Transformers, parallelism, and the role of depth

Daniel Hsu
Columbia University

Math and Data Seminar, NYU

April 3, 2025

Capabilities of large language models?

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. // ________

[Figure from Xie and Min, 2022]

Multi-step reasoning
[Weston, Chorpa, Bordes, 2014]

John is in the playground.
Helen is playing with John.
Helen picked up the football.
Where is the football?

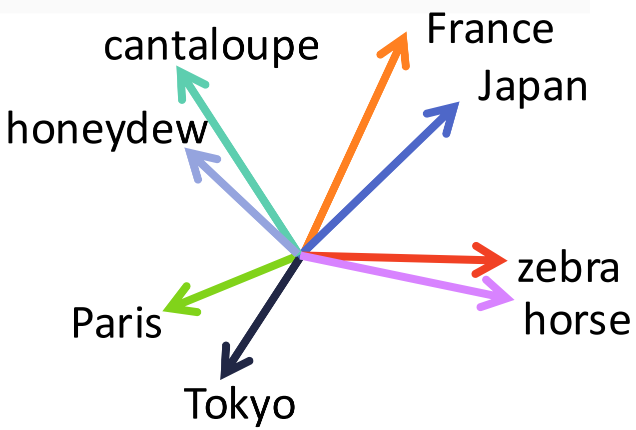
Plan for the talk

1. Role of depth in transformers
2. Transformers & Massively Parallel Computation
3. Limitations of sequential neural architectures (if time permits)

Joint work with:
Clayton Sanford (Columbia ⟶ Google Research)
Matus Telgarsky (NYU)
[NeurIPS 2023, ICML 2024, arXiv:2408.14332]

0. Basics about transformers

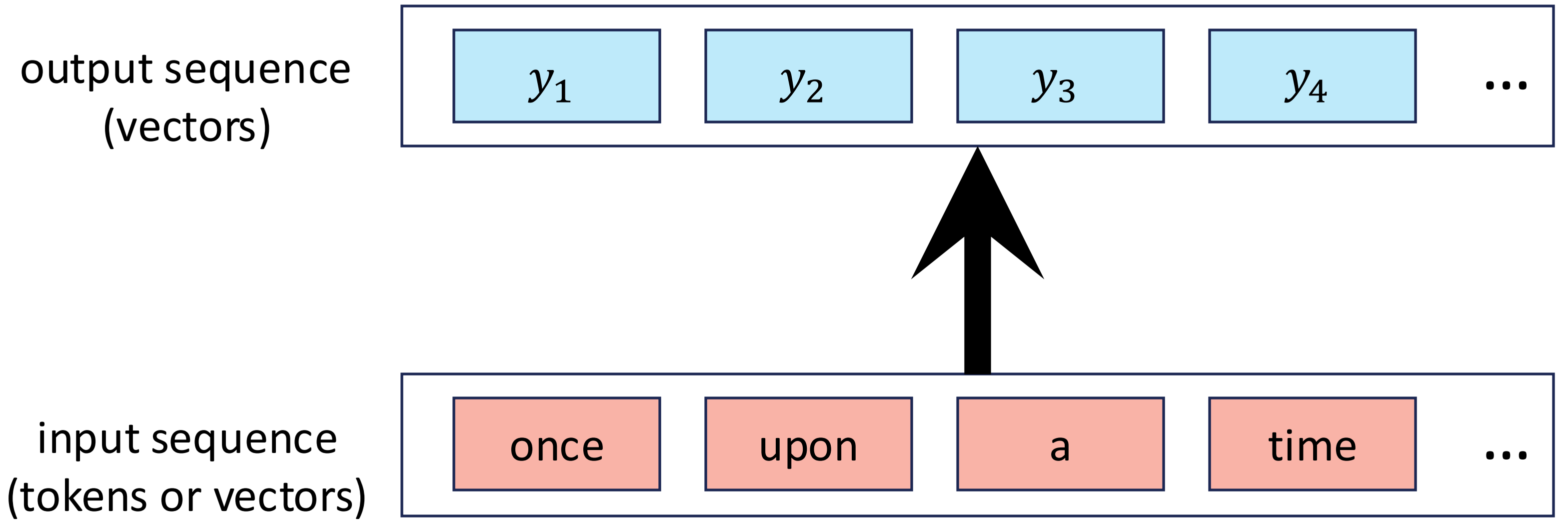
Transformers [Demircigil et al, 2017; Vaswani et al, 2017]



Transformer: a kind of sequence-to-sequence map, formed by compositions of self-attention heads

