Faculty of Computer Science

Научно-исследовательский семинар

2022

What Can Transformers Learn In-Context?

A Case Study of Simple Function Classes

Akulov Dmitry

In-context learning

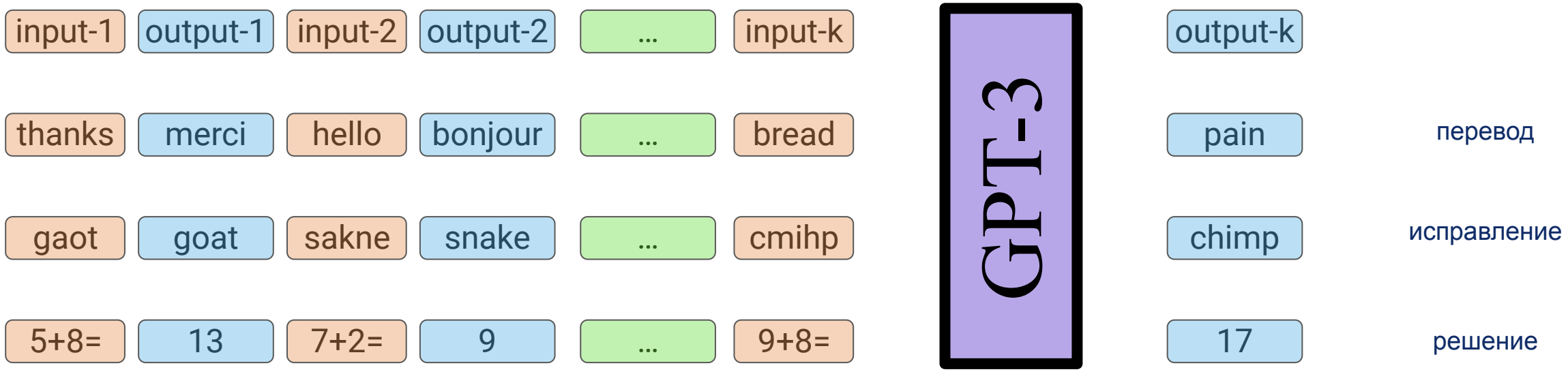
Prompt

input-1 output-1 input-2 output-2 ... input-k

Expected output

output-k

In-context learning



Примеры работы in-context learning задач

In-context learning

Prompt

input-1 output-1 input-2 output-2 ... input-k

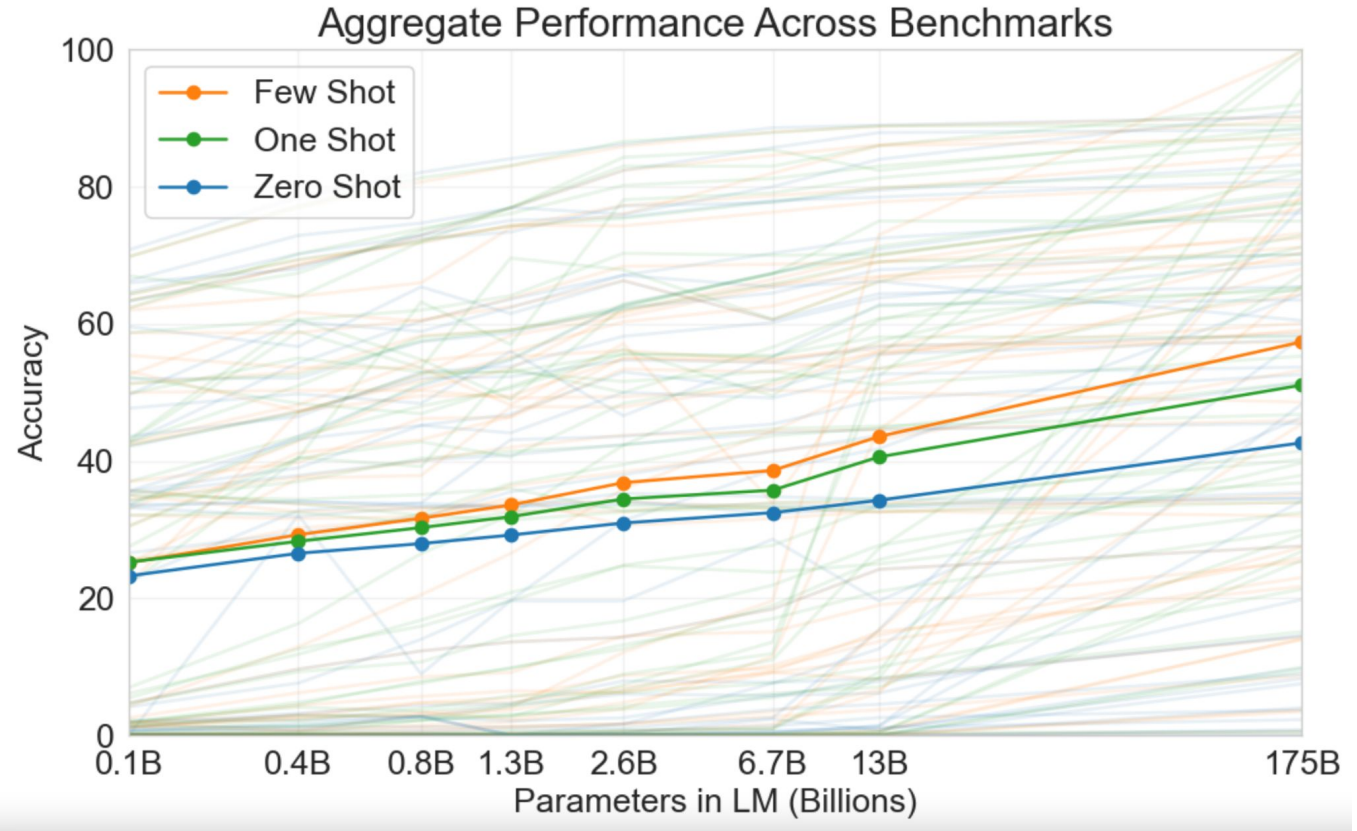
Expected output

output-k мысленная разметка

In-context learning

- Не подается явной информации о разбиении на input-output, ожидается просто генерация токена
