Logistics update

- HW7 is due this week Thursday.
- It is also the last homework … yay!
- Q4 is now an extra credit.

- Project proposals are due this week.
- Please continue to discuss the project ideas with each other and the course staff.

- **Common question:** What is the expected amount of effort for the project?
  - Imagine how much effort you put toward finishing the weekly homework assignments. Now you'd put that toward the final project.
  - final project = ~[4-6] x homework assignment

On Project Proposals

- Make sure that you follow the expected protocol for the proposal.

  - The project proposal is a 2 page description of what you intend to do
    - motivation,
    - hypothesis,
    - experiments,
    - datasets,
    - methods,
    - expected outcome,
    - etc.

- If you're missing these details, we will not receive the "proposal" credits and we will ask you to redo it.

On Project Proposals: Examples of Wordings

"we will first collect an extensive dataset from various sources such as Google News articles, Reddit discussions, and Twitter conversations."

What data? What annotations?

"the model will be trained to produce a sentiment indicator that ranges from -10 to 10 …"

What model?
Where did we get these labels?

You want to be as clear and as specific as possible.

Scaling LMs 

Scaling

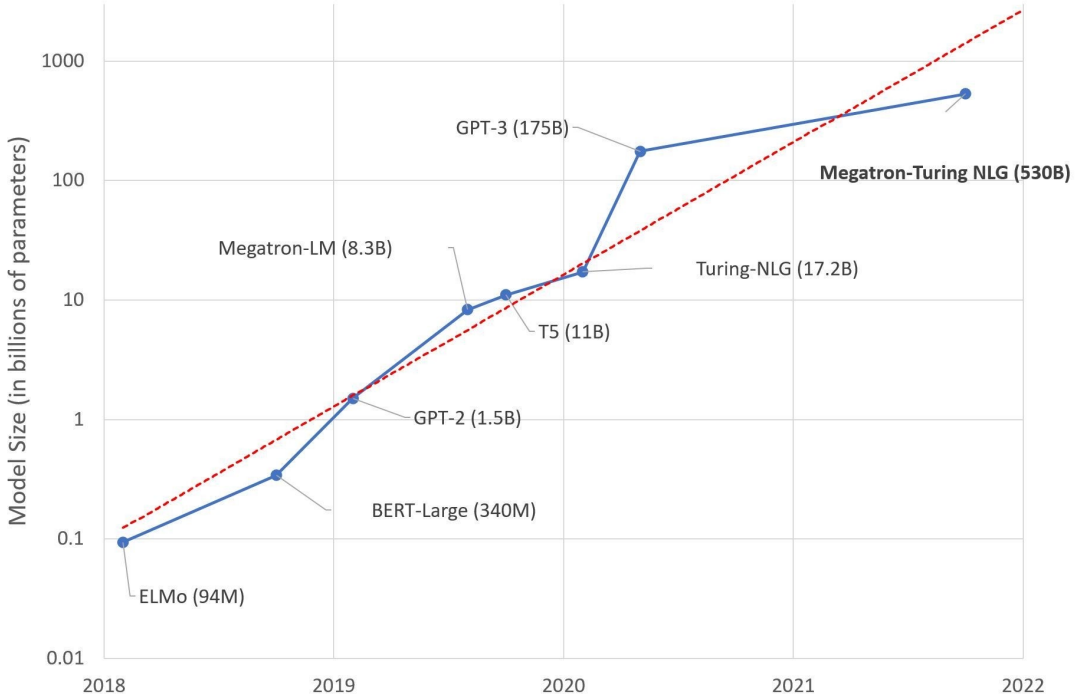
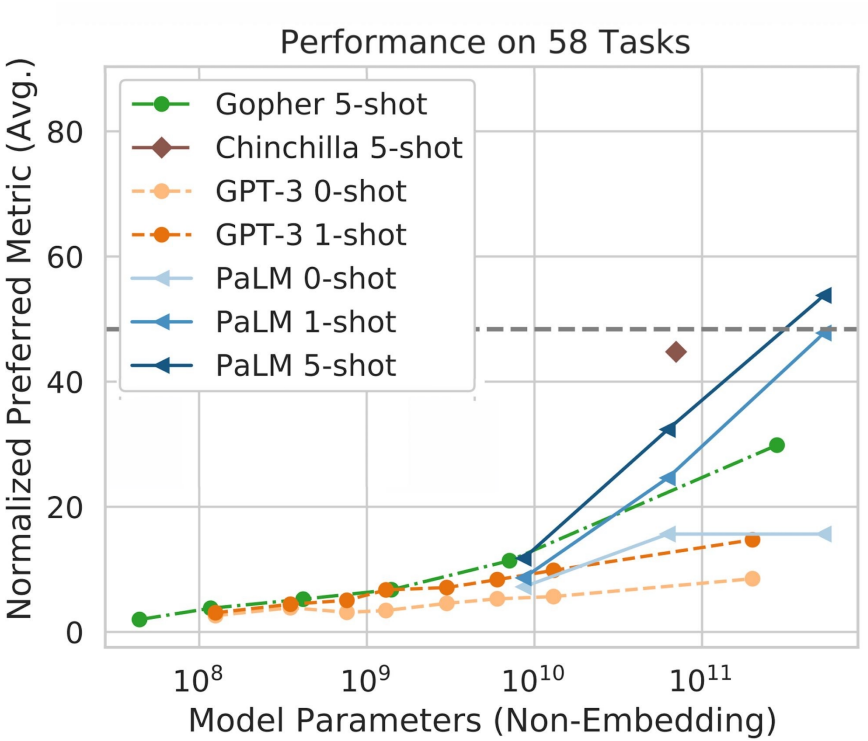


