Houston Machine Learning - LLM Reading Group

SCALING LAWS

Stanford CS324

Yan Xu

Motivating problem: hyperparameter costs

Hyperparameter tuning is a huge cost!

How can we solve this?

1. Guess and pray
2. Exhaustive search
3. Have simple rules that find optimal hyperparams

Consumption	**CO_2e (lbs)**
Air travel, 1 passenger, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000
Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

Table 1: Estimated CO_2 emissions from training common NLP models, compared to familiar consumption.[1]

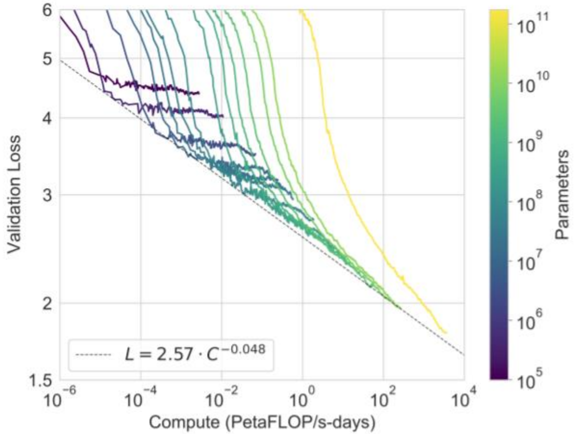
Strubell+ 2019

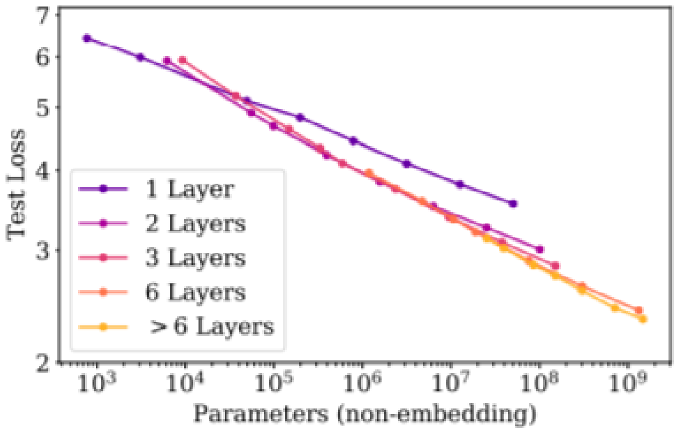
Teaser: simple, predictive ‘laws’ for behaviors of LMs

What you’ll learn today:

scaling laws which are simple, predictive rules for model performance

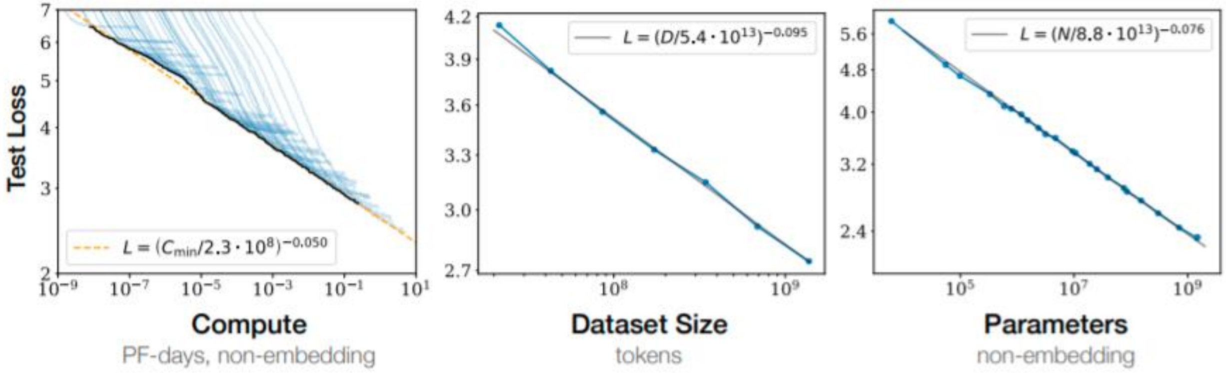
Old and unpleasant: tune hyperparameters on big models
New and exciting: tune on small models, extrapolate to large ones



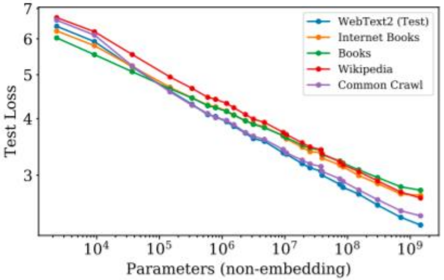


Scaling laws: surprisingly clean and robust

These scaling laws hold on *many* different kind of phenomena!



They even hold in non-standard settings (when train ≠ test)



All you want to know about scaling laws (and more)

Organization: simple to complex

1. Data vs performance

“Are there simple rules that determine how data affects performance?”

2. Hyper-parameters vs performance

“Are optimal hyperparmeters the same across different data/models?”

3. Forecasting with scaling laws

“Does benchmark performance follow predictable trends”?

