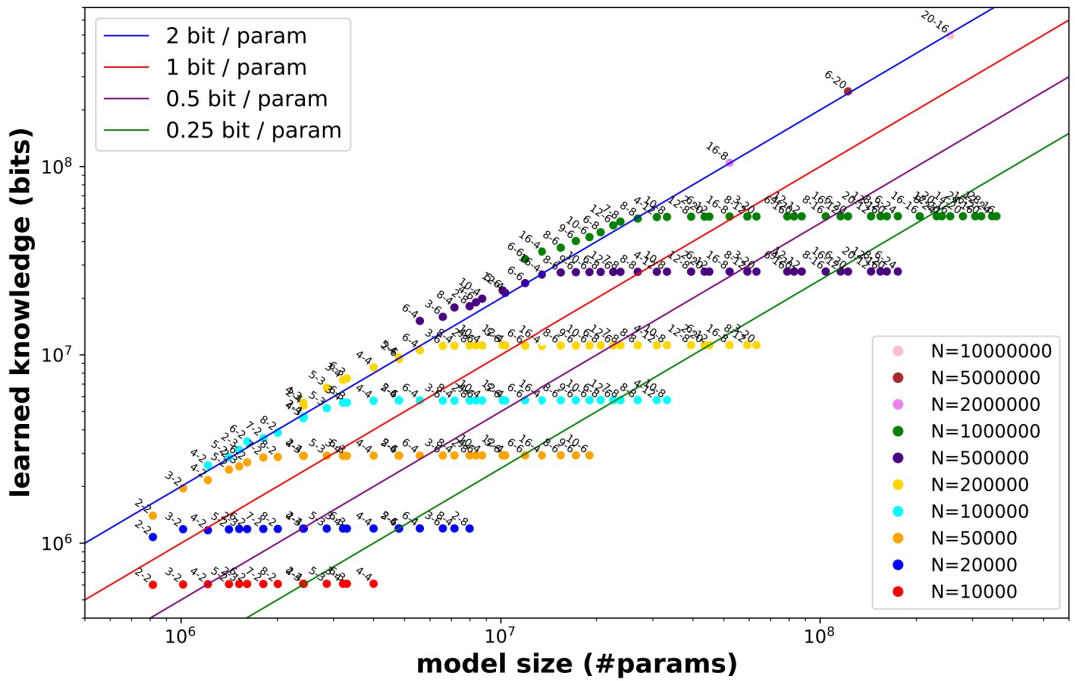
$$\log_2 |\mathcal{W}| \geq \mathop{\mathbb{E}}_{\mathcal{Z}} \Big[N \log_2 \frac{N_0 - N}{e^{\mathbf{loss}_{name}(\mathcal{Z})}} + NK \log_2 \frac{D^C}{e^{\mathbf{loss}_{value}(\mathcal{Z})}} + KD \log_2 \frac{T^L - D}{De^{(1+o(1))\mathbf{loss}_{value1}(\mathcal{Z})}} - o(KD) \Big]$$

Information-theoretic lower-bound

- Suppose Information theoretically, we need R bits to store N biography data up to certain accuracy.
  - Then how many parameters W do we need in the transformer to store the data?
  - (First obvious observation): W is a linear function of R.
  - What is the slop?