Ingredients:

1. Ways to embed tokens into vector space
2. Way to for embedded tokens to "interact" and produce new vectors

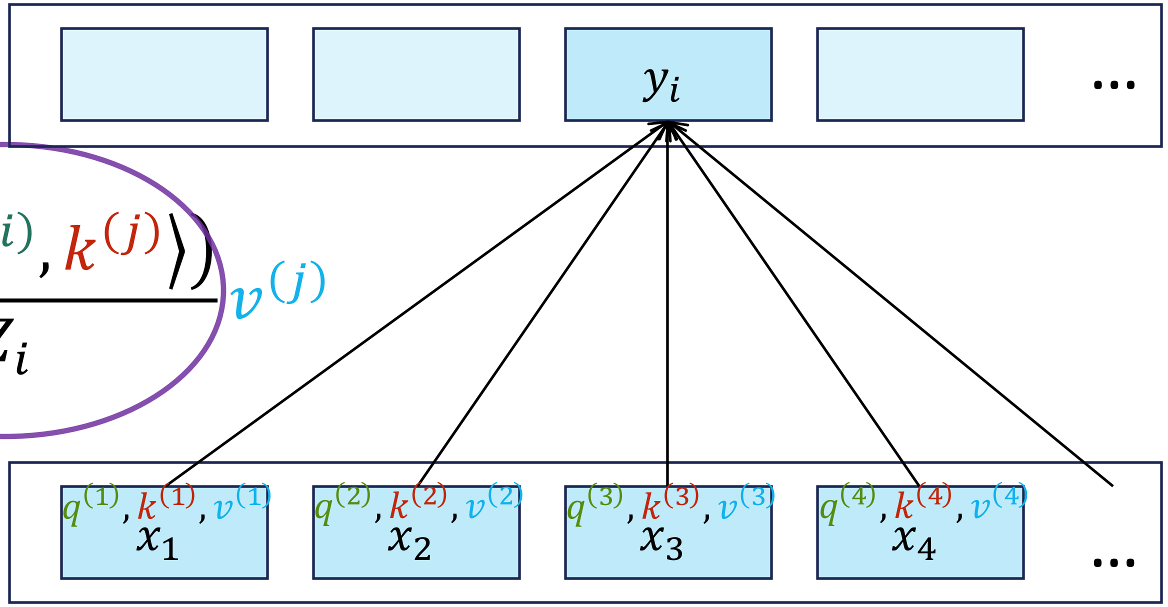


Self-attention head

Token embeddings produced using "trained" multilayer Perceptrons (MLPs)

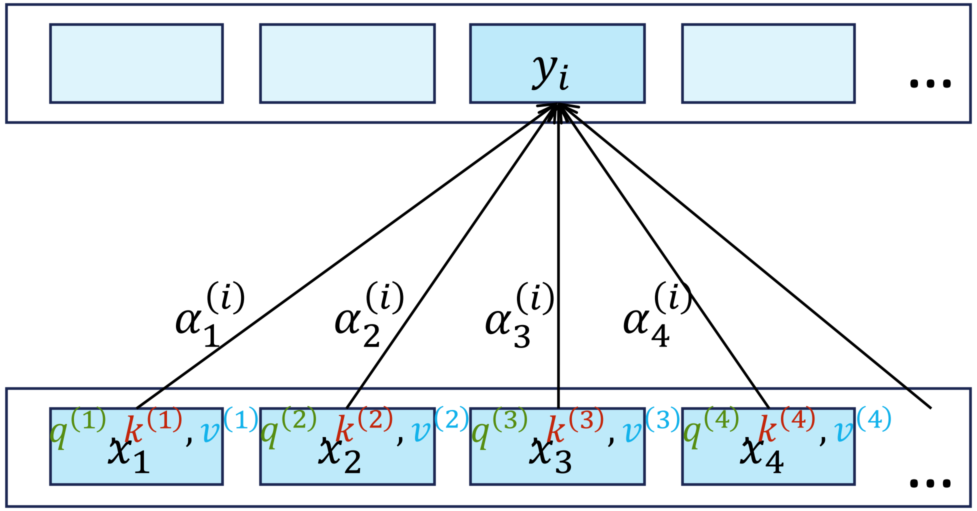
1. Independently create N query/key/value vectors from $x_1, \dots, x_N$
2. For each $i \in [N]$: i^{th} output y_i = weighted average of all N values, where weights = "softmax" of $\langle i^{\text{th}}$ query, j^{th} key$\rangle$ for all $j \in [N]$