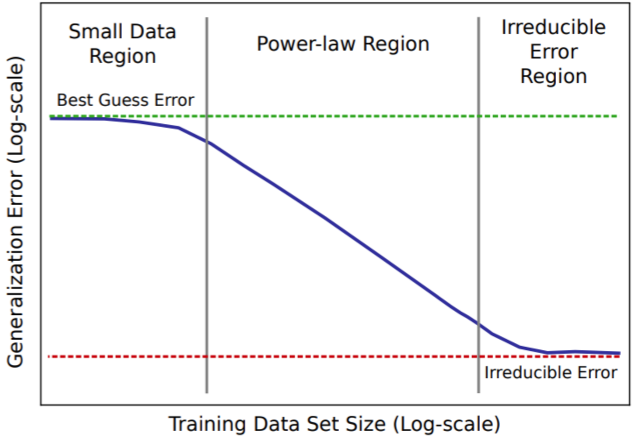
Data vs performance

What's a data scaling law?

Data scaling laws : simple formula that maps dataset size (n) to error

What do we expect out of scaling laws?

Monotonic, logistic-like curves

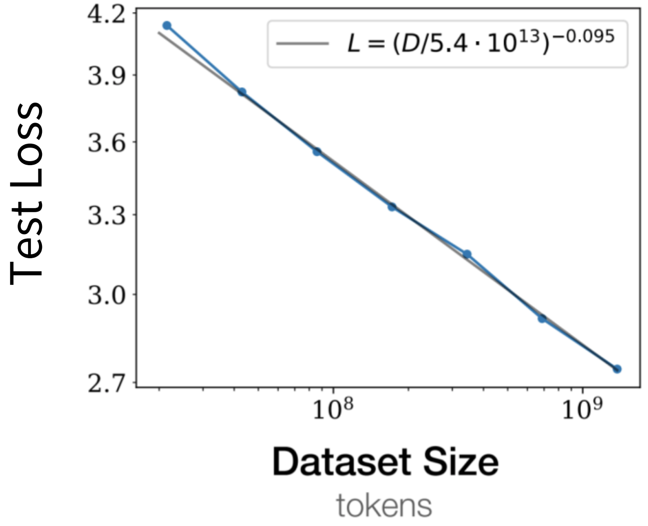


[Hestness+ 2017]

Data scaling laws for language models

First, an empirical observation

Loss and dataset size is linear on a log-log plot

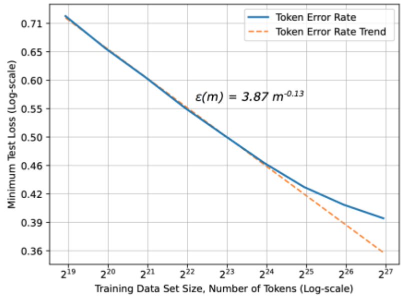


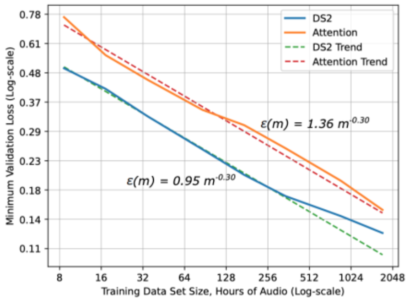
“Scale-free” or
“Power law”

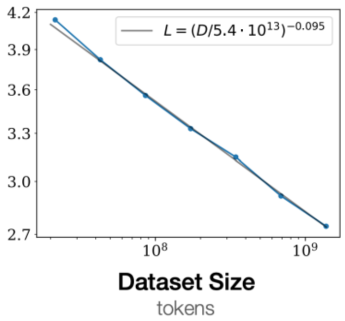
(For language modeling, from Kaplan+ 2020)

Scaling laws: past works and other areas

Scaling laws hold in many domains







Machine translation

Speech

Language modeling

Hestness et al 2017.

Kaplan et al 2020.

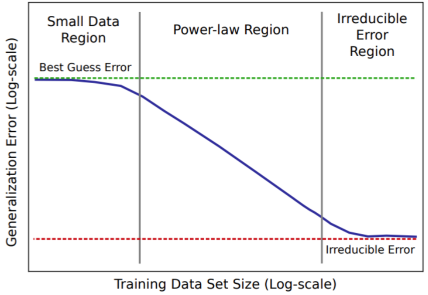
Data scaling has been known for a while

Kolachina+ 2012 for machine translation, Hestness+ 2017 for neural

Conceptual foundations of data scaling laws.

Q: Why do scaling laws show up?

We know error should be monotone



But why is it a power law / linear in log-log?

A: Estimation error naturally decays polynomially.

But this answer may take a moment to understand. Let's work through an example.

Example: If our task is to estimate the mean of a dataset, what's the scaling law?

Toy example: mean estimation

Input: $x_1 \ \ldots x_n \ \sim N(\mu, \sigma^2)$

Task: estimate the average as $\hat{\mu} = \frac{\sum_i x_i}{n}$

What's the error? By standard arguments..

$$\mathrm{E}[(\hat{\mu} - \mu)^2] = \frac{\sigma^2}{n}$$

This is a scaling law!!

$$\log(Error) = -\log n + 2 \log \sigma$$

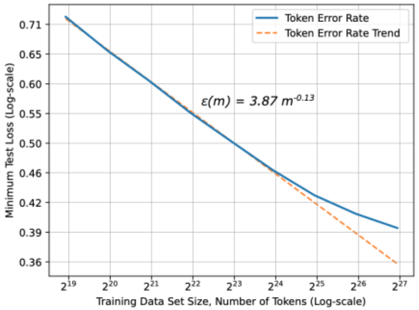
More generally, any polynomial rate $1/n^{\alpha}$ is a scaling law

