CHALLENGES IN BRINGING LARGE LANGUAGE MODELS TO THE MARKET

Teratec 2022 — IA & HPC dans l'Industrie

Julien Launay

Extreme-Scale Project Lead | LightOn

{firstname}@lighton.ai

Large Language Models (LLMs) are eating machine learning

LLMs provide a universal text-based interface to tackle any tasks:

Text instructions → **LLM** → **Answer (text output)**

What companies were mentioned in this article?
Write an essay based on these notes:
Write an Instagram ad for...

e.g. GPT-3

Key aspects of LLMs:

🧠 They are **generalists**, able to tackle broad tasks just from instructions.

📈 Their **capabilities** increase as you **scale-up** in size/compute.

🧪 One of the **main business & research interest** in machine learning.
from Google, DeepMind, Microsoft, etc. + large start-ups such as OpenAI and Cohere.

🚀 Just the **beginning**: proper **prompting** + addition of other **modalities**.

But Large Language Models are hard!

Today is not about what/why, but about how, from experience with:

BigScience

176B multilingual model

Muse

muse.lighton.ai

Chonk' me up Scotty

For the next generation of LLMs, we will need to scale...

use case

varied applications

datasets

quality at scale

compute

engineering challenges

modeling

accelerate scaling

As many use cases as they are users...

You can use LLMs for countless downstream applications...

From straightforward text completion...

Yourtext.Guru

writing assistant

...or text classification...

Pipplet

Part of ETS GLOBAL

language assessments

...to web development...

Iteration X

code changes from issues

and more!

Unexpected use of LLMs: socratic models

From an egocentric video, understand what happened during the day.

Zeng et al.,2022.

Egocentric video

Rich input

Perception stack

Information

"Was I happy today?"

Query on input

Prompting

Dialogue

LLM

"Yes. You spent time with your kids watching a show you like."

Answer

Unexpected use of LLMs: socratic models

How do we prepare for all these use cases when building an LLM?

❌ NLP datasets don't capture all these tasks well... "real-world datasets" (e.g. RAFT)

😵‍💫 Variety in task types & best architectures... aggregated benchmarks (e.g. EAI, Tk-Few)

🤖 Alignment with human intentions specialised models (e.g. instructGPT)

🦾 Specialisation of the model?

- zero/few-shot use
- finetuning
- parameter efficient finetuning
- multitask finetuning
- different tradeoffs/practices

Model quality is all about data quality

Training data matters a lot!

(more than most modeling choices?)

Aggregated performance on EAI harness

Model	Parameters	Pretraining tokens			
		Dataset	112B	250B	300B
OpenAI — Curie	6.7B				49.28
OpenAI — Babbage	1.3B				**45.30**
EleutherAI — GPT-Neo	1.3B	The Pile			42.94
	13B	OSCAR			47.09
Ours	1.3B	The Pile	**42.79**	43.12	43.46
	1.3B	C4	42.77		
	1.3B	OSCAR	41.72		

Le Scao et al.,2022.

Same architecture, different data:

45.30% — OpenAI-Babbage(1.3B)

43.46% — Ours-1.3B@The Pile

Model quality is all about data quality

Training data matters a lot!

(more than most modeling choices?)

Aggregated performance on EAI harness

Model	Parameters	Pretraining tokens			
		Dataset	112B	250B	300B
OpenAI — Curie	6.7B				49.28
OpenAI — Babbage	1.3B				**45.30**
EleutherAI — GPT-Neo	1.3B	The Pile			42.94
Ours	13B	OSCAR			47.09
	1.3B	The Pile	**42.79**	43.12	43.46
	1.3B	C4	42.77		
	1.3B	OSCAR	41.72		

Le Scao et al.,2022.

Scale can't compensate for bad data:

49.28%

OpenAI-Curie(6.7B)

47.09%

Ours-13B@OSCAR

We are gonna need a bigger dataset!

Bad news: we need a lot more data than expected...

