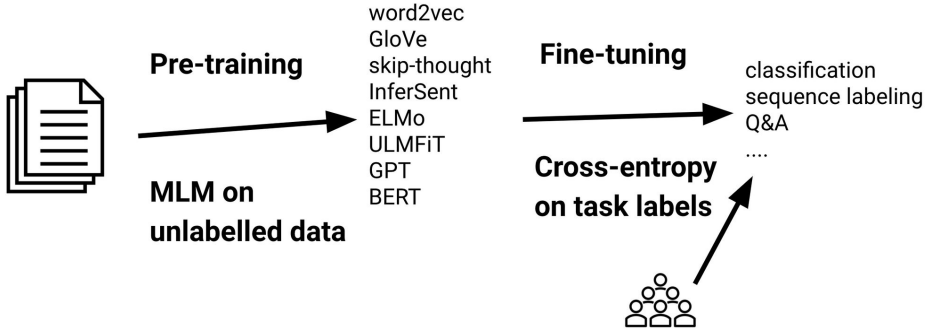
What Can Transformers Learn In-Context? A Case Study of Simple Function Classes

Presenter: Yulai Zhao
9/23/2022

Prompt: A new perspective in NLP

- Why Prompt? Why not “pre-train -> fine-tune”



Cost!

- Prompt is designed to avoid finetuning in downstream task!

Benefit: NO Fine-tuning Cost, No Model Update

Prompt: A new perspective in NLP

- “What Prompt does?”

  Implicitly tell the LM what we want in downstream task.

- How is Prompt used?

  “**pre-train, prompt, and predict**” scheme, a refinement of “pre-train -> fine-tune”

An Example of Prompt Learning

Recall the classic Masked Language Model task in NLP pre-training.(e.g., BERT)

- We have a sentence (could be fetched from a corpus, e.g., Wikipedia)

  "**I love this movie. Overall it is so exciting**!"

- In NLP pre-training, MLM will do a word puzzle by masking words randomly

  "**I ___ this movie. Overall it is so exciting!"**

- This incomplete sentence will be fed into model, which tries to predict the blank from its word vocabulary.

Prompt Learning: “pre-train, prompt, and predict”

For the simplest prompt

- We make the formulation of downstream task **similar** to that of pre-training.
- Prompt learning gives: $x' = f_{pr}(x)$

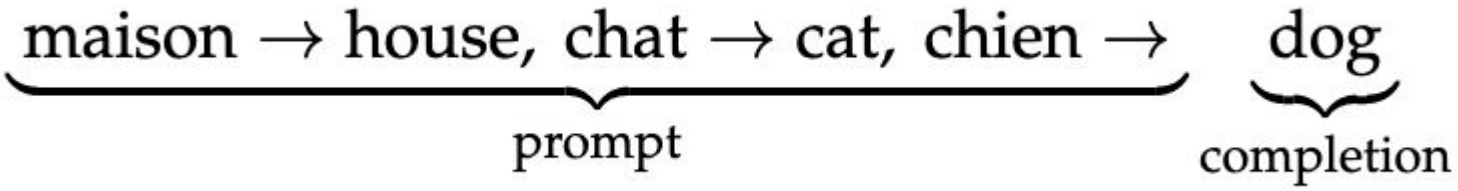
  where x = **“I love this movie”**, then x’ should be

  “I love this movie. Overall it is so ___!“

- Finally, language model would predict ___, which is “exciting” in this case.

In-context Learning

Brown et al. found that GPT-3 can perform ***in-context learning***---i.e., given a prompt containing examples from a task (input-output pairs) and a new query input, it can produce the corresponding output.



French to English in-context learning

In-context Learning of A Transformer

A typical learning algorithm takes:

- a sequence of input-output examples (x_i, f(x_i)), and estimates f(x_query) on a new input x_query.

WHY Study It?

And How To?

- Can a transformer encode such learning algorithms?

Problem Formulation

Suppose prompt is (x_1,f(x_1), ..., x_k, f(x_k), x_query). Here data and f are all randomly sampled.

We say a languate model M (e.g., GPT-3) could perform "intext-learning" w.r.t. function class F when:

$$\mathbb{E}\ \mathbf{L}(M(x\_prompt), f\_query)$$

is small.

However, analyzing from theories is not promising, at the moment.

How do we do this experimentally?