Photo credit: Microsoft Research Blog, Alvi et. al., 2021

LM are getting larger and more expensive

Scaling

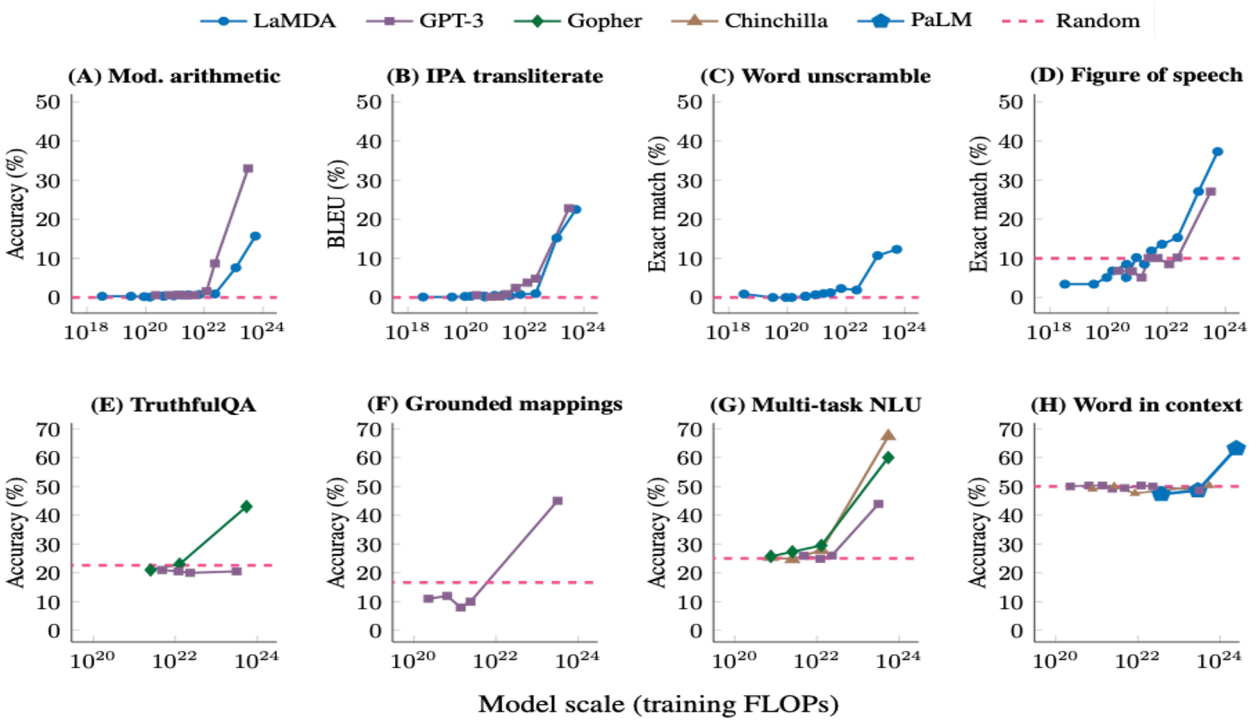
Photo credit: PaLM, Chowdhery et. al., 2022



Larger LMs ⇒ better zero/few-shot performance

Large language models exhibit emergent abilities

- With scaling models their ICL perf consistency improves.



Emergence —qualitative changes in behavior with some scaling parameter

What is “Scaling”?

- “scaling means **larger model size**”
  - But model parameters may be under-utilized.
- “scaling means **more compute**”
  - But computation may be unnecessarily wasted.
- “scaling means **more data**”
  - But more data might not necessarily contain more information (e.g., duplications)
- Scale means all the above: *effective compression of information*
  - Requires model capacity
  - Requires compute
  - Requires large, rich data

Scaling Laws

- **Hypothesis:** there are fundamental principles that govern effective scaling

- **Importance:** understanding these "laws" would allow us to find optimal models for a given data/compute budget.

- Think of Newton's laws
  - Provide the basis for understanding and analyzing the motion of objects in the physical world
  - Can be used to calculate the trajectory of a rocket, the speed of a car, or the motion of a planet.

Scaling - Optimal Model Size

- Smaller models don't have enough capacity to utilize the extra compute. They plateau early.
- Larger models are initially slower to train, but with more compute they reach lower losses.
- Given a compute budget, what is the optimal model size to use?