- Модель не обучается явно ни на одну из подобных задач (могла ли просто запомнить что-то?)
- Обучение происходит только на стадии применения, веса не обновляются
- Обычно в контексте языковых моделей, но в целом применимо к генеративным
- Релевантно для **больших** моделей

In-context learning

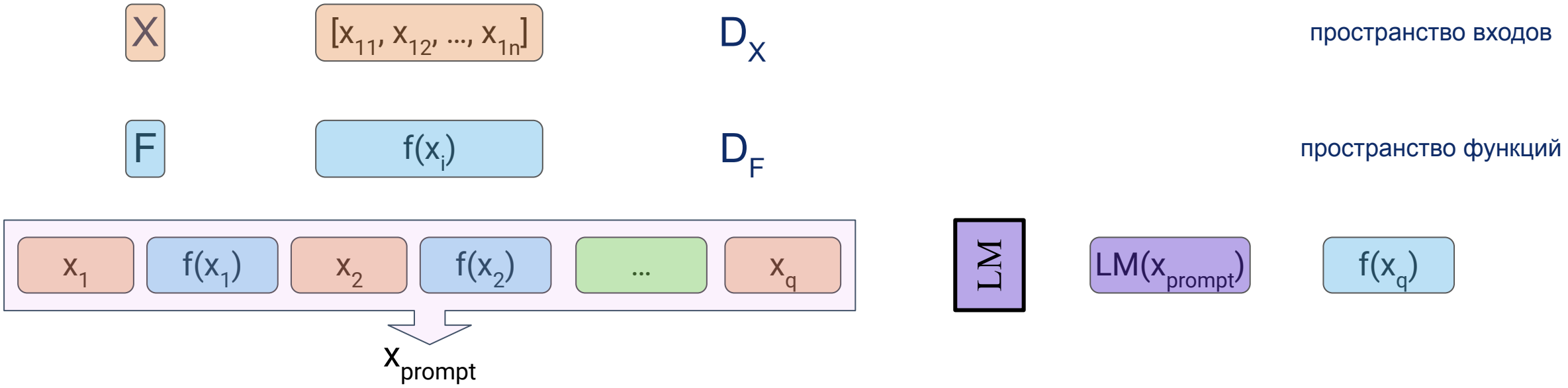


Качество работы GPT-3 с разным количеством параметров на различных задачах

Вопрос работы

Непонятно, какова связь между in-context задачами, на которых преуспевают GPT-3 и другие модели, и тем, что представлено в тренировочных данных.

