How do transformers work? (Part II)

Daniel Hsu (Columbia) and Ankur Moitra (MIT)

Simons Bootcamp, September 4th

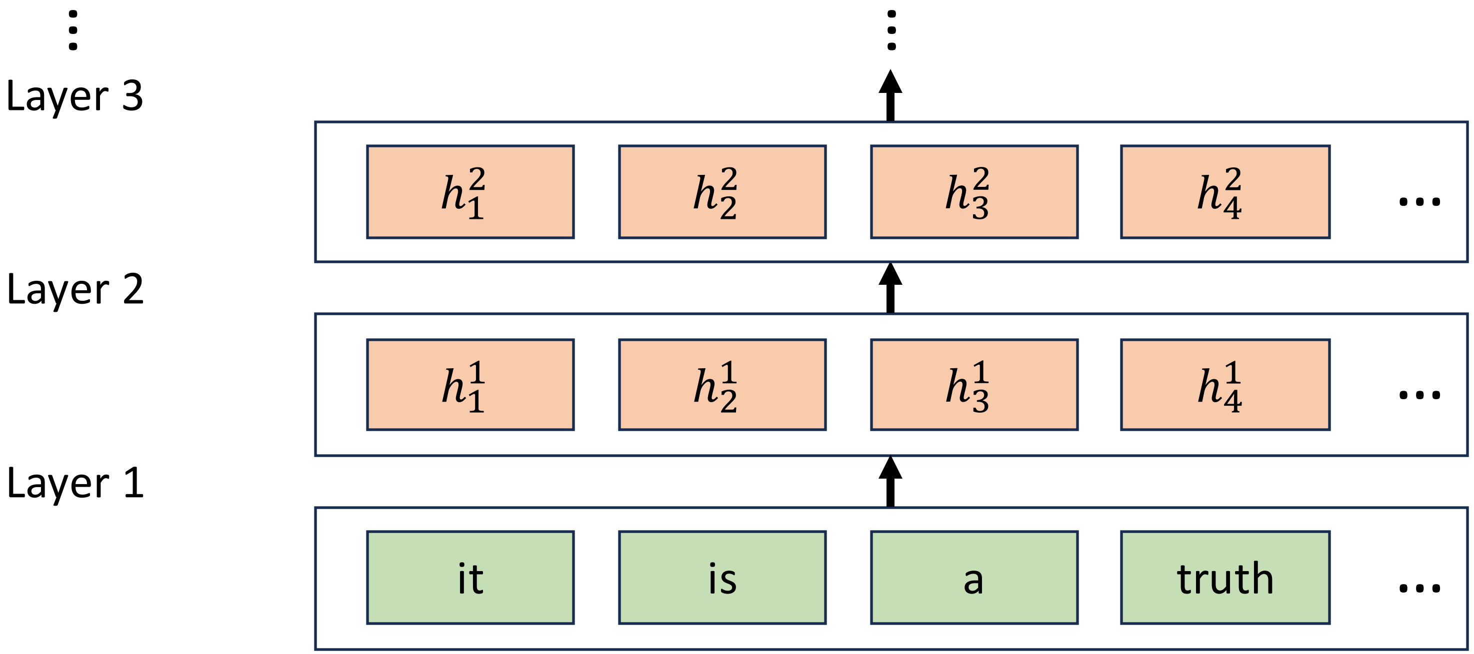
Plan for Part II

1. What can transformers do?
2. Overview of some theoretical perspectives

1. What can transformers do?

TRANSFORMERS

Transforms sequence of N tokens to sequence of N vectors by composing several sequence-to-sequence maps