$$y_i = \sum_j \frac{\exp(\langle q^{(i)}, k^{(j)} \rangle)}{Z_i} v^{(j)}$$



Outputs $y_1, \dots, y_N$ can be produced in parallel

Comparison to feedforward neural networks



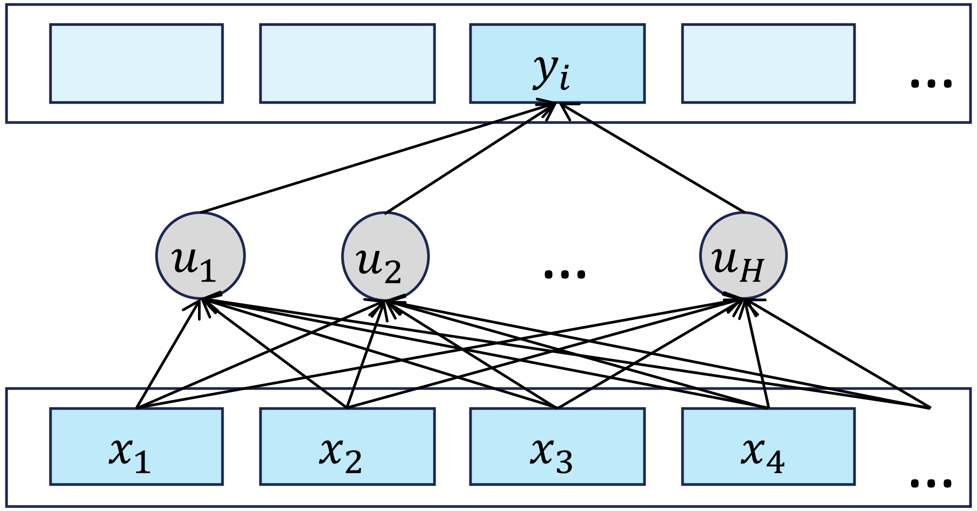
Self-attention head

Shared parameterized mapping

$$x_i \mapsto \left(q^{(i)}, k^{(i)}, v^{(i)}\right)$$

Weights $\alpha_j^{(i)}$ determined via softmax

Universal approximation if embedding dimension $D \to \infty$



Feedforward neural network

Each "weight" is a separate parameter

$$y_i = \sum_{j=1}^{H} A_{i,j}\,\sigma\left(\sum_{k=1}^{N} W_{j,k} x_k\right)$$

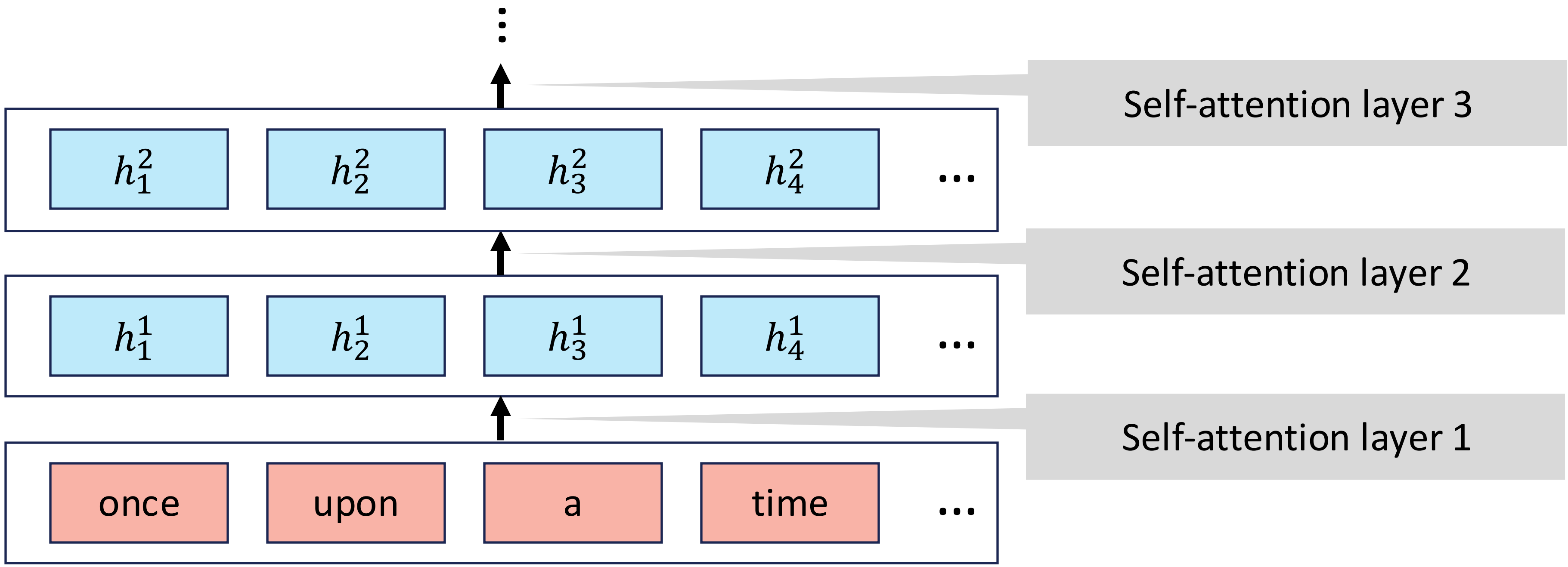
Universal Approximation Bounds for Superpositions of a Sigmoidal Function