Scaling law exponents: an intriguing mystery

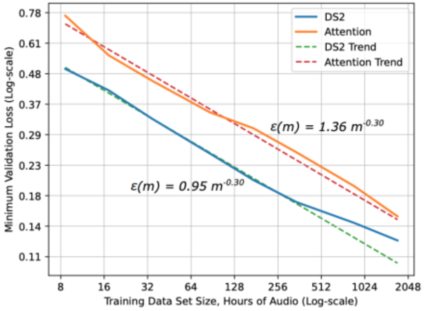
Fact: Similar arguments show most ‘classical’ models (regression, etc) have $\frac{1}{n}$ scaling

This means we should see $y = -x + C$

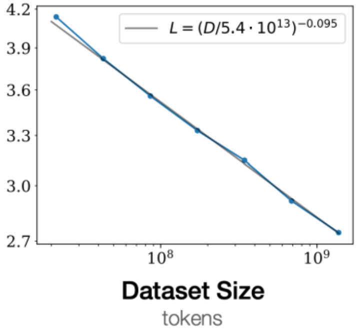
What do we find in neural scaling laws?



Machine translation



Speech



Language modeling

Very different from predictions.. Why might this be?

Intrinsic dimensionality theory of data scaling laws

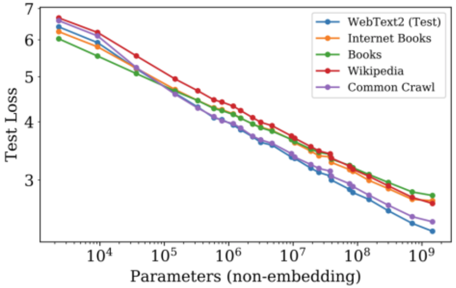
1. Scaling laws arise due to polynomial rates of learning $\frac{1}{n^{\alpha}}$
2. The slope α is closely connected to the *intrinsic dimensionality* of the data.

Other advanced data scaling law: distribution shift

Data scaling thus far: how does dataset size relate to performance?

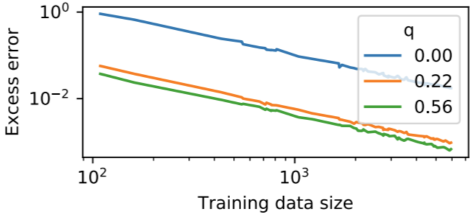
Related question: how does dataset *composition* affect performance

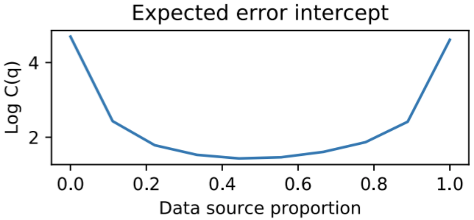
A: Data composition affects the offset, not the slope.



[Kaplan+ 2021]

These 'distribution shift' scaling laws can tell us about the importance of collecting diverse data!





[Hashimoto 2021]

Recap: data scaling laws

Remarkably linear relationship between log-data size and log-error

Holds across domains and models

Theory understanding: similar to generalization bounds: mean estimation example

Applications: data collection, fairness.

Scaling laws for model engineering

Now for what I promised at the start: **model scaling!**

Our motivation: how can we efficiently design huge LMs?

- LSTMs vs Transformers
- Adam vs SGD

How should we allocate our limited resources?

- Train models longer vs train bigger models?
- Collect more data vs get more GPUs?

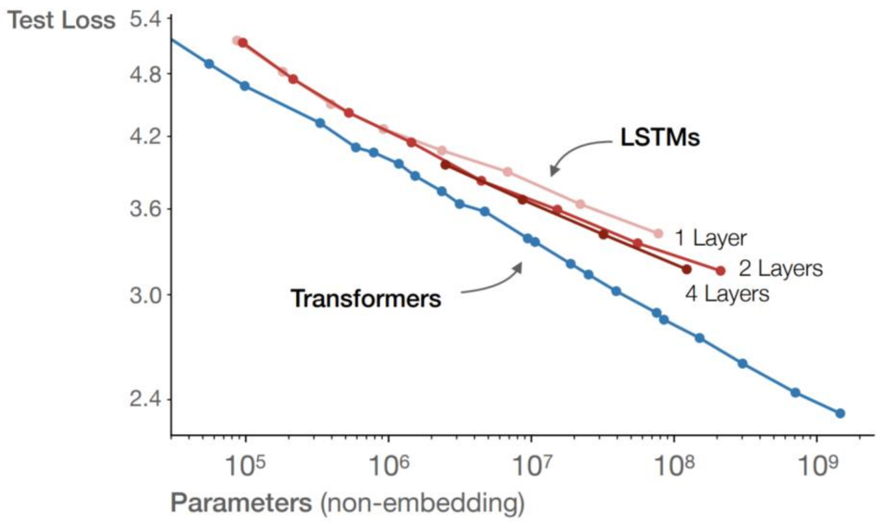
Scaling laws provide a simple procedure to answer these.

Cross-model: transformers vs LSTMs

Q: Are transformers better than LSTMs?