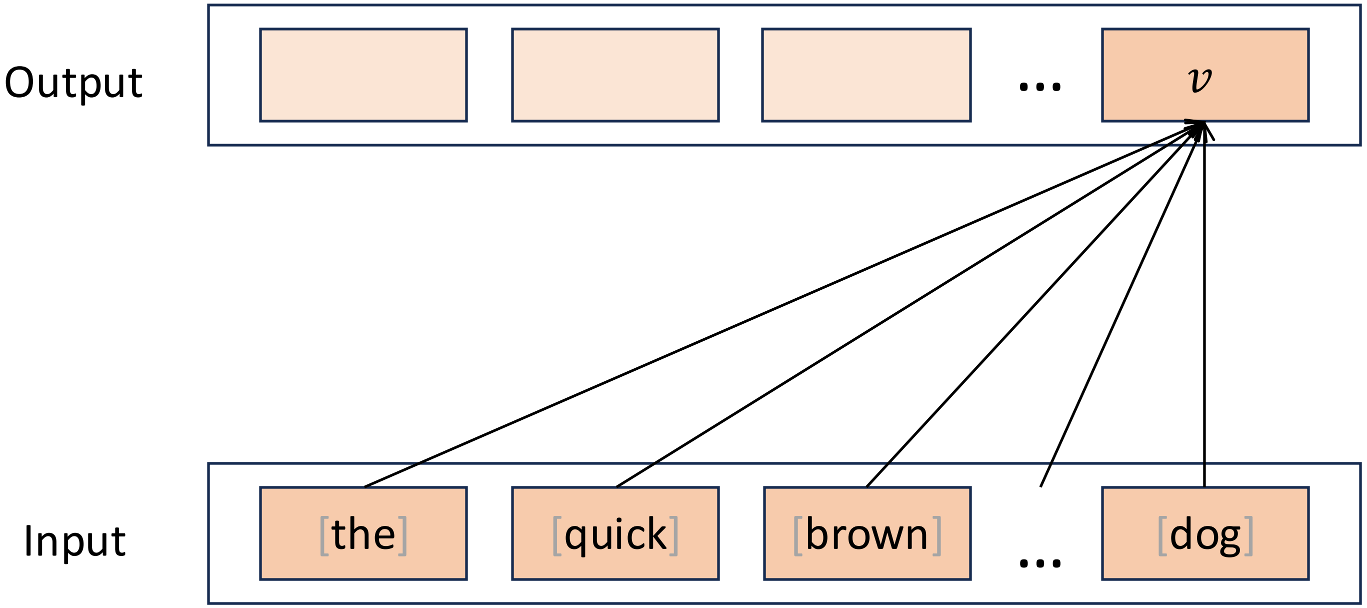
SINGLE QUERY ATTENTION

Given a query q, and keys and values for previous words compute

$$v = \sum_t \alpha_t v_t \quad \text{where} \quad \alpha_t = \frac{\exp(q^T k_t/\sqrt{d})}{\sum_u \exp(q^T k_u/\sqrt{d})}$$

weighted average of other values

weights are given by softmax



ATTENTION PATTERNS

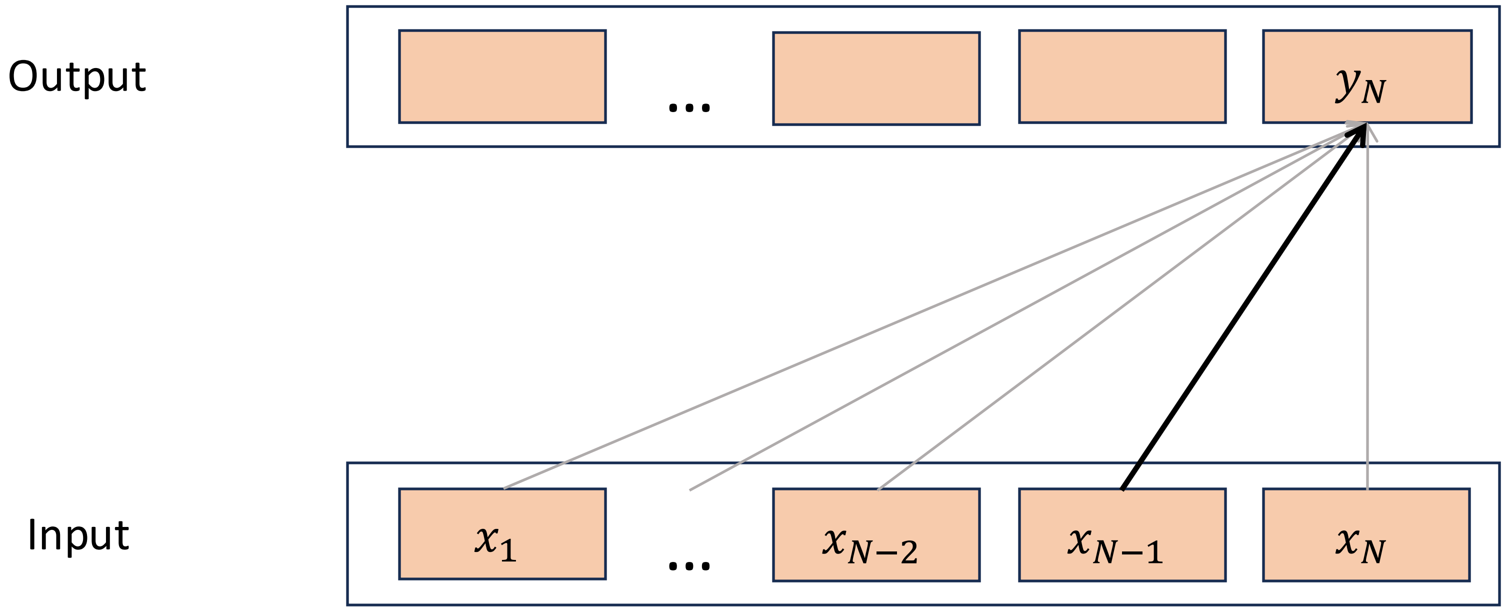
1. Query aligns with only a few keys
   → sparse weighted average of values