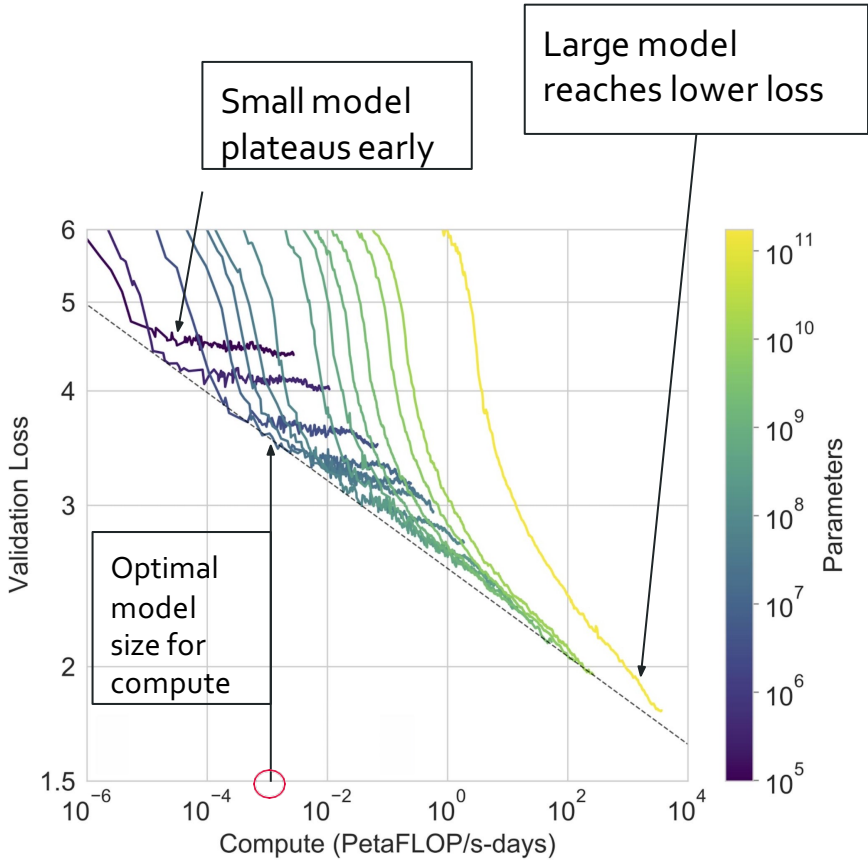


Photo credit: GPT3, Brown et. al., 2020

[ACL 2022 Tutorial Beltagy, Cohan, Logan IV, Min and Singh]

Scaling - Optimal Model Size

- The idea of "optimal model size for given compute" was introduced by Kaplan et. al.

- In ideal world, we are given lots of compute to train many models to find the optimality.
- Alas not feasible when you have budget to train a single model.

- If we have the equations ("laws") describing the behavior, we can compute it analytically.

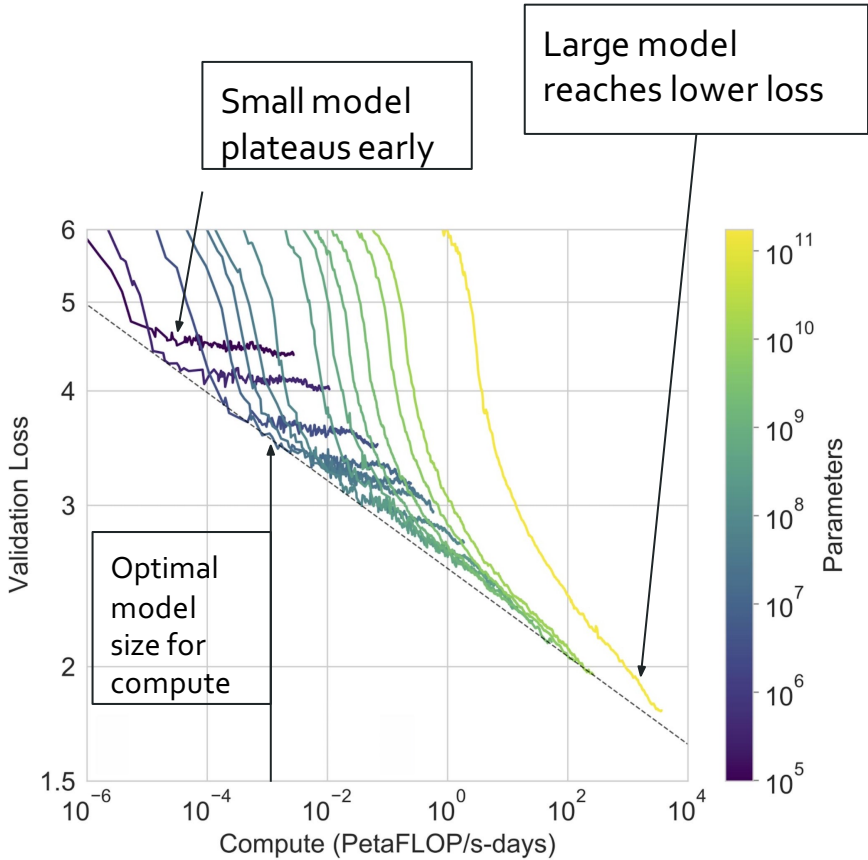


Photo credit: GPT3, Brown et. al., 2020

[ACL 2022 Tutorial Beltagy, Cohan, Logan IV, Min and Singh]

Scaling Laws of Kaplan et. al., 2020

- Also, fit a power-law function to predict the "compute efficient" frontier:

$$L \propto C^{-0.048}$$

- Use it to predict the optimal model size and optimal number of tokens, for a given amount of compute