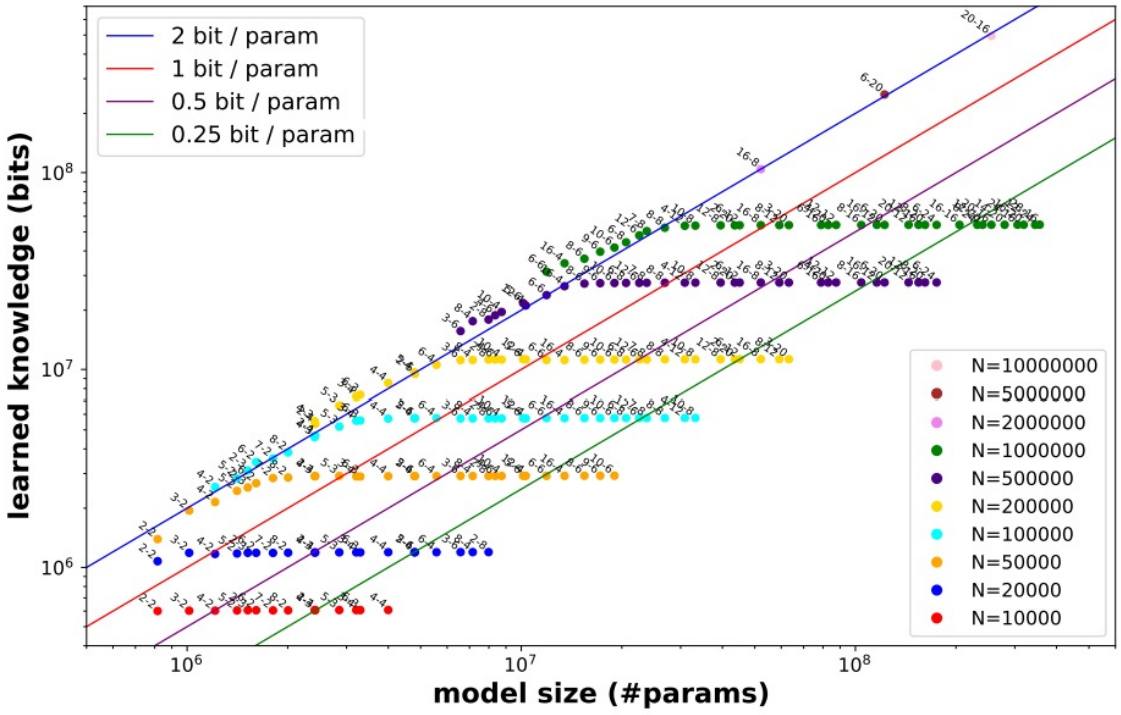
2bit/parameter

- Transformer models, trained with FP16 using Adam, can store and only store
- 2bit/parameter (no matter how long you train the transformer model).



2bit/parameter even with FP8/Int8

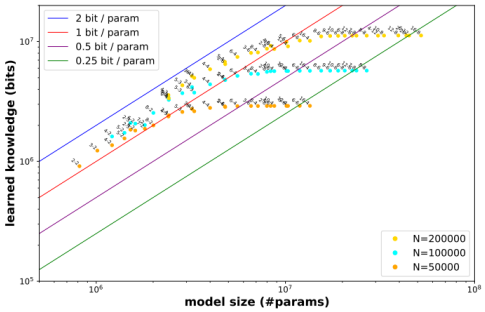
If we train with FP16, and then quantalize the model to Int8, we still see 2bit/parameter



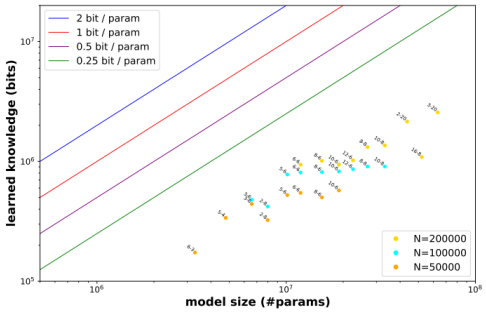
(a) 1000 pass Figure 1(a) quantized to 8bit

How long do we need to train the transformer to reach maximum capacity?

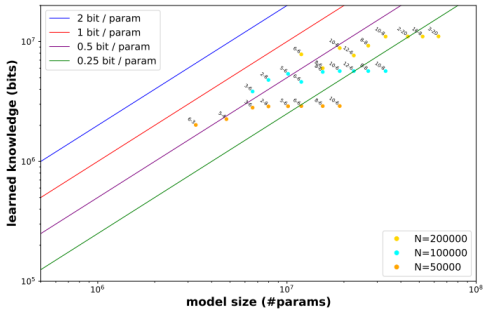
100 pass over the data is roughly enough, but we need to train on clean data.