2. Query equally (mis)aligned with all previous keys
   → uniform average all previous values

How might these patterns arise?

EXAMPLE: POSITIONAL PATTERN

Query aligns only with previous token's key

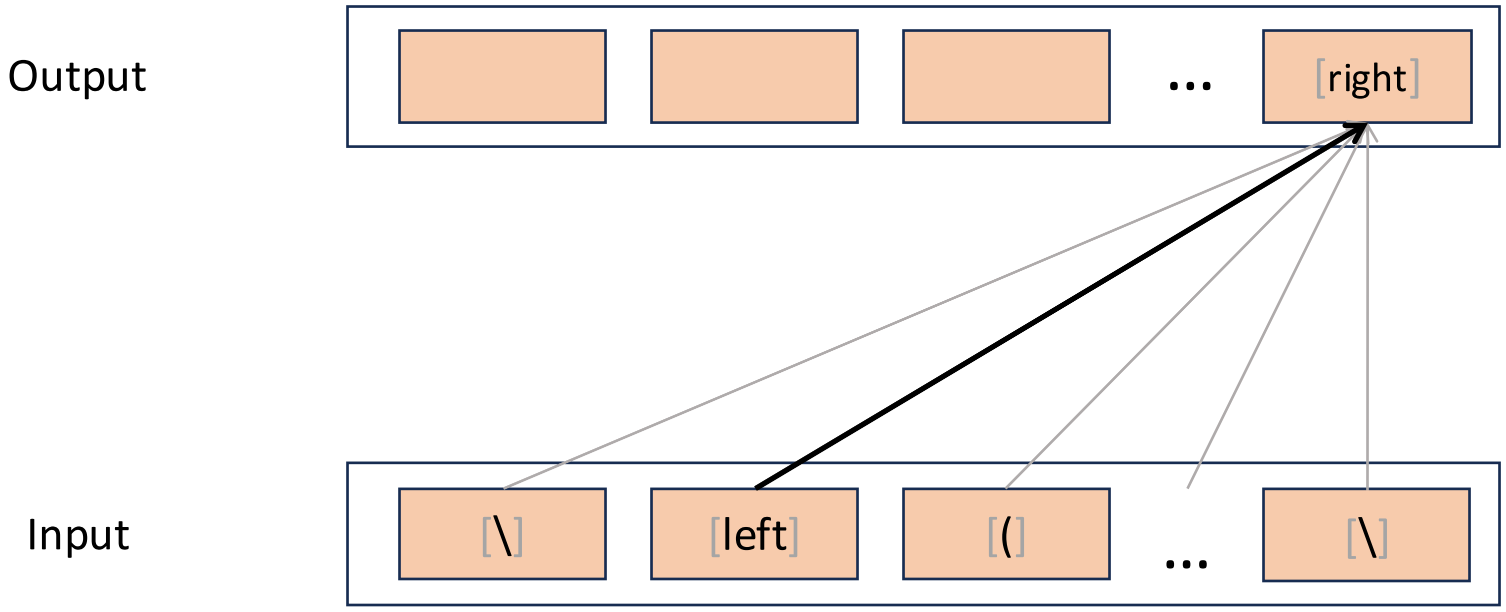


(Recall: input vectors = word embeddings + positional embeddings)

EXAMPLE: SKIP-GRAM PATTERN

[Elhage et al, 2021]