Previously... Kaplan et al., 2020

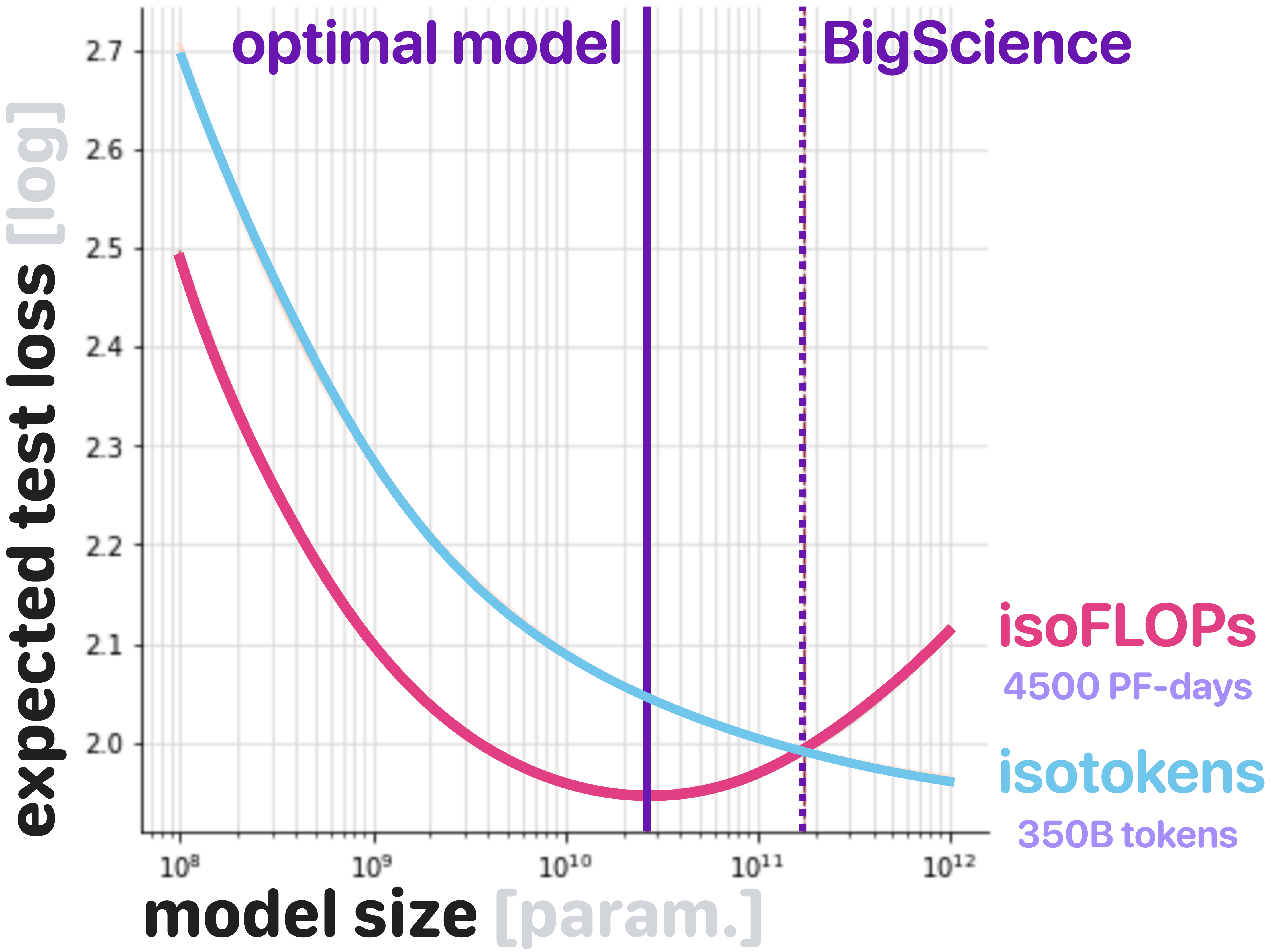
176B parameters ⟶ 300B tokens

Now... Hoffmann et al., 2020

isoFLOPs 50B parameters ⟶ 1000B tokens

isoparams 176B parameters ⟶ 3700B tokens

~1 year of CC English



Will we be data-bound instead of compute-bound?

Fantastic training data and where to find it

🔬 What even is high-quality data? technical filtering deduplication, lack of artefacts, etc.