Постановка задачи

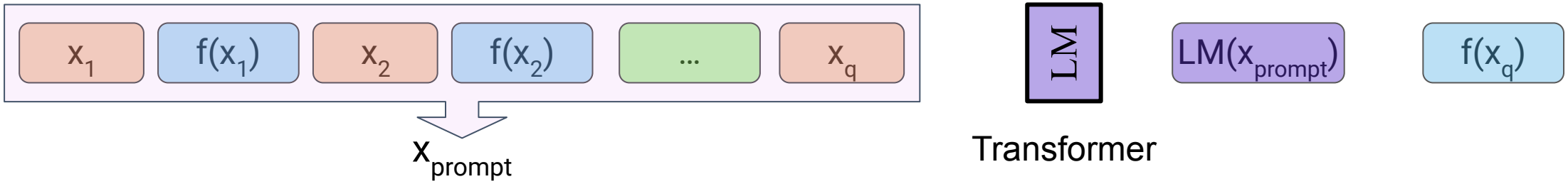


$$E_{x_{prompt}}[l(LM(x_{prompt}), f(x_q))] \leq \epsilon$$

Постановка задачи

$X = \mathbb{R}^{20}$ $[x_{11}, x_{12}, ..., x_{1n}]$ $D_X = \mathcal{N}(0, 1)$ пространство входов

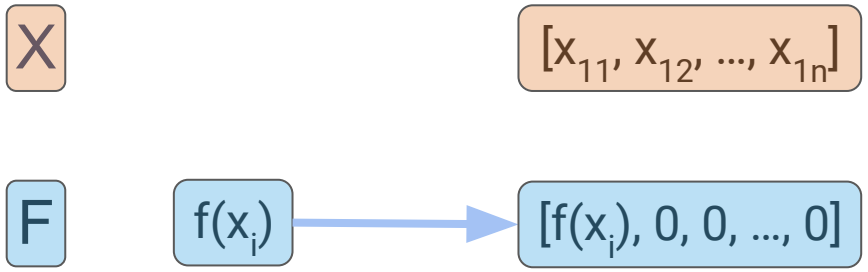
$F = \mathbb{R}^{20}$ $f(x_i)$ $D_F = \mathcal{N}(0, 1)$ пространство функций



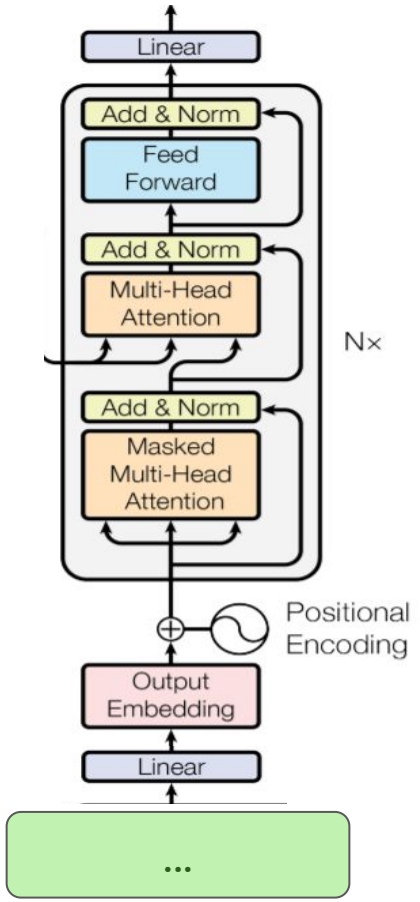
$$E_{x_{prompt}}[(LM(x_{prompt}) - f(x_q))^2] \leq \epsilon$$