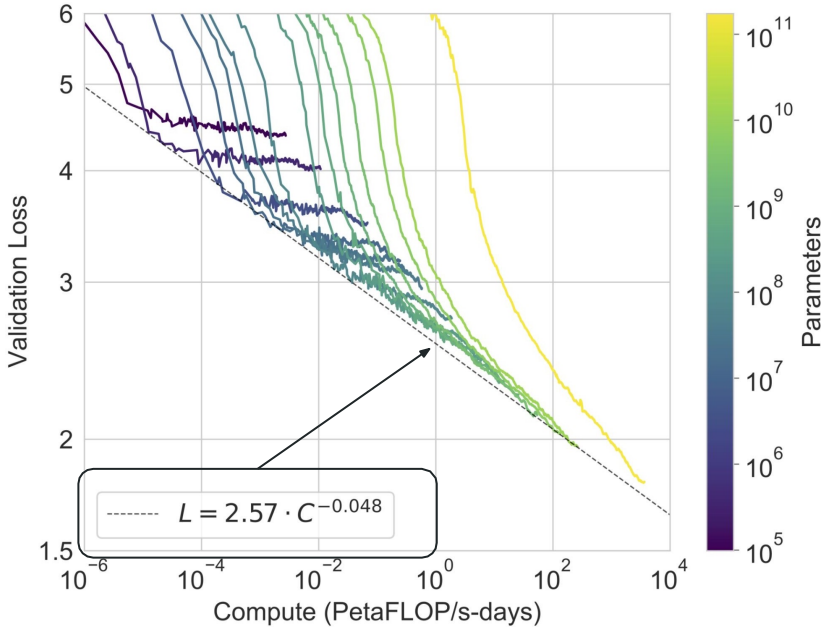


Photo credit: GPT3, Brown et. al., 2020

[ACL 2022 Tutorial Beltagy, Cohan, Logan IV, Min and Singh]

Scaling Laws: Kaplan et al.

N: number of model parameters
C: compute
D: dataset size

- Optimal model size and optimal number of tokens, for a given compute budget

Kaplan et. al. 2020	$N_{opt} \propto C^{0.73}$	$D_{opt} \propto C^{0.27}$

N_{opt} exponent >> D_{opt} exponent

- **Takeaway:** grow the model size faster than growing the number of tokens.
  - **Example:** Given 10x compute, increase N by 5.5x, and D by 1.8x
- GPT3 (and many other followed this recipe) training a 175B model on 300B tokens

[ACL 2022 Tutorial Beltagy, Cohan, Logan IV, Min and Singh]

Scaling Laws: Hoffmann et al.

N: number of model parameters
C: compute
D: dataset size

- Optimal model size and optimal number of tokens, for a given compute budget

Kaplan et. al. 2020	$N_{opt} \propto C^{0.73}$	$D_{opt} \propto C^{0.27}$
Hoffmann et. al., 2021	$N_{opt} \propto C^{0.50}$	$D_{opt} \propto C^{0.50}$

N_{opt} exponent $\cong$ D_{opt} exponent

- Compute and tokens should increase at the same rate.
  - **Example:** Given 10x compute, grow N by 3.2x and D by 3.2x
- Hoffmann et. al., followed this recipe:
  - Trained a 70B model on 1.4T tokens (Chinchilla) outperforming their previous 280B model trained on 300B tokens (Gopher)

[ACL 2022 Tutorial Beltagy, Cohan, Logan IV, Min and Singh]

Scaling - Kaplan et. al., 2020 vs. Hoffmann et. al., 2022

- The main difference is the learning rate schedule!!!

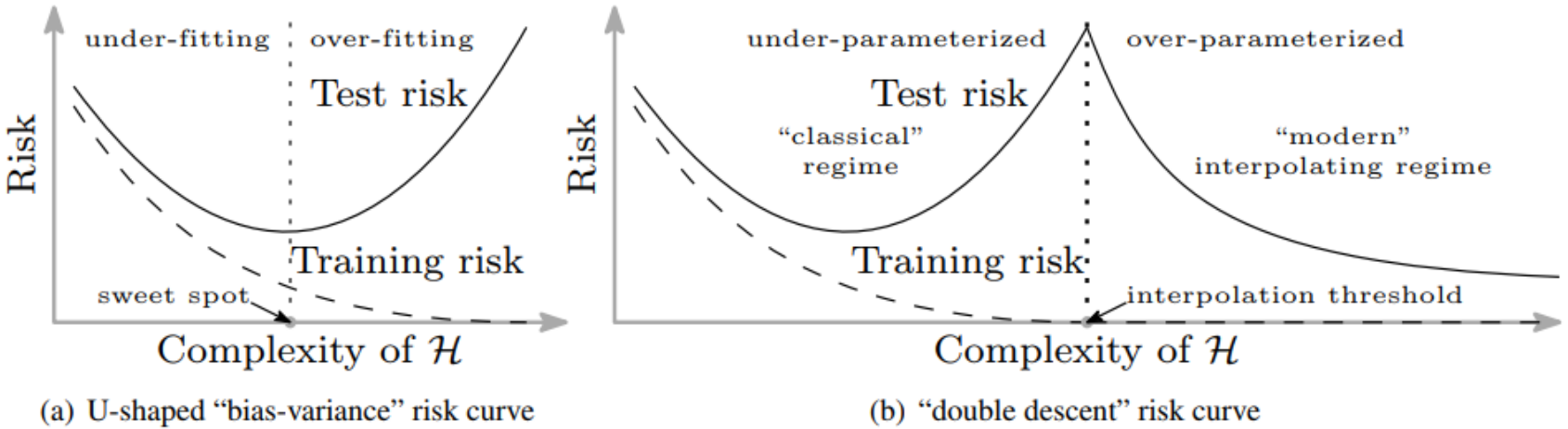
- Kaplan et. al.: all experiments same LR schedule
- Hoffmann et. al. experiment while changing different LR schedule
  - Their LR reaches zero at the end of the training

[ACL 2022 Tutorial Beltagy, Cohan, Logan IV, Min and Singh]

Why didn’t we scale earlier??

ML Theory Told Us Otherwise

- Learning theory made us allergic to over-parametrized models.