Brute force way: spend tens of millions to train a LSTM GPT-3

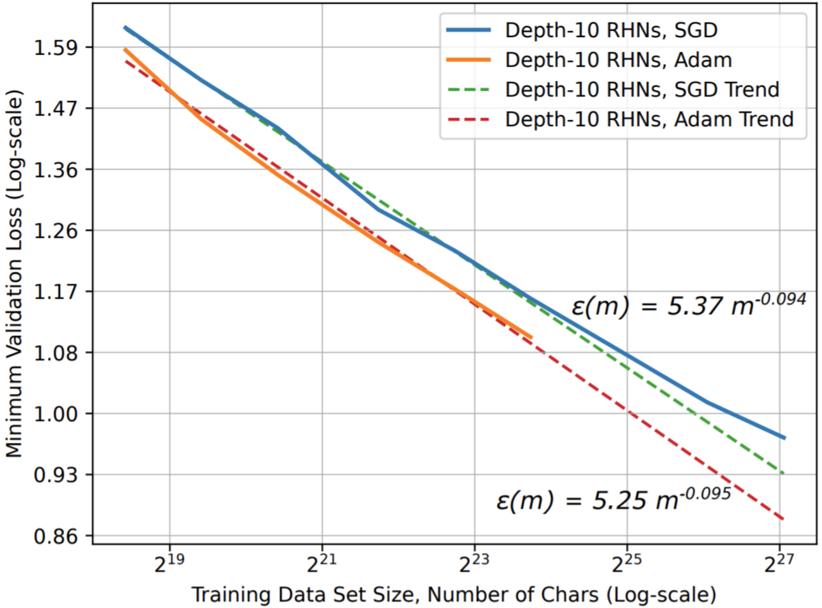
Scaling law way:



[Kaplan+ 2021]

Optimizer choice

What about ADAM vs SGD?

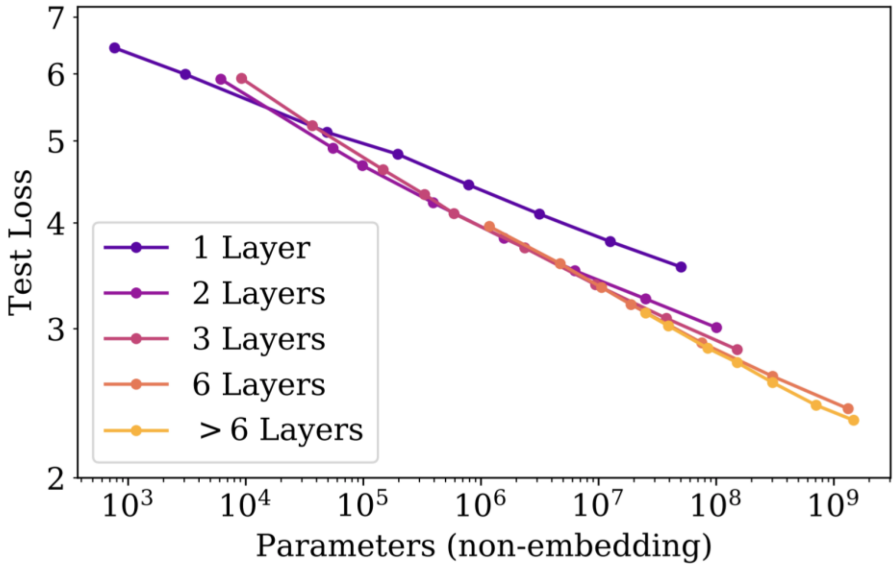


[Hestness+ 2017]

(Note, this is in 2017, so pre-transformers. RHN is recurrent highway nets)

Number of layers

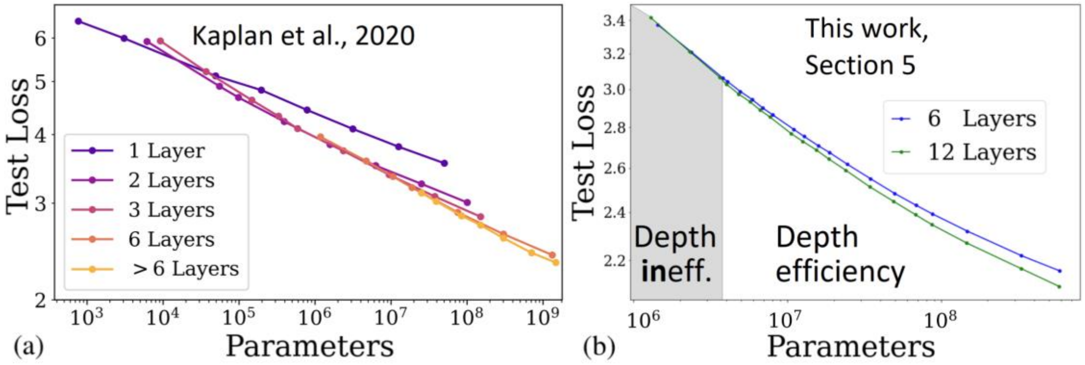
Does depth or width make a huge difference?



- 1 vs 2 layers makes a huge difference.
- More layers have diminishing returns below 10^7 params

Side note – scaling laws can sometimes lead us astray

These scaling laws are already used in the design of LMs



Model	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	n_{vocab}
GPT-3 6.7B	6.7B	32	4096	32	128	50K
J1-Large	7.5B	32	4096	32	128	256K
GPT-3 175B	175B	96	12288	96	128	50K
J1-Jumbo	178B	76	13824	96	144	256K

Table 1: Comparing the architecture of our Jurassic-1 models to their GPT-3 counterparts.

For a parameter budget of 175B (not including embedding matrix), the optimal depth should be around 80 layers, far from the 96 layers used by GPT-3 175B.

[Levine+ 2021]

Some surprising takeaways

The effect of hyperparameters on big LMs can be predicted *before* training!