Используемая модель

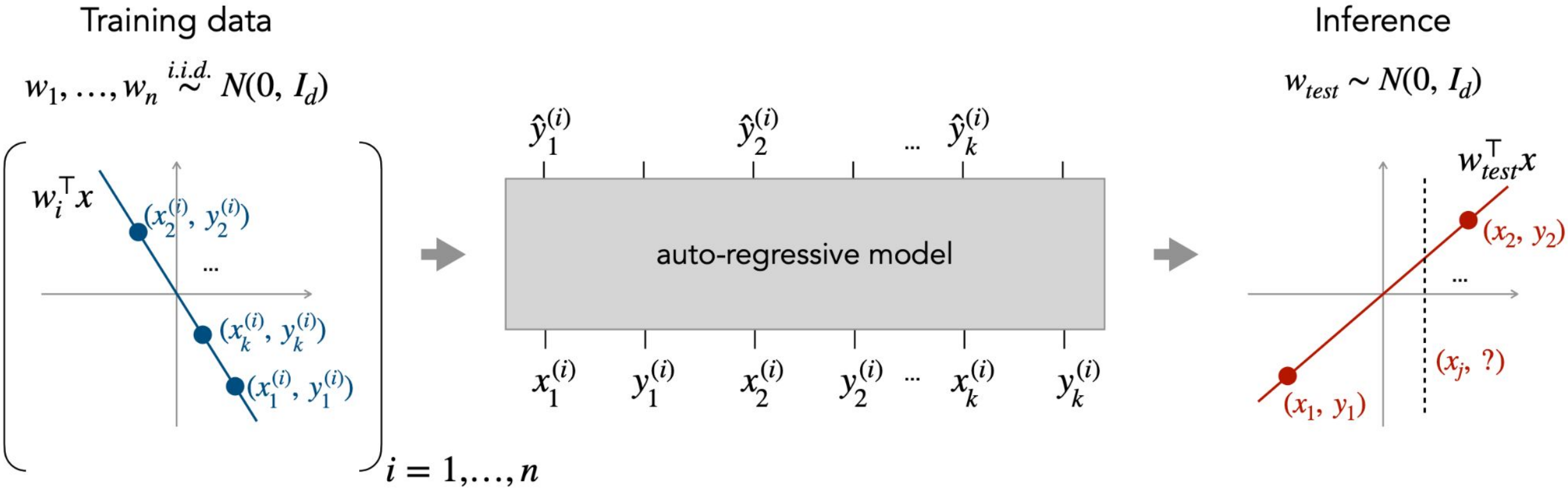
- Декодер архитектуры трансформера [Vaswani et al., 2017] из семейства GPT-2 [Radford et al., 2019]
- Переводим входы и выходы в одно пространство (нулями)
- Переводим элементы подсказки(x_{prompt}) в латентное пространство обучаемым линейным преобразованием.
- Переводим выходы модели из латентного пространства в скаляр обучаемым линейным преобразованием.



$f(x_q) \approx LM(x_{prompt})$



Постановка задачи

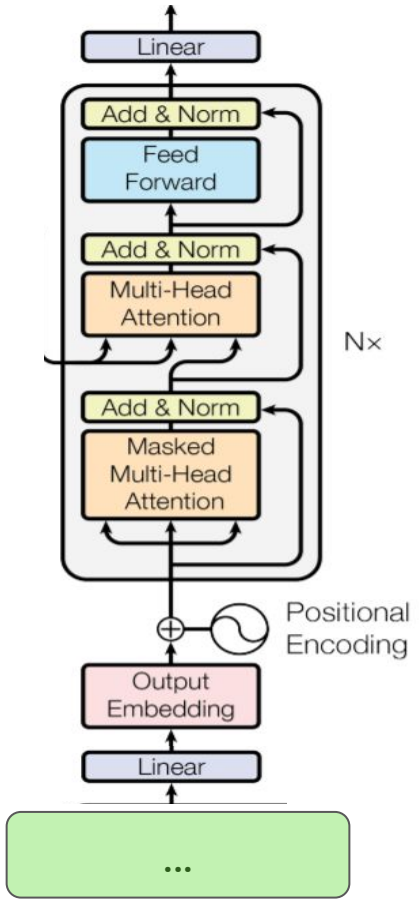


Изображение постановки задачи из статьи

Используемая модель

- 12 слоев, 8 голов, пространство 256, 22.4M параметров
- Обучение с нуля градиентным спуском
- Стартуют с простых функций, постепенно расширяясь до всех