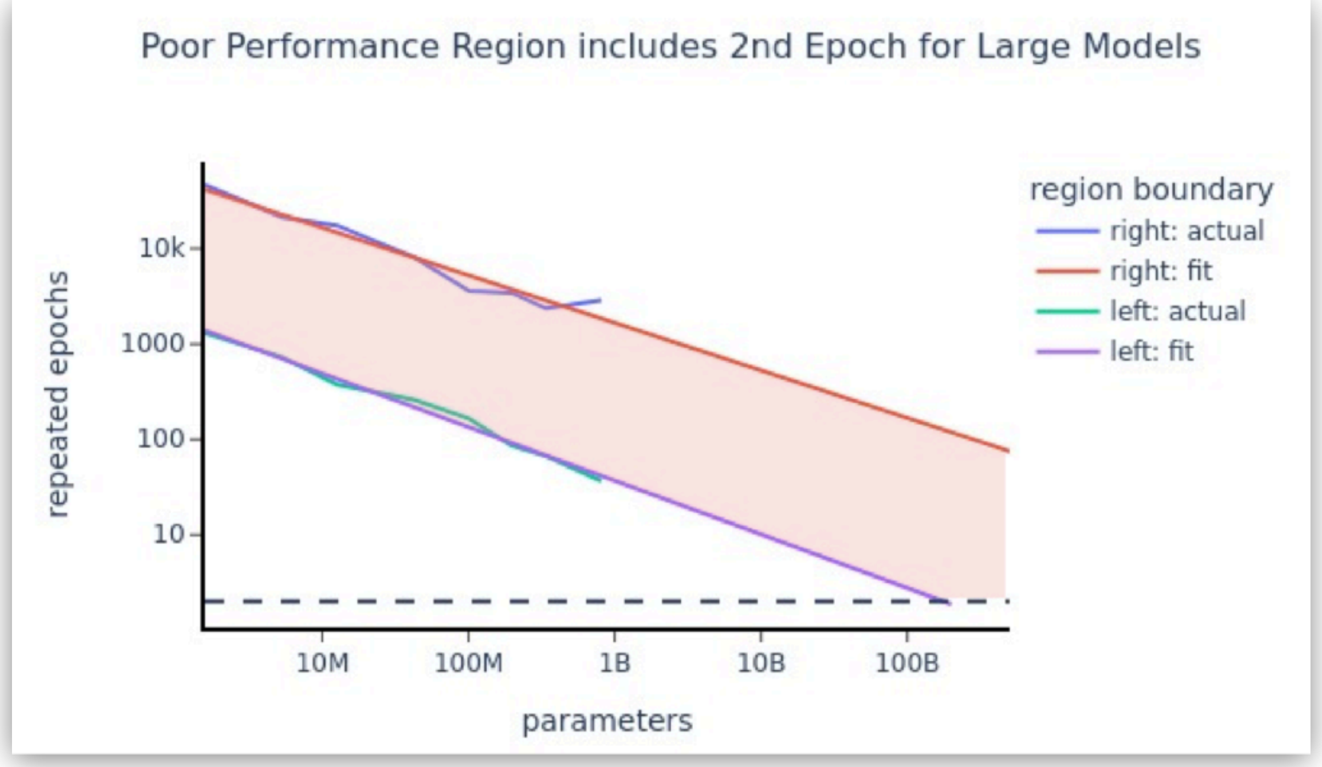
curation diverse, cross-domain, etc.

"social media conversations"

Total dataset size = 780 billion tokens	
Data source	Proportion of data
Social media conversations (multilingual)	50%
Filtered webpages (multilingual)	27%
Books (English)	13%
GitHub (code)	5%
Wikipedia (multilingual)	4%
News (English)	1%

Chowdhery et al.,2022.

double descent for duplication?



Hernandez et al.,2022.

🔮 Currently, dataset construction is more akin to magic... Need principled methods!

⚠️ Emergence of data moats which could stand in the way of research.

Fantastic training data and where to find it

We need this in >100 languages!

Required minimum data

Model size	Minimum tokens
1.5B	20B
6.7B	134B
20B	400B
100B	2,000B
500B	10,000B

Hoffmann et al.,2022.

Data available in one year of CommonCrawl

Ranking	Language	High quality tokens	Medium quality tokens
1st	English	2T	6T
7th	French	260B	880B
15th	Indonesian	50B	145B
30th	Hindi	8B	16B

1GB ~4B tokens, high quality is top 20%, medium 20-40%, one dump per 3-4 months.

Iterating at 88 batches an hour

🤗 Training time on an NVIDIA SuperPod (160 A100):

Model size	Good for...	Tokens	A100h required	Training time
1.5B	Simple classification, basic generation	200B	4300	~1 day
6.7B	Most generation & classification use cases	200B	22000	<1 week
20B	Zero/few-shot complex tasks	400B	134000	<1 month

We are doing Big Science, and this comes with challenges...

🚀 LLMs are a true big science and require significant engineering efforts...