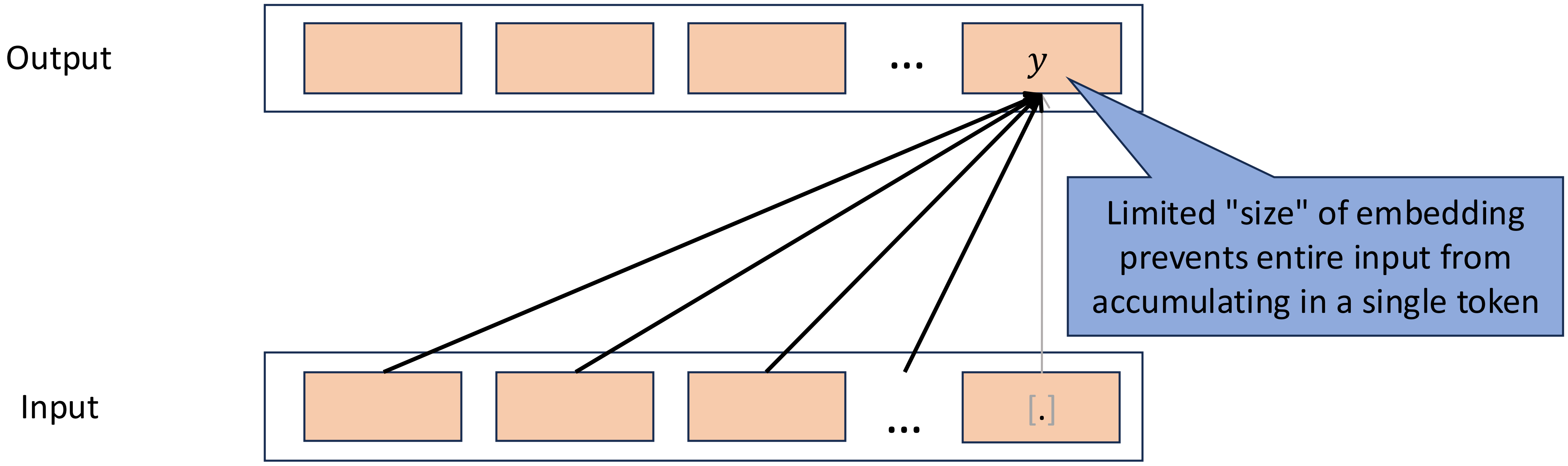
Query for "\" token aligns with key of "left"



Training identifies "skip-grams"---e.g., ("left", "\")---that help predict next token

EXAMPLE: AGGREGATION PATTERN

Query for "." (period) token aligns with keys of all previous tokens



What information gets passed up the layers?

EXAMPLE: INDUCTION HEADS

[Elhage et al, 2021; Olsson et al, 2022]

Prompt (after tokenization):

[Mr] [and] [Mrs] [Durs] [ley] [,] [of] [number] [four] [,] [Pri] [vet] [Drive] [,] [were] [proud] [to] [say] [that] [they] [were] [perfectly] [normal] [,] [thank] [you] [very] [much] [.] [They] [were] [the] [last] [people] [you] ['d] [expect] [to] [be] [involved] [in] [anything] [strange] [or] [mysterious] [,] [because] [they] [just] [didn] ['t] [hold] [with] [such] [nonsense] [.] [Mr] [Durs]

EXAMPLE: INDUCTION HEADS

[Elhage et al, 2021; Olsson et al, 2022]

Prompt (after tokenization):