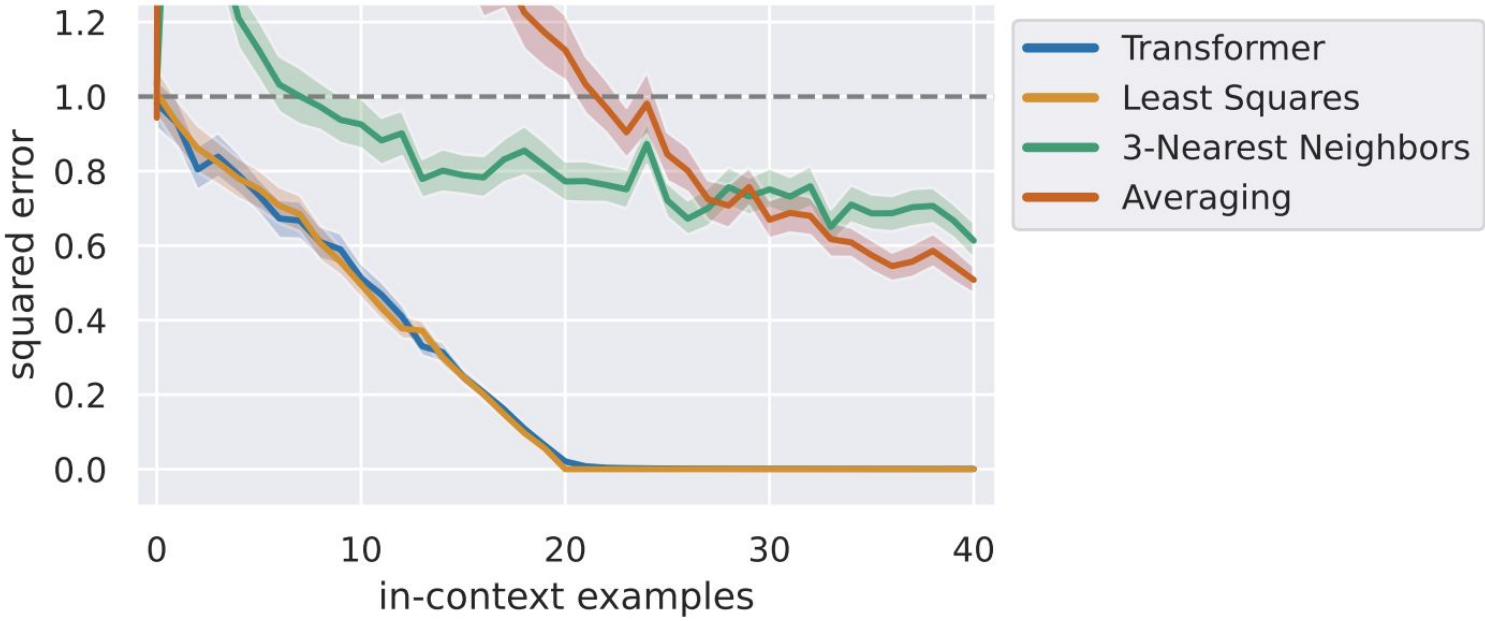
$f(x_q) \approx LM(x_{prompt})$



Сравнение качества

В качестве бейзлайнов берутся

- Метод наименьших квадратов
- 3NN
- Оценка w через mean($f(x_i)x_i$)



Качество моделей от длины подсказки

Качество оптимальное. Но нашей целью было объяснить in-context learning, а не решить классическую задачу

Можно ли было запомнить обучающую выборку?

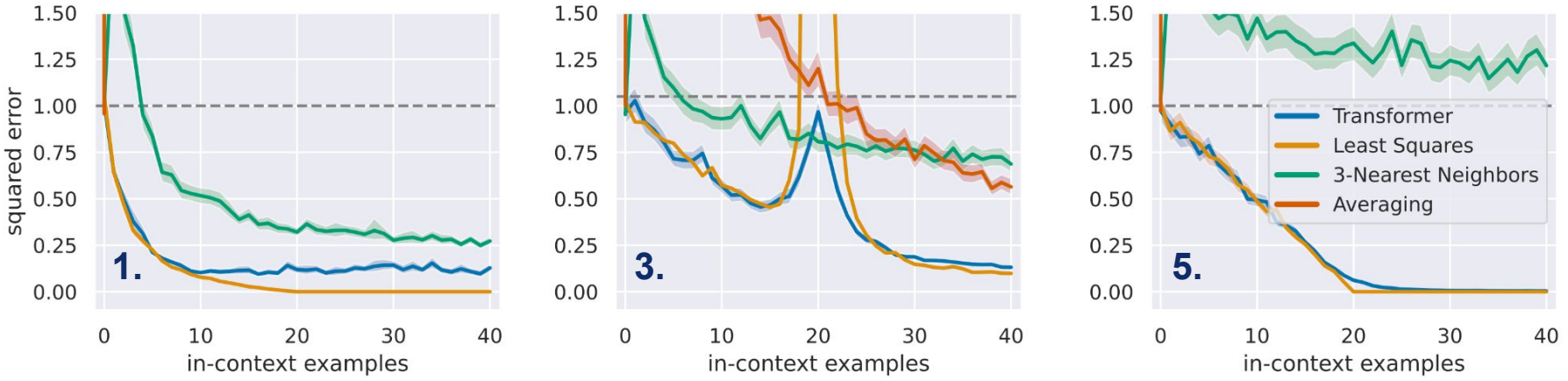
- В обучающей выборке вектора находятся в 800-мерном пространстве(20 входных и 20 выходных векторов размерности 20).
- В обучении использовалось 32M примеров.
- Эмпирическими оценками показано, что ошибка при использовании ближайшего примера составляла бы 0.2 (модель же дает 0.001 при длине подсказки 40)
- Также показано, что обучении на 10’000 примерах можно достичь такой же ошибки с моделью, в то время как запоминание даст уже ошибку в 0.5