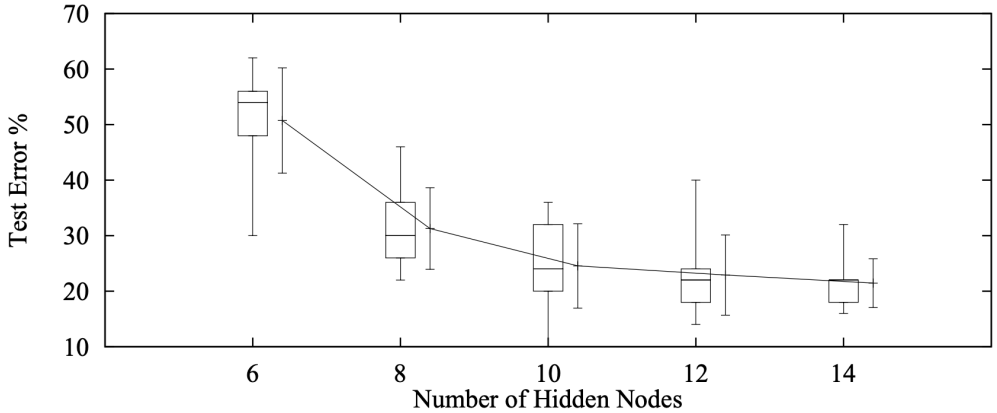
(a) U-shaped "bias-variance" risk curve

(b) "double descent" risk curve

[Reconciling modern machine learning practice and the bias-variance trade-off, Belkin et al. 2019]

There Were Empirical Evidence

- Even in mid-90's there were evidence supporting the benefit of larger models
- Although they were ignored ….



" …. larger networks may generalize well and better generalization is possible from larger networks if they can be trained more successfully than the smaller networks" -- Lawrence, Giles, and Tsoi in 1997

[Lawrence, Giles, and Tsoi 1997]

There Were Empirical Evidence

- Even in mid-90's there were evidence supporting the benefit of larger models
- Although they were ignored ….

" …. "Nets of all sizes overfit some problems. But generalization is surprisingly insensitive to excess capacity if the net is trained with backprop."

-- Caruana, Lawrence, and Giles (2000)

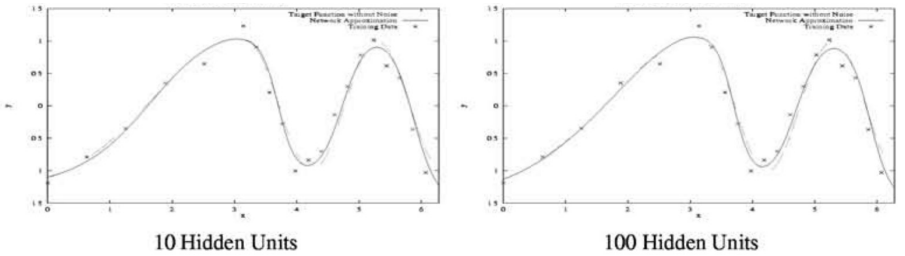


Figure 3: MLP approximation using backpropagation (BP) training of data from Equation 1 as the number of hidden units is increased. No significant overfitting can be seen.

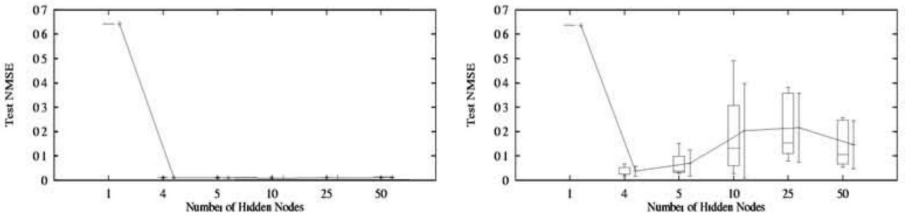


Figure 4: Test Normalized Mean Squared Error for MLPs trained with BP (left) and CG (right). Results are shown with both box-whiskers plots and the mean plus and minus one standard deviation.

[Caruana, Lawrence, and Giles 2000]

Retrieval-Augmented LMs

What is Missing from LMs?

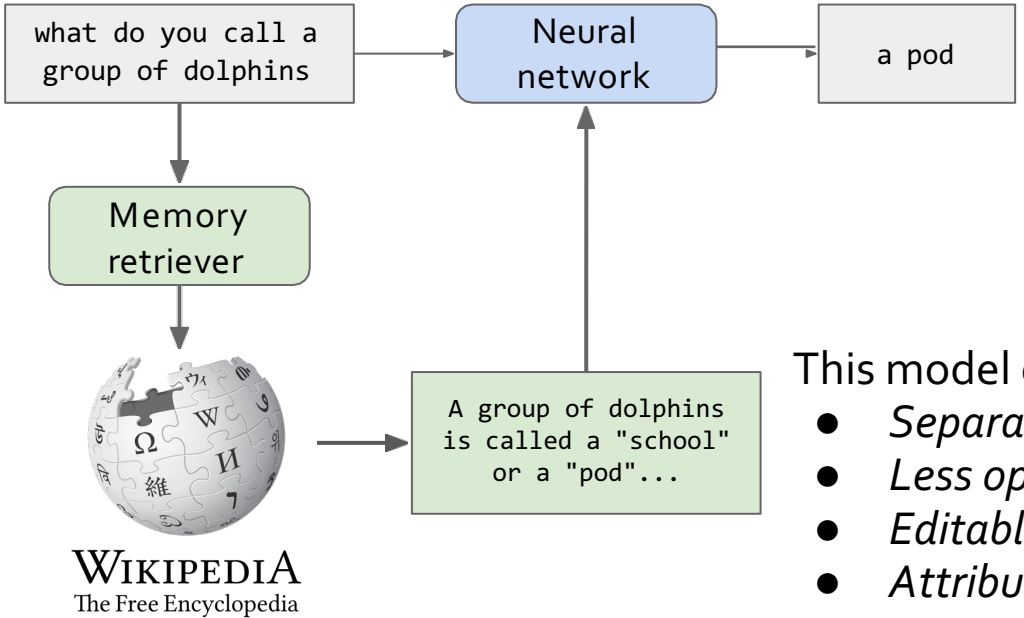
- LMs automatically acquire knowledge from the web, but…
- … a lot of it is noisy or incorrect: misinformation, rumors, opinions.
- … we cannot trace the model's knowledge back to an attributable source.