The simplest setting: **Linear Class, Transformer**

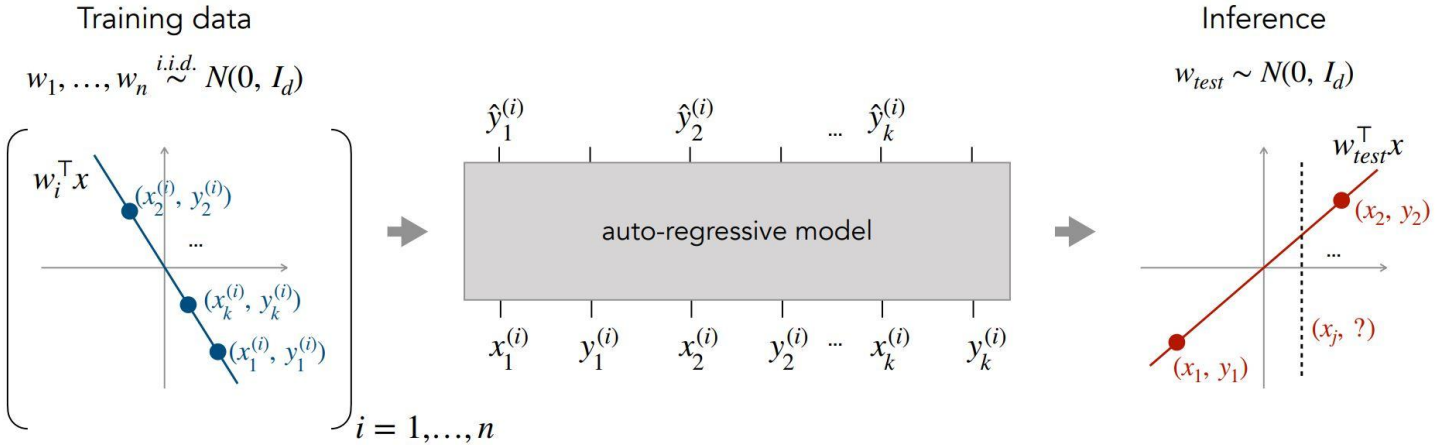
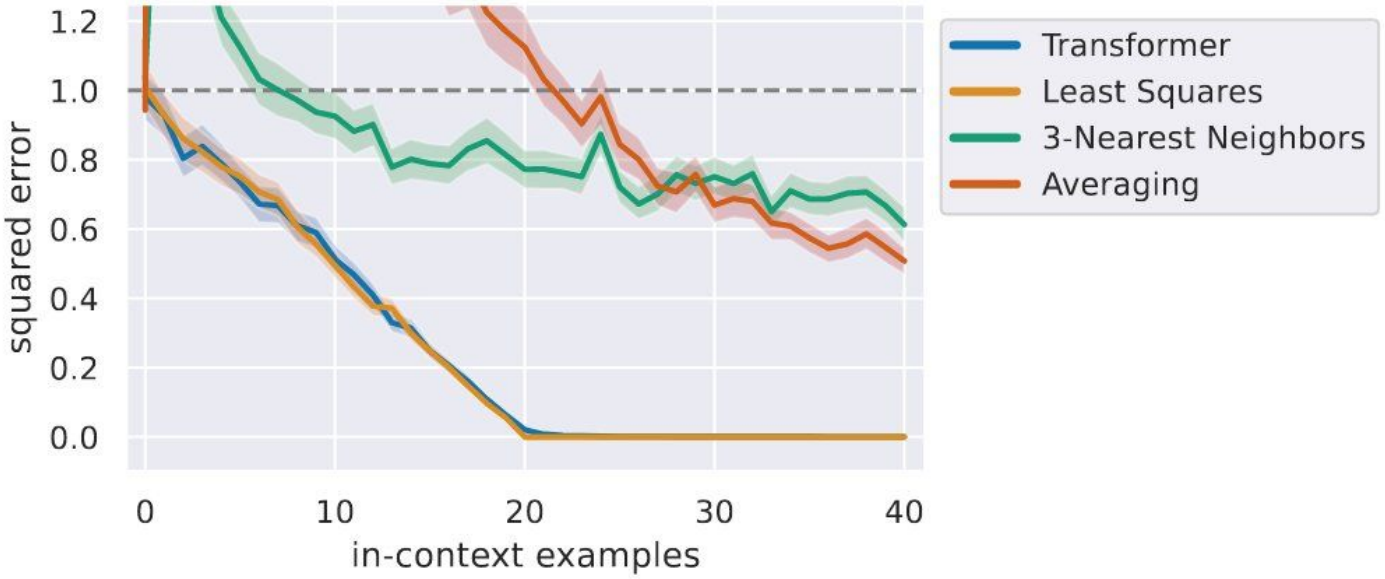