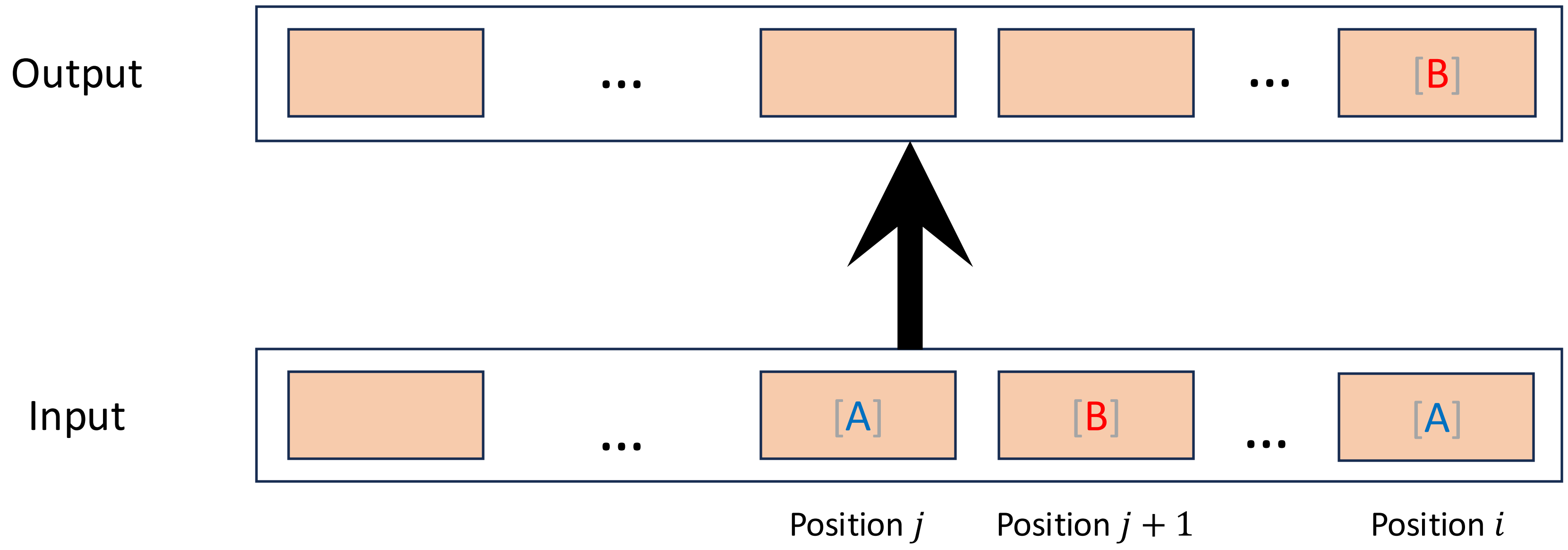
[Mr] [and] [Mrs] [Durs] [ley] [,] [of] [number] [four] [,] [Pri] [vet] [Drive] [,] [were] [proud] [to] [say] [that] [they] [were] [perfectly] [normal] [,] [thank] [you] [very] [much] [.] [They] [were] [the] [last] [people] [you] ['d] [expect] [to] [be] [involved] [in] [anything] [strange] [or] [mysterious] [,] [because] [they] [just] [didn] ['t] [hold] [with] [such] [nonsense] [.] [Mr] [Durs]

INDUCTION HEADS ABSTRACTION

[Elhage et al, 2021; Olsson et al, 2022]

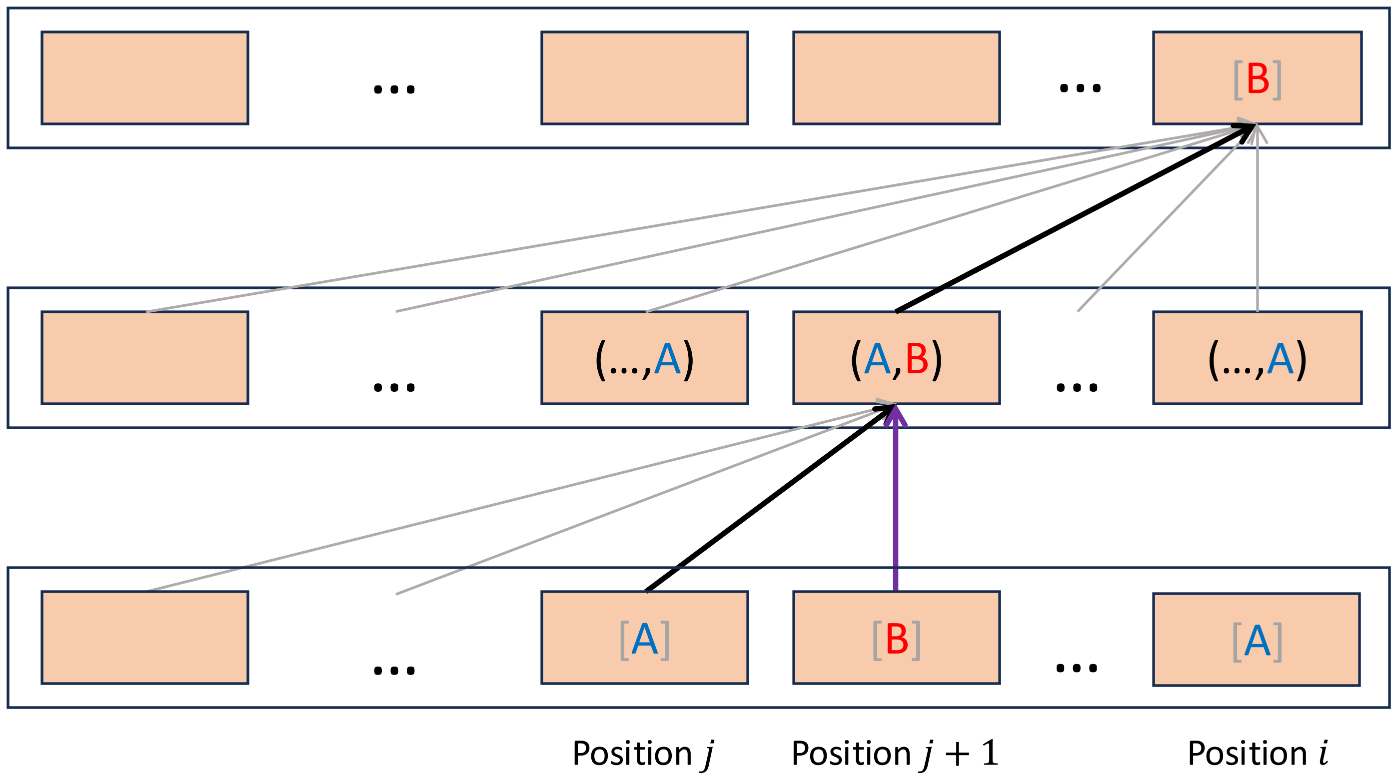
Induction head: abstraction of a salient sub-circuit found in LLMs