- We can edit individual facts inside a Transformer's memory, but…
- … it doesn't work reliably yet.
- … current approaches break down after multiple edits.

- We can store knowledge inside feedforward layers, but…
- … current memory capacity is too small, and scaling up is expensive!

[Slides: Kelvin Guu]

What is a Memory-Augmented Model?



A memory could be:

- *Documents on the web*
- *Record in a database*
- *Images of the world*
- *…*

This model can have desirable attributes:

- *Separates world knowledge from LM parameters*
- *Less opaque; more interpretable*
- *Editable knowledge*
- *Attribution*
- *More efficient scaling*
- *…*

[Slides: Kelvin Guu]

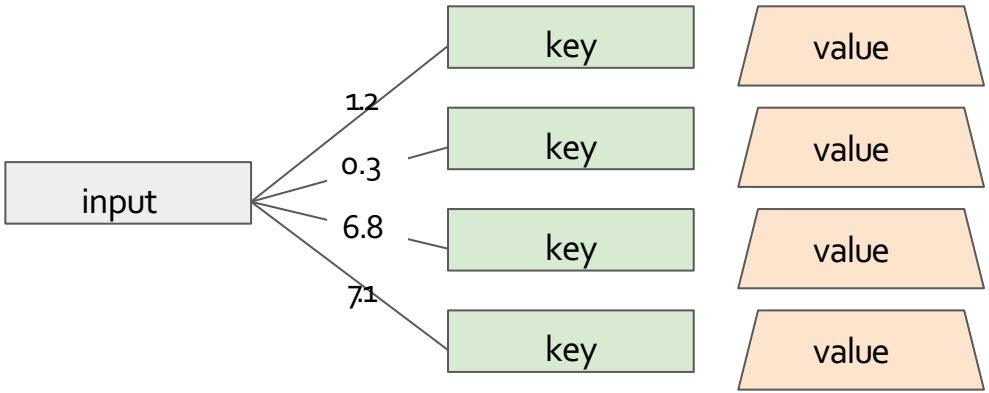
What are the Key Design Questions?

- What are your memories?
  - Documents, database records, training examples, etc.
- How to retrieve memories?
  - Use an off-the-shelf search engine (e.g. Google, StackOverflow).
  - How to train your own memory retriever.
- How to use retrieved memories?
  - "Text fusion", "label smearing".
  - Common failure modes:
    - Underutilization: model ignores retrieved memories.
    - Overreliance: model depends too much on memories!

[Slides: Kelvin Guu]

Anatomy of a Neural Retriever

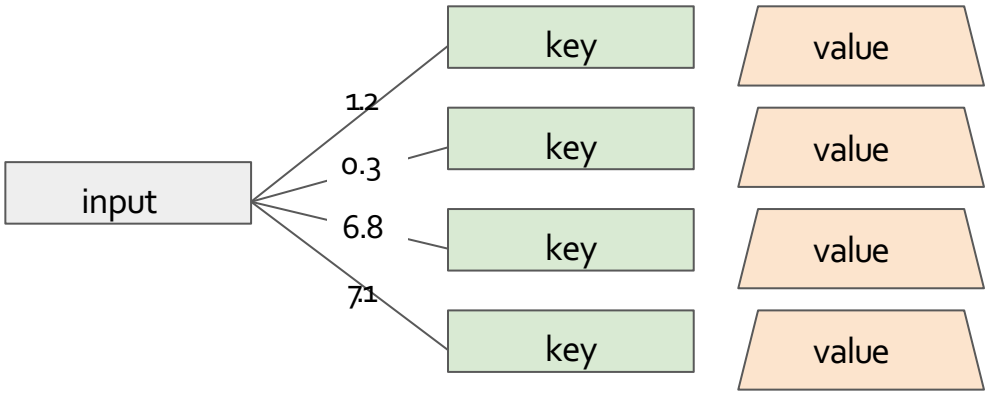
1. Score the input against each key.
2. Return the value for the highest scoring key.



[Slides: Kelvin Guu]

Anatomy of a Neural Retriever

1. Score the input against each key.
2. Return the value for the highest scoring key.



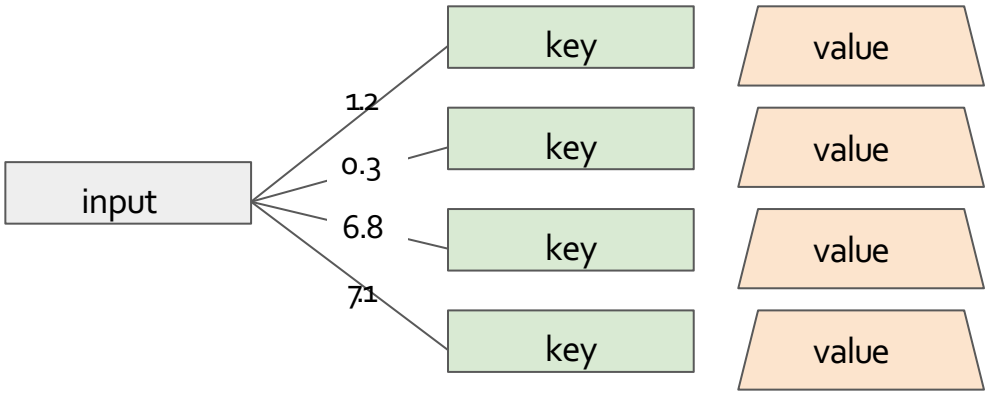
Example:

- Input = “An essay on Eiffel Tower”
- Key = <doc title>
- Value =<doc content>

[Slides: Kelvin Guu]

Anatomy of a Neural Retriever

1. Score the input against each key.
2. Return the value for the highest scoring key.

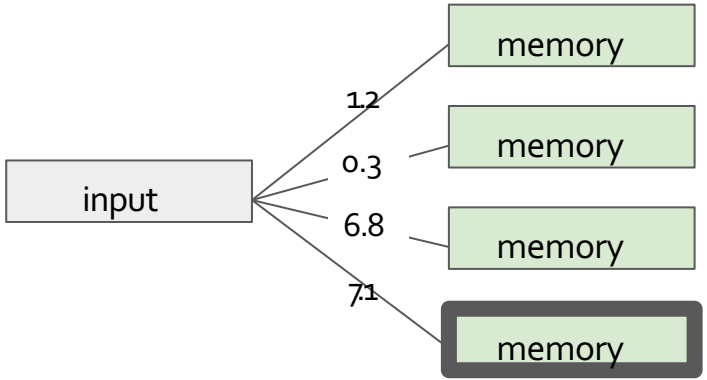


A retriever is just a function: **`f(input, key) → score`**

[Slides: Kelvin Guu]

Anatomy of a Neural Retriever: Simplified

1. Score the input against each memory.
2. Return the highest scoring memory.

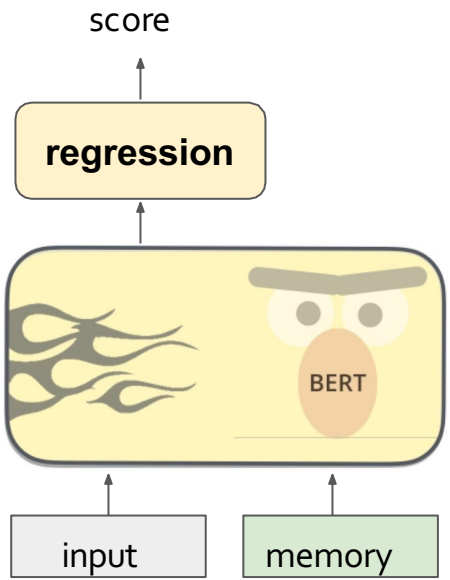


A retriever is just a function: `f(input, memory)` → `score`

[Slides: Kelvin Guu]

Retrieval Scoring Function

`f(input, memory) → score`



Advantages:

- Using a powerful Transformer model to compare the input against each memory.
- Differentiable -- can optimize with gradient descent.

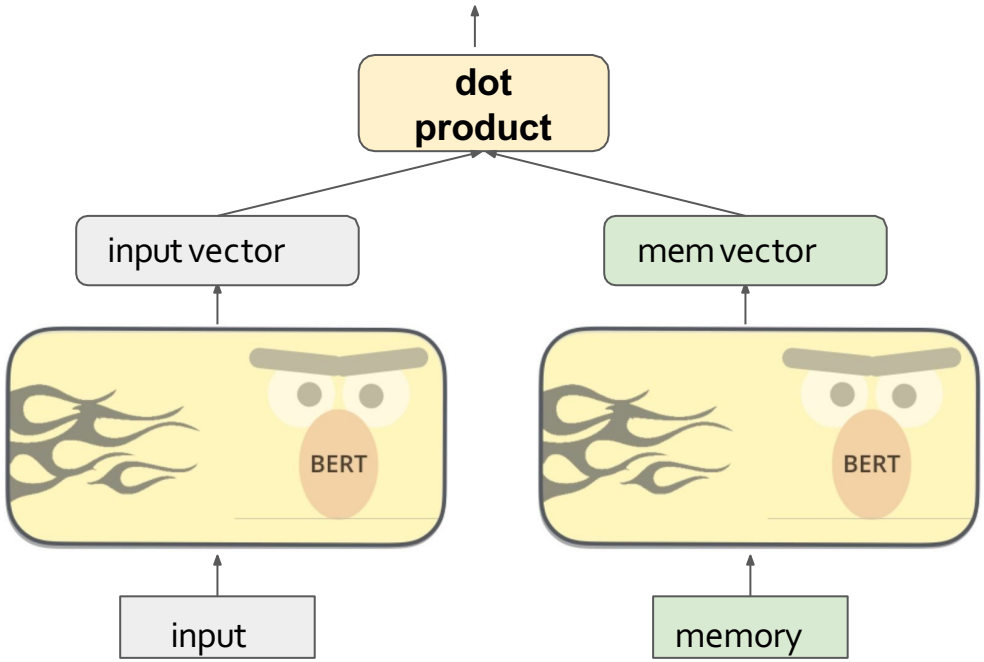
Disadvantages:

- For each new input, you have to do this comparison against EVERY memory.
- Too slow if you have millions of memories.

[Slides: Kelvin Guu]

Retrieval Scoring Function

`f(input, memory) → score`



Advantages:

- Can precompute all memory vectors.
- Only have to do this once, NOT for every input.
- Computing a simple dot product is fast.
- Differentiable -- can optimize with gradient descent.