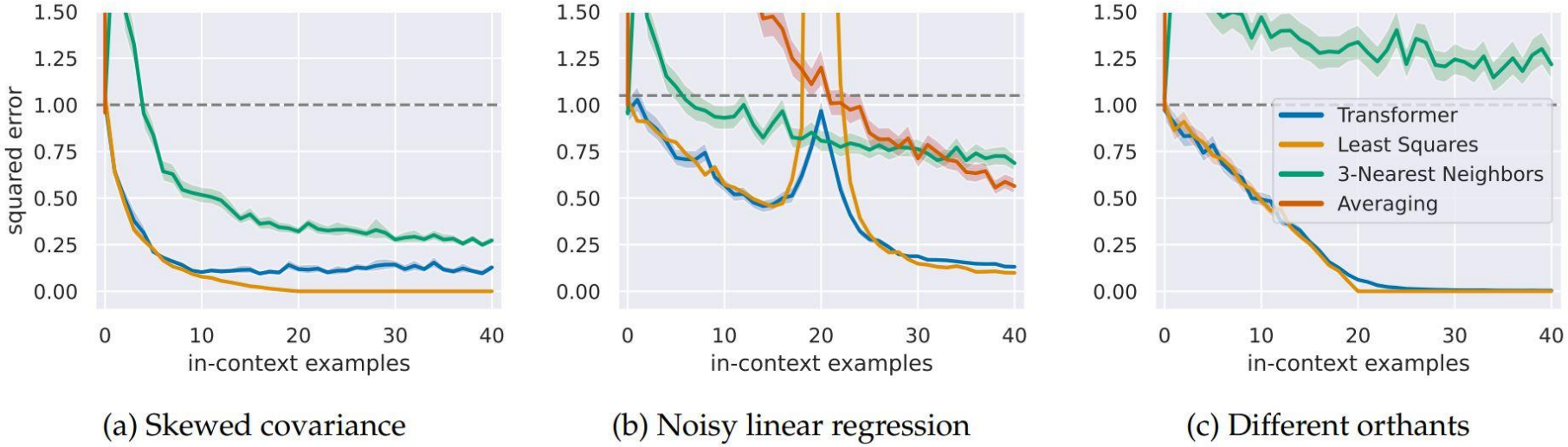
Figure 1: *Can we train a model that in-context learns a function class (here linear functions)?* We train Transformers by repeatedly sampling a random function f from that class, as well as random inputs $x_1, \ldots, x_k$ and training the model to predict each $f(x_i)$ given the prompt $x_1, f(x_1), \ldots, x_{i-1}, f(x_{i-1}), x_i$ (wrt squared loss). Then, during inference, we evaluate the model's ability to predict accurately on new, *unseen* functions.

Transformers CAN learn in-context linear functon



Prompts are sampled from the same distribution as in training.(obvious)

In-context learning on out-of-distribution prompts? No problem!



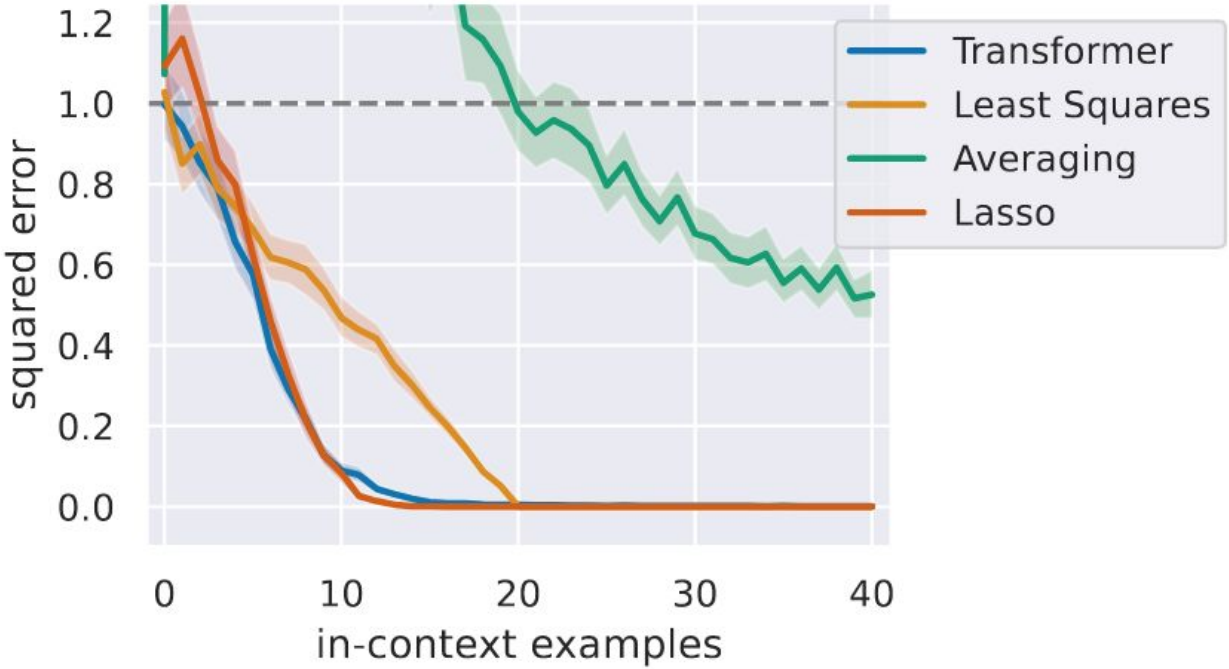
(a) Skewed covariance (b) Noisy linear regression (c) Different orthants