- i^{th} output: Find latest time $j < i$ that x_i occurs, output x_{j+1}



INDUCTION HEADS IMPLEMENTATION

Composition of two self-attention heads



Layer 2: find $\langle k, q \rangle$ match

Notation: (KEY, QUERY/VALUE)

Layer 1: move prev. token's key (+ use "skip connection")

Input to induction head

IN-CONTEXT LEARNING [Brown et al, 2020]

Circulation revenue has increased by 5% in Finland. // Positive

Panostaja did not disclose the purchase price. // Neutral

Paying off the national debt will be extremely painful. // Negative

The company anticipated its operating profit to improve. // ________

LM

Circulation revenue has increased by 5% in Finland. // Finance

They defeated … in the NFC Championship Game. // Sports

Apple … development of in-house chips. // Tech

The company anticipated its operating profit to improve. // ________

LM

[Figure from Xie and Min, 2022]

IN-CONTEXT LEARNING VIA INDUCTION HEADS

Prompt:

The mother of Charlotte is Eve. The mother of John is Helen. [...] Who is John's mother?

Sequence after some processing by a few transformer layers (perhaps):

... [Charlotte] [Eve] ... [John] [Helen] ... [John]

"In-context learning" / "meta-learning" / nearest neighbor prediction

E.g., in-context learning n-gram models: [Edelman, Edelman, Goel, Malach, Tsilivis, 2024]

Also Tengyu's talk this afternoon

FUNCTION COMPOSITION

[Peng, Narayanan, Papadimitriou, 2024; Sanford, Hsu, Telgarsky, 2024]

Prompt:

Jane is a teacher. Helen is a doctor. […] The mother of Charlotte is Eve. The mother of John is Helen. […] What is the profession of John's mother?

Function composition = iterated induction head

What are the key primitives in LLMs, and how are they put together?

2. Some theoretical perspectives

SOME (MORE) THEORETICAL PERSPECTIVES

- Transformer as a formal model of computation
- Learning and Chain-of-Thought
- Prediction vs generation
- Associative memories

TRANSFORMER AS FORMAL MODEL OF COMPUTATION

[Liu, Ash, Goel, Krishnamurthy, Zhang, 2023; Merrill & Sabharwal, 2023; Strobl, 2023]

- $O(1)$-layer $\mathrm{poly}(N)$-size transformers ⊆ (Uniform) TC^0
    - Implications: e.g., cannot simulate all finite automata (unless $TC^0 = NC^1$)

[Hahn, 2020; Hao, Angluin, Frank, 2022; Angluin, Chiang, Yang, 2023; …]

- Restrictions on "softmax" and/or masking further limit expressivity

[Sanford, Hsu, Telgarsky, 2024]

- Simulation of/by Massively Parallel Computation algorithms
    - Lower bounds for induction heads and other primitives

What abstraction is relevant for transformers at practical scales?