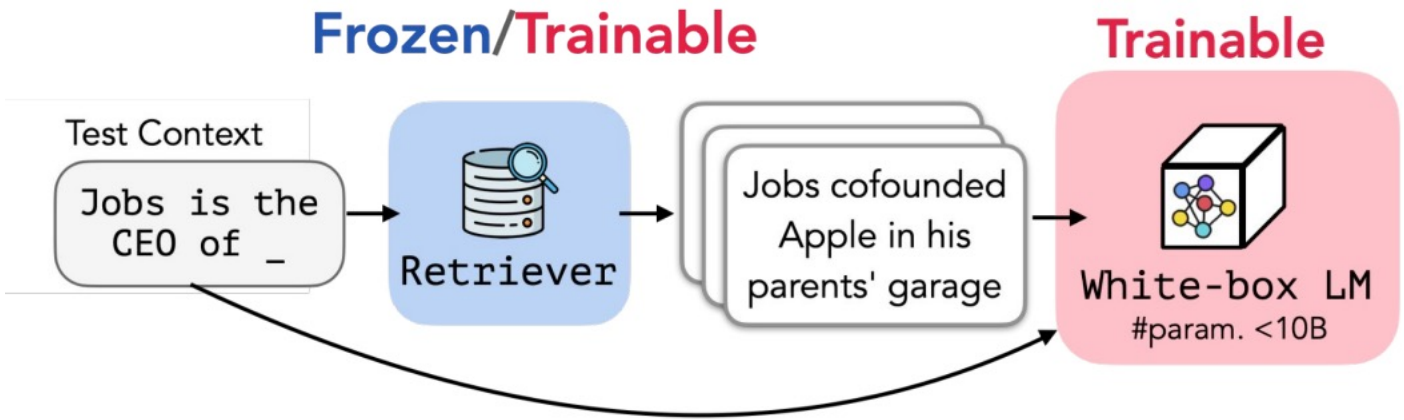
Disadvantages:

- Dot product is not very expressive.

[Slides: Kelvin Guu]

End-to-end Training

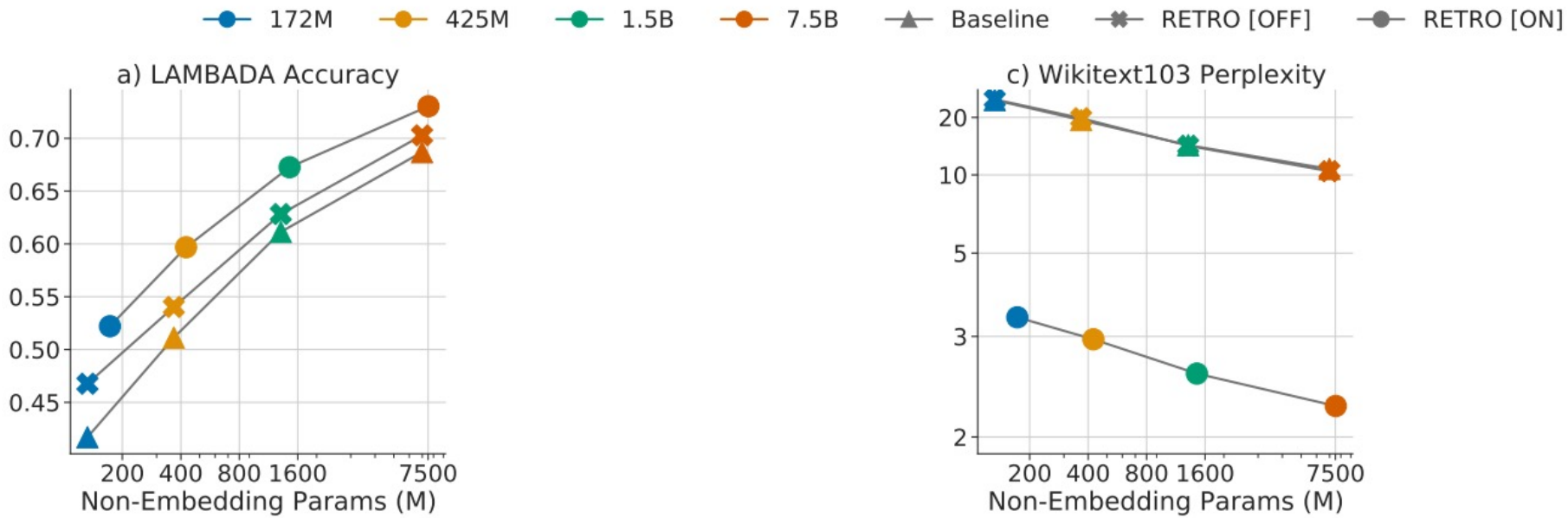
- There are various ideas in the literature for how to train these models efficiently and in an end-to-end fashion.



Retrieval-Augmented Language Models

- RAG (Lewis, et al. 2020)
- REALM (Guu et al. 2020)
- SPALM (Yogatama et al. 2021)
- RETRO (Borgeaud, et al. 2022)
- RePlug (Shi et al. 2022)
- RA-CM3 (Yasunaga et al. 2022)
- BlenderBot 2.0 (Komeili et al. 2021)

Highlighting Some Results



Main takeaways

- How do we enable LMs to utilize external knowledge?
  - Retrieval-augmented language models
- A retriever is a function, f(input, memory) → score
- What we did not discuss:
  - Tracing decisions to the source knowledge
  - How to modify the knowledge
  - Conflicting knowledge
  - ….