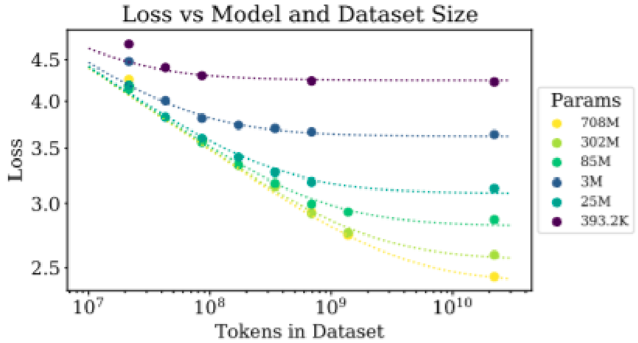
- Optimizer choice
- Model depth
- Architecture choice

The scaling law based design procedure.

1. Train a few smaller models
2. Establish a scaling law (e.g. ADAM vs SGD scaling law)
3. Select optimal hyperparam based on the scaling law prediction.

Model size data joint scaling

Q: Do we need more data or bigger models?



Clearly, lots of data is wasted on small models

Joint data-model scaling laws describe how the two relate

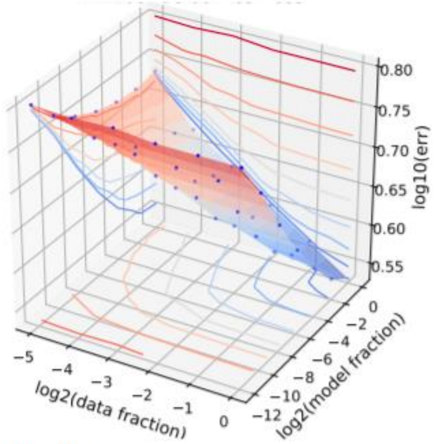
From Rosenfeld+ 2020,

$$Error = n^{-\alpha} + m^{-\beta} + C$$

From Kaplan+ 2021

$$Error = [m^{-\alpha} + n^{-1}]^{\beta}$$

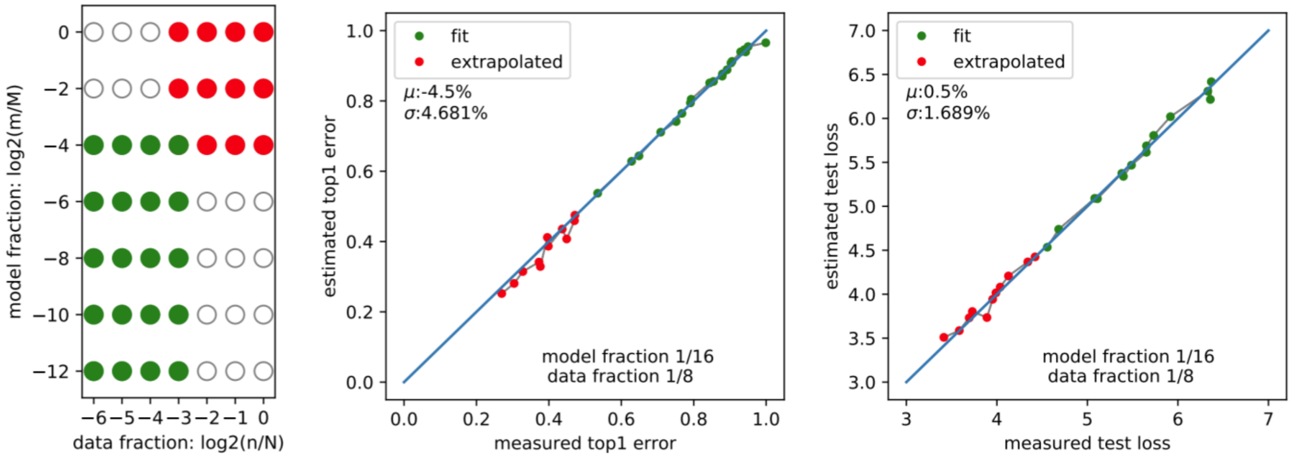
Provides surprisingly good fits to model-data joint error.



(a) Wiki103 error (cross entropy) landscape.

Model-data joint scaling is accurate

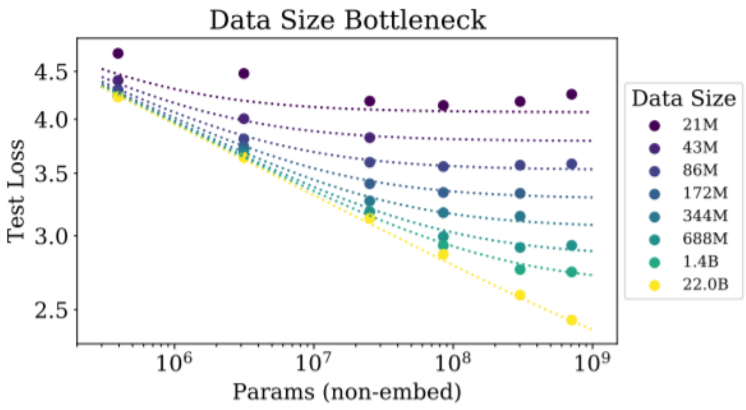
From Rosenfeld – fit scaling exponents on small data, small models. Predict rest.



(a) Illustration. (b) Extrapolation on ImageNet (c) Extrapolation on WikiText-103.

Trading off data size and model size: optimize $n^{-\alpha}+m^{-\beta}+C$ with your costs.

Do we have enough data to feed our models?



From Kaplan:

Fitted training laws suggest 22B token WebText can fit 10^9 parameters.