Вывод: просто запоминания трейна недостаточно

- Почему модель обязана запоминать именно так?

Усложнение задачи

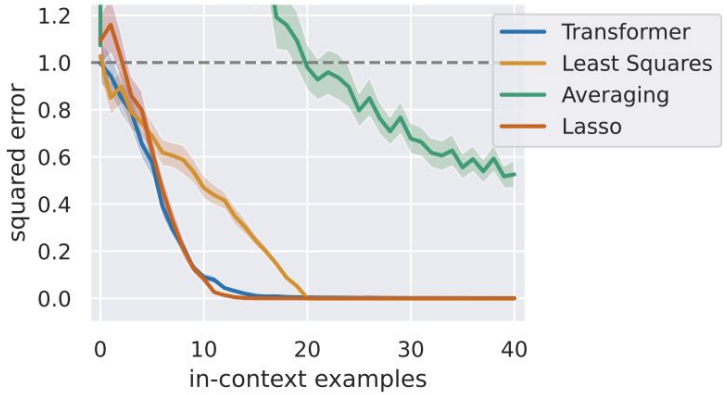
1. Искаженная ковариация $D_X = \mathcal{N}(0, \Sigma)$
2. Примеры в обучении из подпространства размера d/2 (d=20)
3. Зашумление обучения $f(x_i) + \mathcal{N}(0, 1)$
4. Масштабирование подсказок с коэффициентом 2 или 3
5. Разные ортанты для обучения и применения