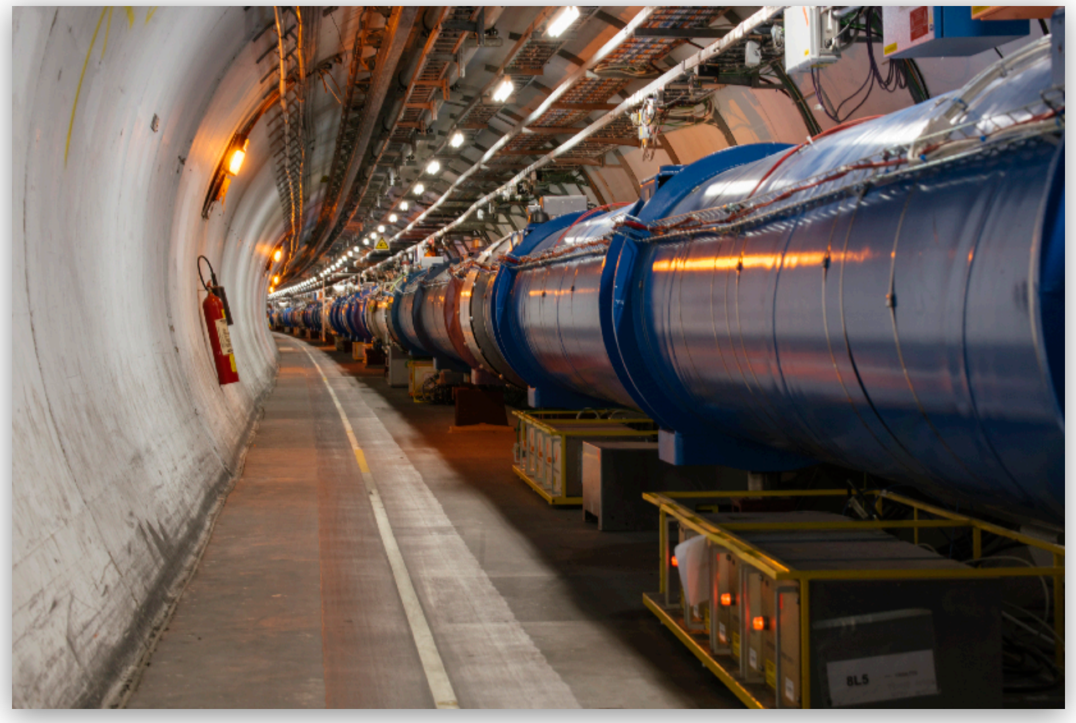
state-of-the-art HPC challenges

📒 Principled approaches are very much needed:

- tested and validated frameworks
- expert HPC/software engineering knowledge
- performance tuning is magic currently

e.g. tile/wave quantization, distributed hyperparameters, etc.

BLOOM: >100 configurations tested!



(let's avoid this)

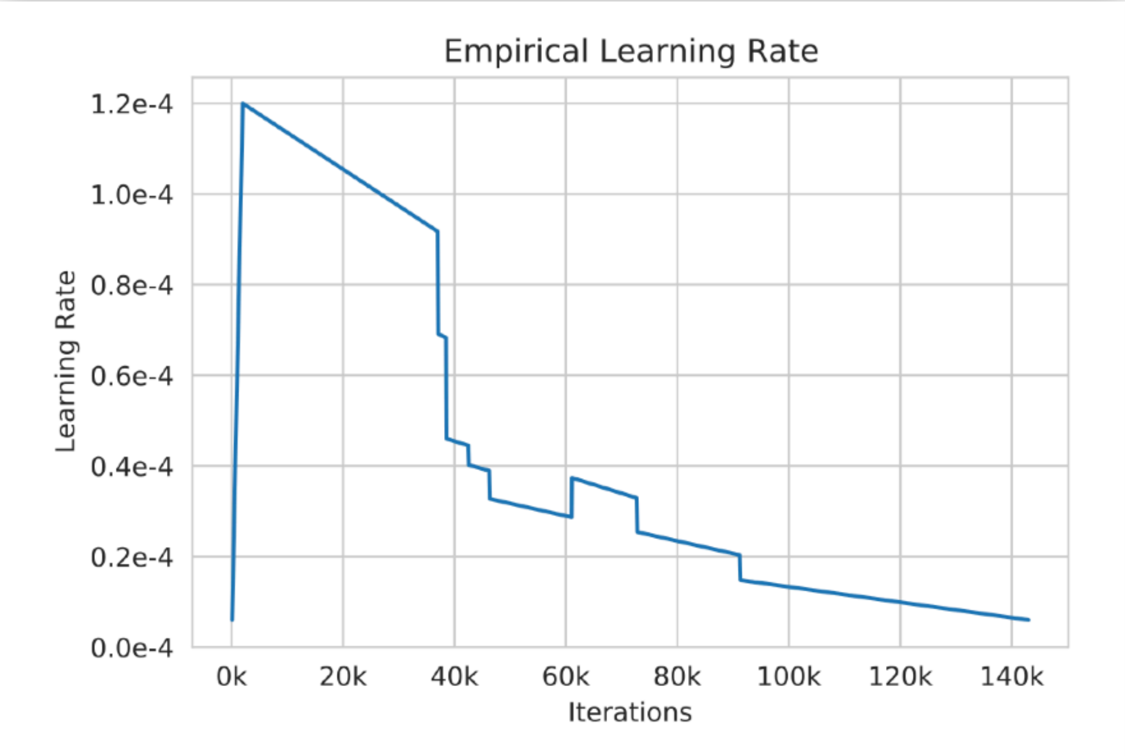
Case-study of how hard it can get: Meta's OPT