(a) no junk, 100 pass



(b) 7/8 junk, 100 pass on main data



(c) 7/8 junk, 800 pass on main data

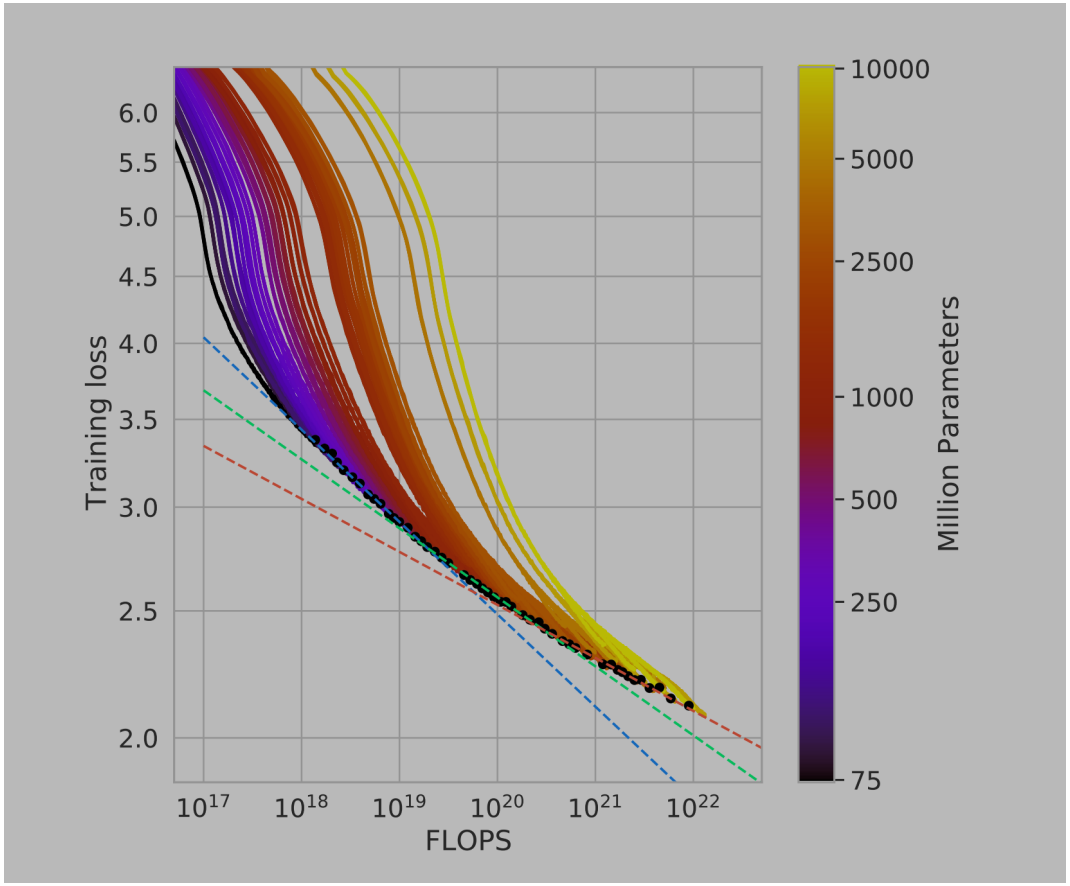
Implication:

- If we want to train language models optimally:
  - When we double the size of the language model.
  - We need to more than double the total number of training tokens.

- If a 7B model requires 2T training tokens to be trained optimally
- Then a 70B model should require more than 20T training tokens!!!

How do we train models more efficiently

- Computation Scaling Law (By Deepmind)
- Important observation:
  - If a language model of size X requires K passes over the data to memorize all the knowledges in it.
  - Then a language model of size p * X (for p > 1) requires less than K / p passes over the data to memorize all the knowledge in it.
- Larger model trains faster to memorize the same amount of knowledge.


Training loss
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
10^{17}
10^{18}
10^{19}
10^{20}
10^{21}
10^{22}
FLOPS
10000
5000
2500
1000
500
250
75
Million Parameters

Informal Observation

- If you want to memorize k bits of knowledge
- Model of k/2 parameters require 1000 passes over the data
- Model of k parameters requires 100 passes over the data
- Model of 10k parameters only require ~1 pass over the data