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

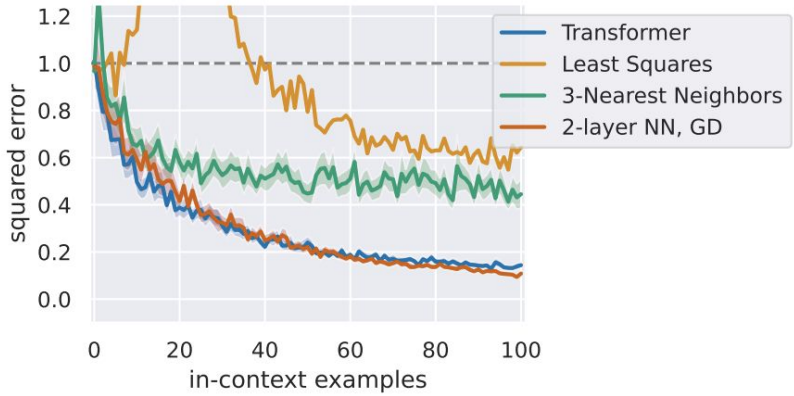
Другие домены функций

1. Разреженные линейные функции
2. Решающие деревья
3. Двухслойные ReLU NN

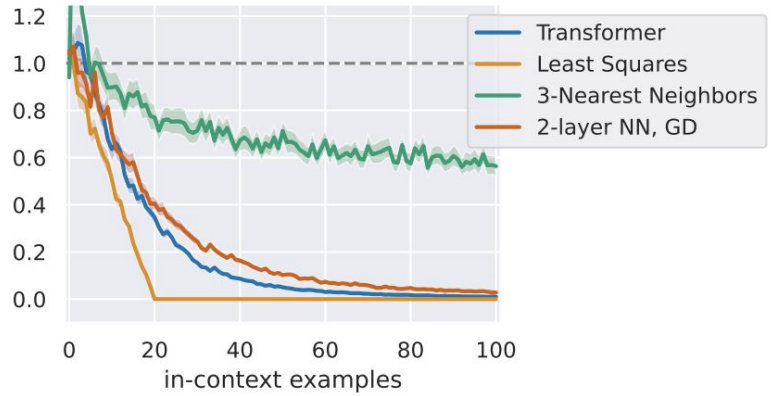


(a) Sparse linear functions

(b) Decision trees



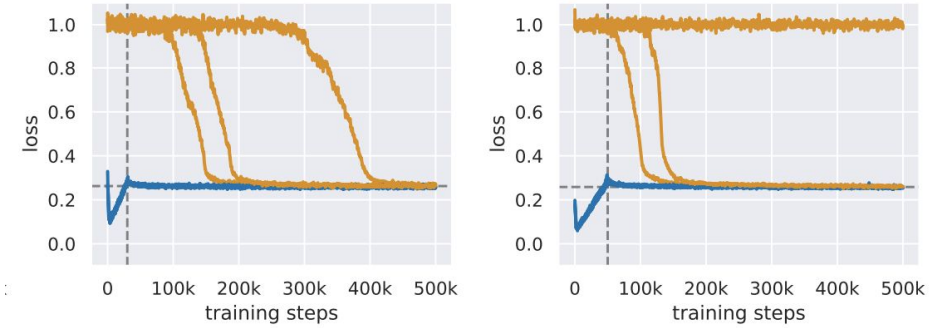
(c) 2-layer NN



(d) 2-layer NN, eval on linear functions

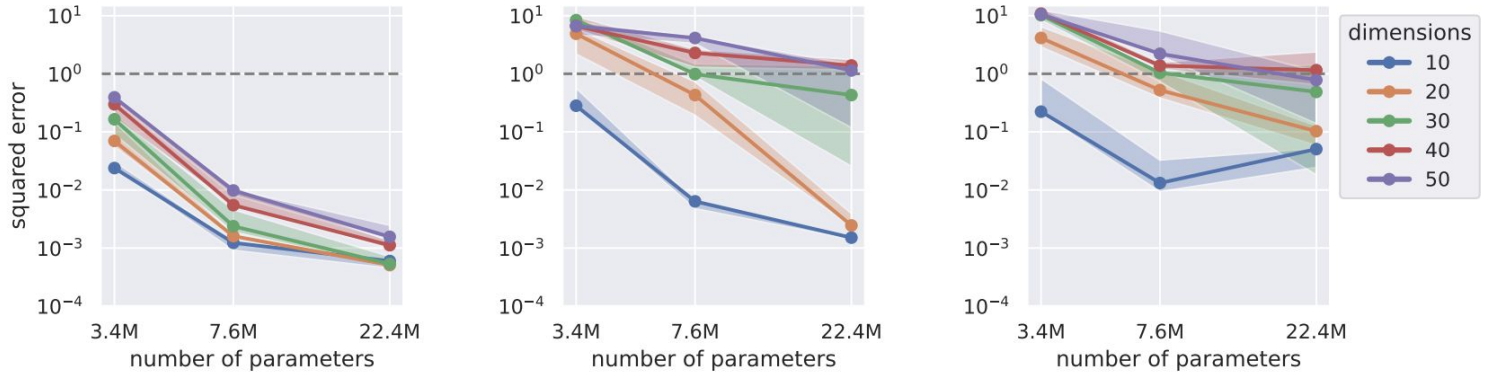
Важные детали обучения

1. Без специального планомерного обучения(curriculum) модель долго не обучается, после чего происходит резкий скачок в обучении
2. Вместимость(размер) модели влияет на положительно качество
3. Иногда значительно хуже удается работать с большими размерностями векторов



(b) 20 dimensions (c) 30 dimensions

Желтые линии - скачок при обучении без плана



(a) Standard (b) Different orthants (c) Skewed covariance

Перебор размерности входных векторов и размера модели для разных подходов

Вопросы

- Можно ли как-то связать с языковыми моделями?
- Что будет, если использовать дискретные входы?
- Не исследованы внутренности модели, трансформер использовался как черный ящик.
- Авторы, в отличие от языковых моделей явно обучаются на задачу.

Ваши вопросы?

Ссылка на статью

- What Can Transformers Learn In-Context? https://arxiv.org/abs/2208.01066
- Language Models are Few-Shot Learners https://arxiv.org/abs/2005.14165
- Attention Is All You Need https://arxiv.org/abs/1706.03762