OPT: Open Pre-trained Transformer Language Models

OPT: Open Pre-Trained Transformer Language Models

Zhang et al., 2022

😅 Meta's open "reproduction" of GPT-3 was... a challenging experience!



manually tuned learning rate

hundreds of restarts, spikes, etc.

But why?

FP16

BF16

template: Karpathy, 2020

He who controls the chips controls the LLMs


arXiv:2009.06489v2 [cs.CY] 21 Sep 2020

The Hardware Lottery

Sara Hooker

Google Research, Brain Team
shooker@google.com

Abstract

Hardware, systems and algorithms research communities have historically had different incentive structures and fluctuating motivation to engage with each other explicitly. This historical treatment is odd given that hardware and software have frequently determined which research ideas succeed (and fail). This essay introduces the term hardware lottery to describe when a research idea wins because it is suited to the available software and hardware and not because the idea is superior to alternative research directions. Examples from early computer science history illustrate how hardware lotteries can delay research progress by casting successful ideas as failures. These lessons are particularly salient given the advent of domain specialized hardware which make it increasingly costly to stray off of the beaten path of research ideas. This essay posits that the gains from progress in computing are likely to become even more uneven, with certain research directions moving into the fast-lane while progress on others is further obstructed.

1 Introduction

History tells us that scientific progress is imperfect. Intellectual traditions and available tooling can prejudice scientists against certain ideas and towards others (Kuhn, 1962). This adds noise to the marketplace of ideas, and often means there is inertia in recognizing promising directions of research. In the field of artificial intelligence research, this essay posits that it is our tooling which has played a disproportionate role in deciding what ideas succeed (and which fail).

What follows is part position paper and part historical review. This essay introduces the term *hardware lottery* to describe when a research idea wins because it is compatible with available software and hardware and not because the idea is superior to alternative research directions. We argue that choices about software and hardware have often played a decisive role in deciding the winners and losers in early computer science history.

These lessons are particularly salient as we move into a new era of closer collaboration between hardware, software and machine learning research communities. After decades of treating hardware, software and algorithms as separate choices, the catalysts for closer collaboration include changing hardware economics (Hennessy, 2019), a "bigger is better" race in the size of deep learning architectures (Amodei et al., 2018; Thompson et al., 2020b) and the dizzying requirements of deploying machine learning to edge devices (Warden & Situnayake, 2019).

Closer collaboration has centered on a wave of new generation hardware that is "domain specific" to optimize for commercial use cases of deep neural networks (Jouppi et al., 2017; Gupta & Tan, 2019; ARM, 2020; Lee & Wang, 2018). While domain specialization creates important efficiency gains for mainstream research focused on deep neural networks, it arguably makes it more even more costly to stray off of the beaten path of research ideas. An increasingly fragmented hardware landscape means that the gains from progress in computing will be increasingly uneven. While deep neural networks have clear commercial use cases, there are early warning signs that the path to the next

1

Hardware progress is secretly shaping machine learning

The Hardware Lottery

Sara Hooker, 2020

GPU

data/model/pipeline/sequence parallelism

diversity in HPC platforms

network topology, etc.

→

it's not enough to have the GPUs,
you need the **platform** around it!

Can better modeling & more efficient pretraining change the playing field?