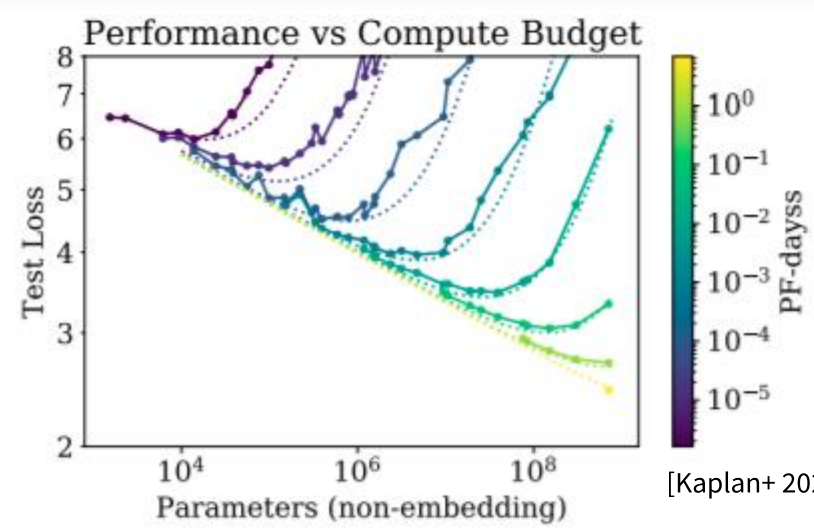
Model size should scale as $O(m^{0.74})$

Compute tradeoffs.

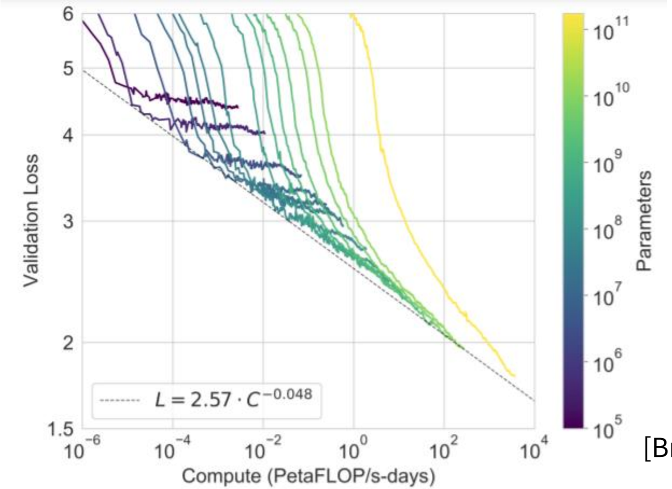
Q: what about other resources? Compute vs performance?

For a fixed compute budget…

Big model that's undertrained vs small model that's well trained?



[Kaplan+ 2021]



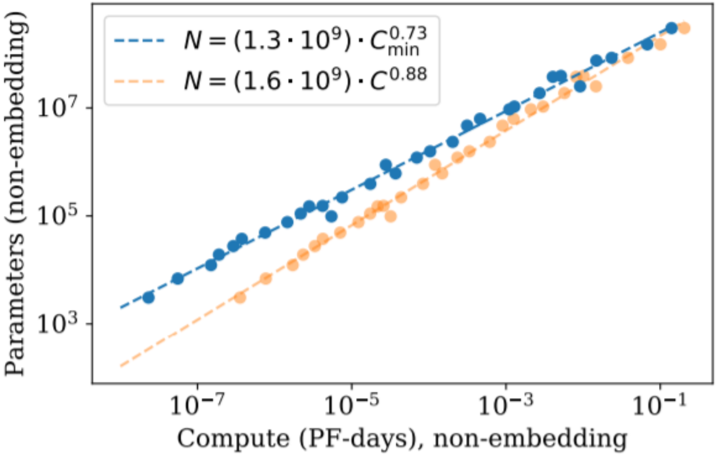
[Brown+ 2020]

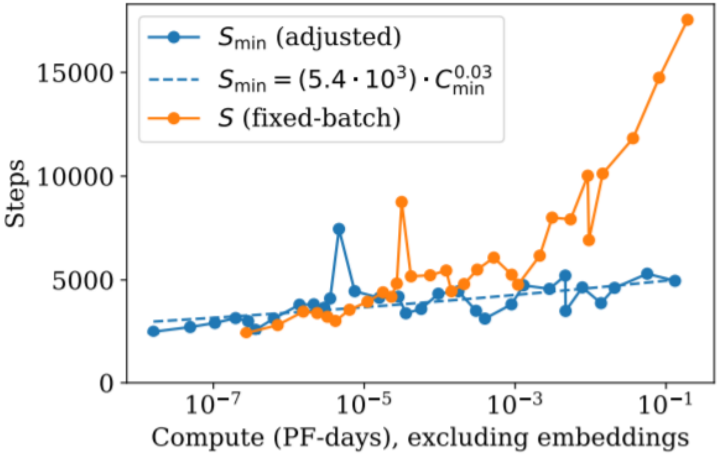
Scaling laws tell us: properly undertrained models are better

Compute tradeoffs (2)

Q: as we increase both compute and model size, how should we scale training?

- Huge batches, same number of steps
- Fixed batches, more steps





As we scale up language modeling with an optimal allocation of computation, we should predominantly increase the model size N, while simultaneously scaling up the batch size with negligible increase in the number of serial steps.

Good news for data parallel processing (?)

Scaling laws for models and compute

Log-linearity extends to model parameters and compute!

Lets us set the following based on small models

- Pick optimizer
- Pick architecture and model sizes

Also lets us make smart resource tradeoffs

- Big models vs more data?