Showing the generality of its in-context learning ability

In-context learning on more complex function classes? More examples studied

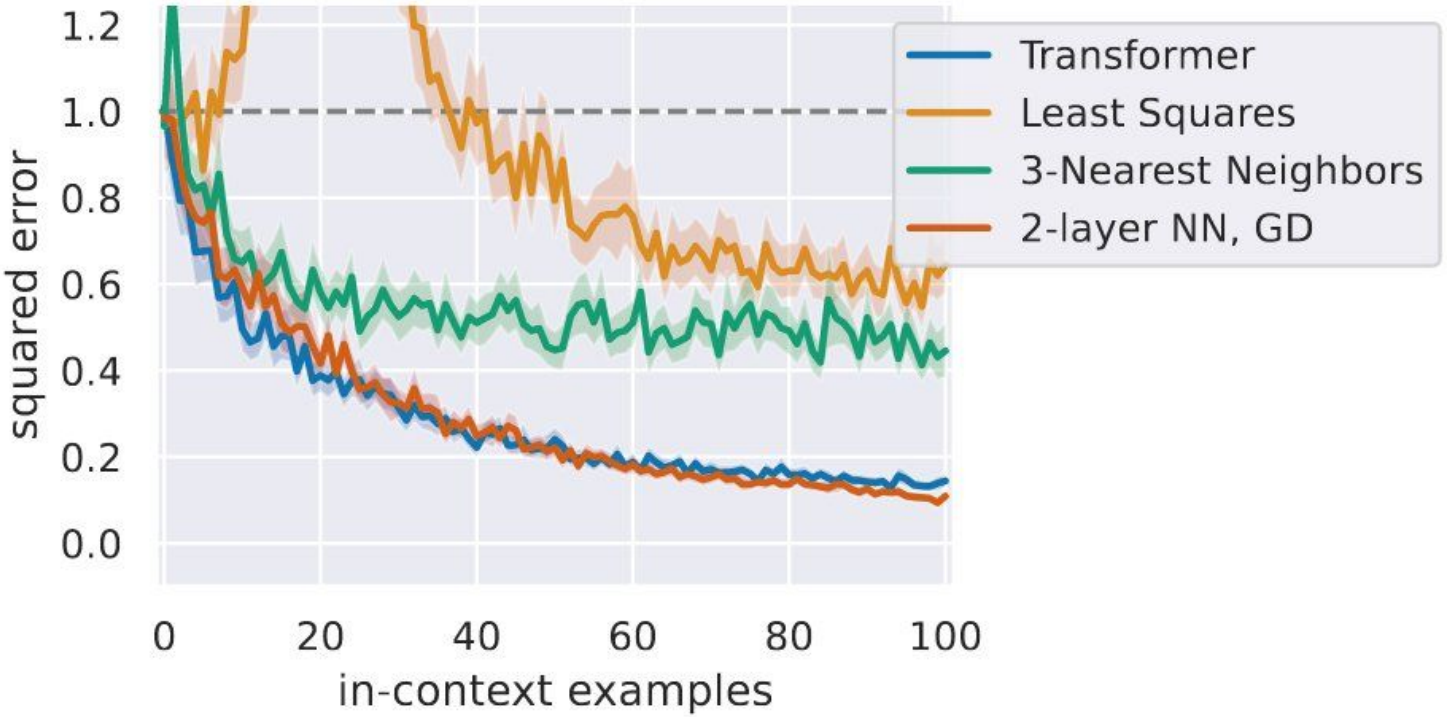
1. Sparse Linear functions
2. Random decision trees
3. Random 2-layer neural networks

Comparable with Lasso on Sparse linear regression



Better than XGBoost on random decision trees

Comparable with GD on 2-layer random NN



Conclusion

1. It is shown that Transformers can be trained to encode far-from-trivial learning algorithms.
2. The performance of Transformers is comparable, if not better, to classical methods.
3. Why? No idea… Future work.