💨 We can gain in efficiency...

current approaches, ~50% GPU FLOPs usage

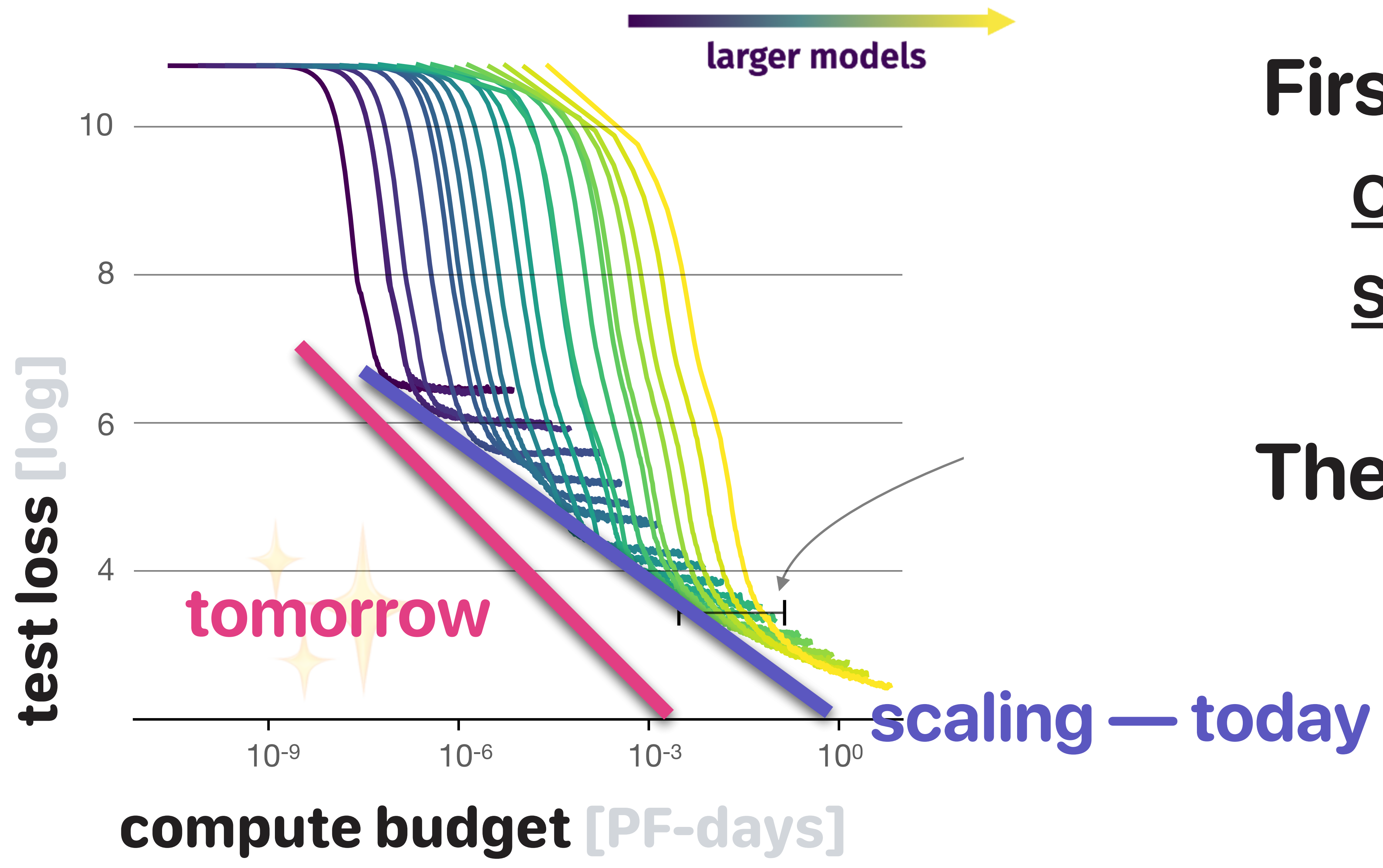
reduced numerical precision: down to int8

see Transformer engine in H100

reduce number of computations

efficient attention, etc.

But can we also fundamentally change scaling behaviour?



First, optimise pretraining:

- Curriculum learning, grow sequence length — Li et al., 2021
- Staged training, progressively grow model — Shen et al., 2022

Then, can we get better scaling?

Can better **modeling** & more efficient **pretraining** change the playing field?

Significant compute resources will remain key!