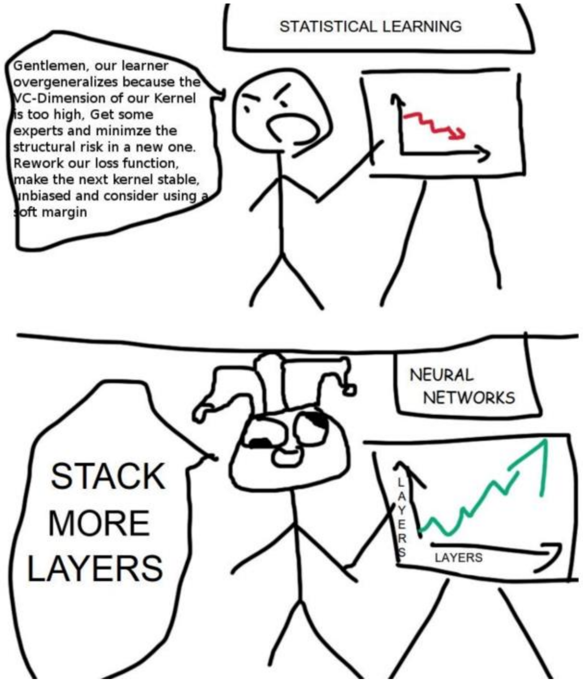
Scaling laws and the future

Q: Can big language models solve every problem?

We can use scaling laws to answer this!

- For each capability (e.g. question answering)..
- Build a scaling law for compute capacity.
- Extrapolate the scaling curve.

Can 'reasonable' amounts of compute solve our problems?



Taken from r/programmerhumor

Forecasting question: will we solve the Winograd schema?

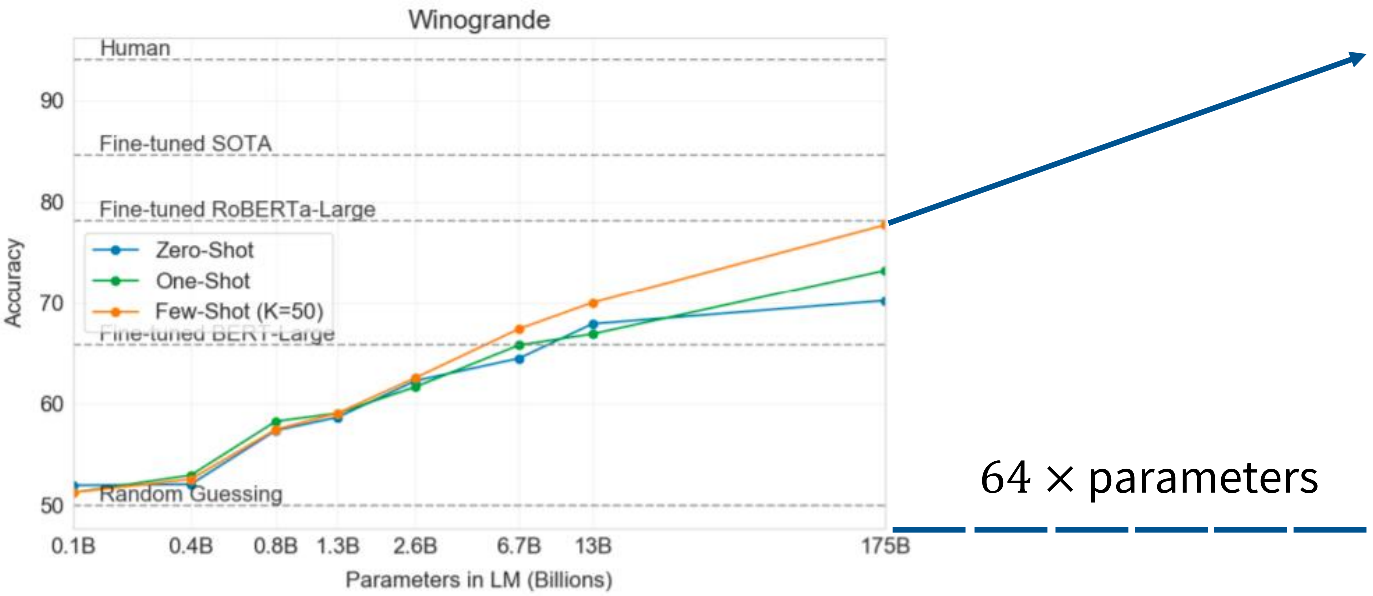
Classic AI challenge: Winograd schema

		Twin sentences	Options (**answer**)
✓ (1)	a	The trophy doesn't fit into the brown suitcase because **it**'s too *large*.	**trophy** / suitcase
	b	The trophy doesn't fit into the brown suitcase because **it**'s too *small*.	trophy / **suitcase**
✓ (2)	a	Ann asked Mary what time the library closes, *because* **she** had forgotten.	**Ann** / Mary
	b	Ann asked Mary what time the library closes, *but* **she** had forgotten.	Ann / **Mary**
✗ (3)	a	The tree fell down and crashed through the roof of my house. Now, I have to get **it** *removed*.	**tree** / roof
	b	The tree fell down and crashed through the roof of my house. Now, I have to get **it** *repaired*.	tree / **roof**
✗ (4)	a	The lions ate the zebras because **they** are *predators*.	**lions** / zebras
	b	The lions ate the zebras because **they** are *meaty*.	lions / **zebras**

Current GPT-3 performance after seeing 50 examples: 77%. Can we push this further?

How much more compute for human-level reasoning?

Just extend the line for the scaling law..