LEARNING IN PRACTICE

- Transformer maps context (e.g., "the quick brown fox jumped over the lazy") to vector h, which is used in a **log-linear model** $P_\theta(\text{next word} \mid h)$
- **Training:** Tune parameters θ = ((Q, K, V) matrices, feedforward nets, …, log-linear model) to minimize cross-entropy on training data

$$\sum_{t=1}^{T} -\log P_\theta(\text{word } t \mid \text{previous } t-1 \text{ words})$$

May truncate to last N words

- **Equivalent:**
  - Maximize likelihood of θ given data
  - Minimize relative entropy of empirical frequencies w.r.t. P_θ

LEARNING IN THEORY

[Edelman, Goel, Kakade, Zhang, 2022]

- If I manage to find an L-layer transformer with low training error, will its test error also be low?
- Probably YES if:
    - Training/test data are i.i.d. from same distribution over length-N sequences);
    - Token embeddings are computed by "nice" functions and are not too "large";
    - Training data size $\gtrsim \exp(L)\log(N)$

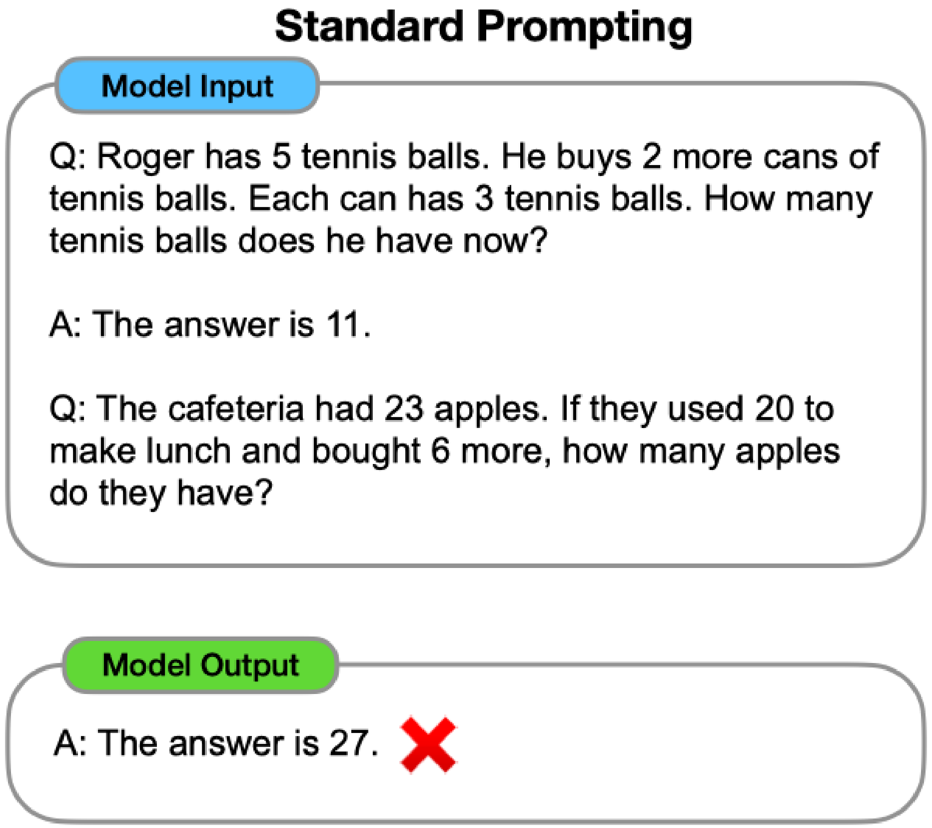
[Chen, Li, 2024; Oymak, Rawat, Soltanolkotabi, Thrampoulidis, 2023; Nichani, Damian, Lee, 2024; …]

- Can I efficiently find a low error transformer? With gradient descent?

Relevant notion of generalization for LLMs?

CHAIN-OF-THOUGHT (CoT)

[Wei, Wang, Schuurmans, Bosma, Xia, Chi, Le, Zhou, 2022; Kojima, Gu, Reid, Matsuo, Iwasawa, 2022; …]



[Figure from Wei et al, 2022]

BENEFITS OF CoT

1. Extra "work space" to compute prediction [Merrill & Sabharwal, 2024; …]
2. Extra "worked steps" available during training

Traditional labeled training example:

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Labeled training example with worked steps:

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

DOES CoT MAKE LEARNING EASIER?

Hard PAC learning problems (e.g., decision trees, DNFs, circuits) become easy with extra "worked steps" / "clues" during training

	Extra "worked steps" / "clues"
[Sloan & Rivest, 1988; Malach, 2023]	Values of all gates in circuit
[Dvir, Rao, Wigderson, Yehudayoff, 2012]	Randomly restricted access to circuit

Where do these "worked steps" come from?

GOALS OF LANGUAGE MODELING

Two roles of a language model $\hat{P}$:

1. **Prediction** **(what comes next?)**

$$\underset{\text{next word}}{\arg\max}\ \hat{P}(\text{next word}|\text{context})$$

2. **Generation** **(write new sentences)**

$$\text{next word} \sim \hat{P}(\cdot\,|\text{context})$$

PREDICTION VS GENERATION

[Kalai and Vempala, 2024]

Even in an "idealized" setting: for any trained language model $\hat{P}$,

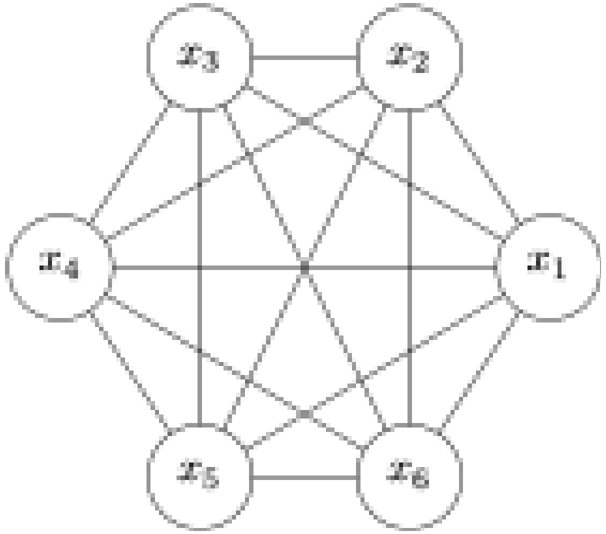
$$\text{Hallucination rate} \geq \widehat{\text{MF}} - \text{miscalibration} - \frac{300|\text{Facts}|}{|\text{Possible hallucinations}|} - \frac{7}{\sqrt{n}}$$

Number of facts seen only once in training / n
≈ "missing mass" of facts not seen in training

ASSOCIATIVE MEMORIES

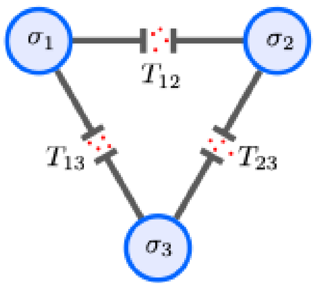
[Hopfield, 1982]

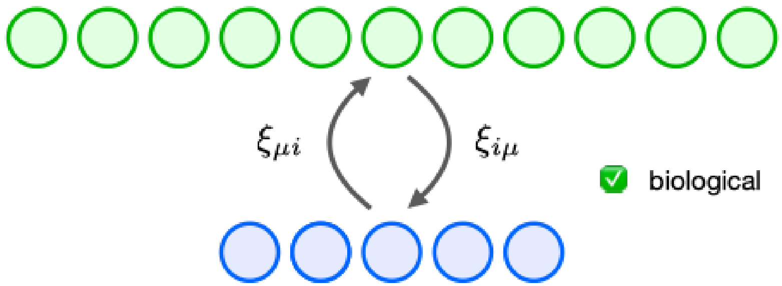
- Hopfield network: Each of d neurons is connected to all others
- State of neurons: $(x_1, \ldots, x_d) \in \{-1,1\}^d$
- How many (random) binary patterns can such a network memorize?



MODERN HOPFIELD NETWORKS

- Hopfield networks: d neurons can memorize $n \sim d$ binary patterns
- "Modern" Hopfield networks: $n \sim \exp(\Omega(d))$ [Demircigil et al, 2017; Ramsauer et al, 2021; Krotov & Hopfield, 2016, 2021]
  - One-step dynamics equivalent to self-attention mechanism in transformers





  - Continuous dynamics [Geshkovski, Letrouit, Polyanskiy, Rigollet, 2023]: related to interacting particle systems and models of opinion dynamics

Implications for capabilities of transformers?

[Figures from Krotov & Hopfield, 2021]

CLOSING

This tutorial:
+ How do transformers work?
+ Some theoretical perspectives

Open question: Which ingredients are essential?

Thank you! Any questions?