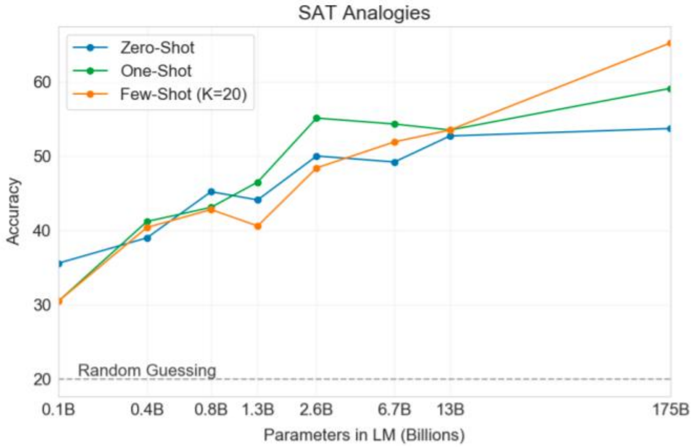


If the scaling law holds.. Roughly 64 times more parameters will get us to human-level

Another setting: SAT analogies

Context →	lull is to trust as
Correct Answer →	cajole is to compliance
Incorrect Answer →	balk is to fortitude
Incorrect Answer →	betray is to loyalty
Incorrect Answer →	hinder is to destination
Incorrect Answer →	soothe is to passion

Task: selecting the correct answer (with highest probability)

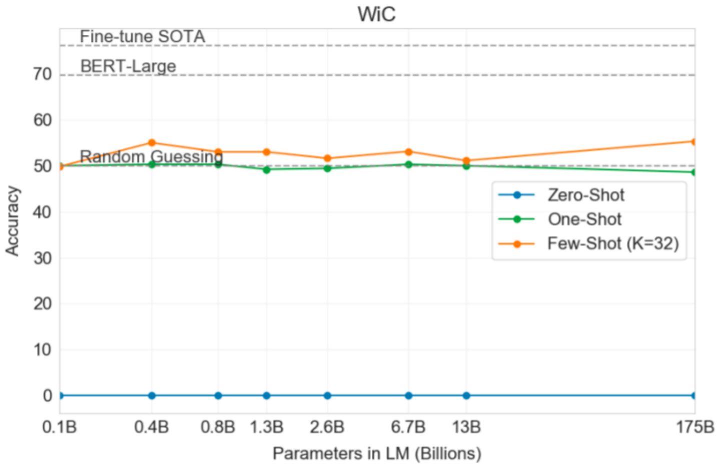


Scaling: clear linear scaling in log space.

Less optimistic scaling curves

Word in context dataset

Label	Target	Context-1	Context-2
F	bed	There's a lot of trash on the bed of the river	I keep a glass of water next to my bed when I sleep
F	land	The pilot managed to land the airplane safely	The enemy landed several of our aircrafts
F	justify	Justify the margins	The end justifies the means
T	beat	We beat the competition	Agassi beat Becker in the tennis championship



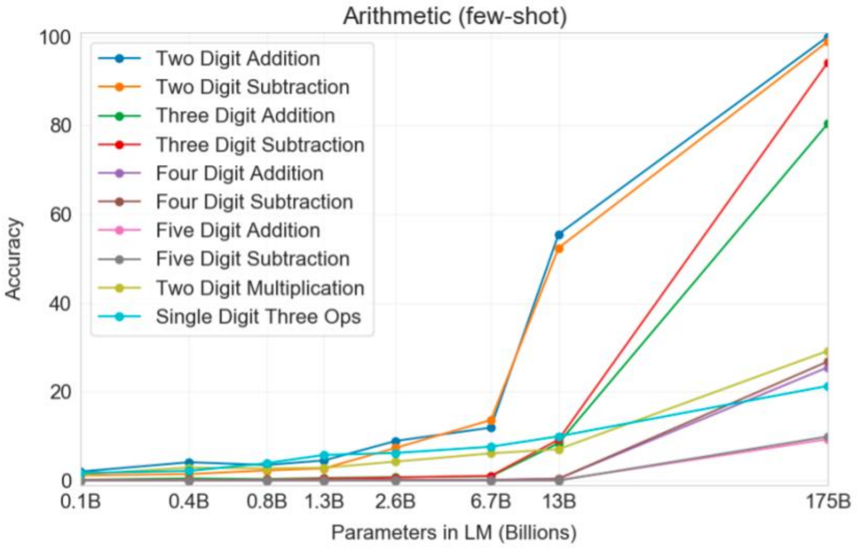
Scaling: near-zero. GPT-3 paper notes ‘pairwise comparison’ tasks are harder.

Phase transitions

Thus far: everything has had linear scaling (with different slopes).

Phase transitions are sudden, discontinuous jumps in performance.

The GPT-3 paper has some intriguing observations on phase transitions..



Do we expect to see more phase transitions?

This is probably the ‘big unknown’ in LM scaling!

Scaling laws and the future

Some tasks will just improve continually via scale (Winograd, SAT etc)

There are some others that may have 'phase transitions' and emergent behavior

Finally, more work to be done on some tasks (WiC?)

Scaling laws can help with a key question: what problems can we brute force?

Recap: scaling laws – surprising and useful!

- **Data scaling**: understand how data affects models, clean theory
- **Model scaling**: dramatically reduce costs for training
- **Scaling as prediction:** understand what problems can be 'brute forced'

Scaling laws are interesting for everyone!

- Theorists (why do we get scaling laws)
- Practitioners (lets use scaling laws to optimize)
- AI enthusiasts (can we get AGI with more gpus?)