Andrew R. Barron, *Member, IEEE* (if width $H \to \infty$)

Transformers as compositions

Transformers: compositions of self-attention layers

(layer = one self-attention head, or sum of several self-attention heads)



Why are multiple layers necessary?

1. Role of depth in transformers

Tasks for transformers

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. // ________

[Figure from Xie and Min, 2022]

Multi-step reasoning
[Weston, Chorpa, Bordes, 2014]

John is in the playground.
Helen is playing with John.
Helen picked up the football.
Where is the football?

In-context learning as associative recall

Prompt: whale 1 dog 1 frog 0 shark 0 bat 1 owl 0 wolf

"Nearest neighbor"-like in-context learning

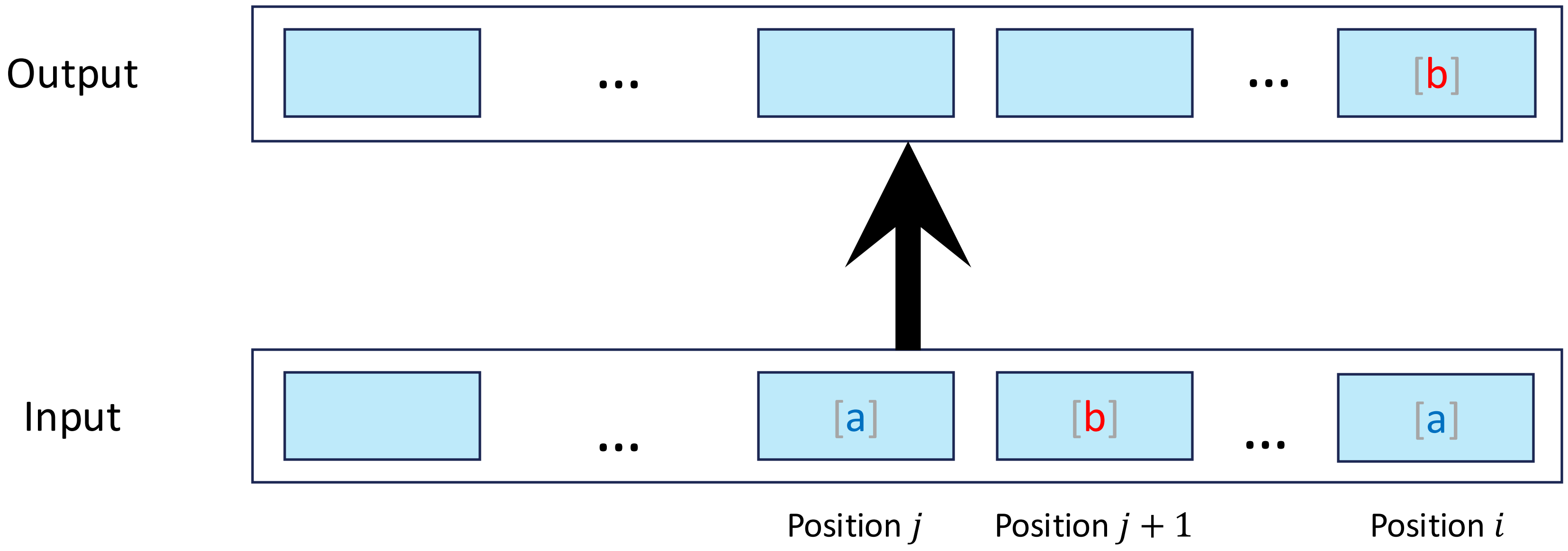
b a c b … c a b d b a

Associative recall task (a.k.a. induction heads task)

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

<u>(Most recent) associative recall task</u>:

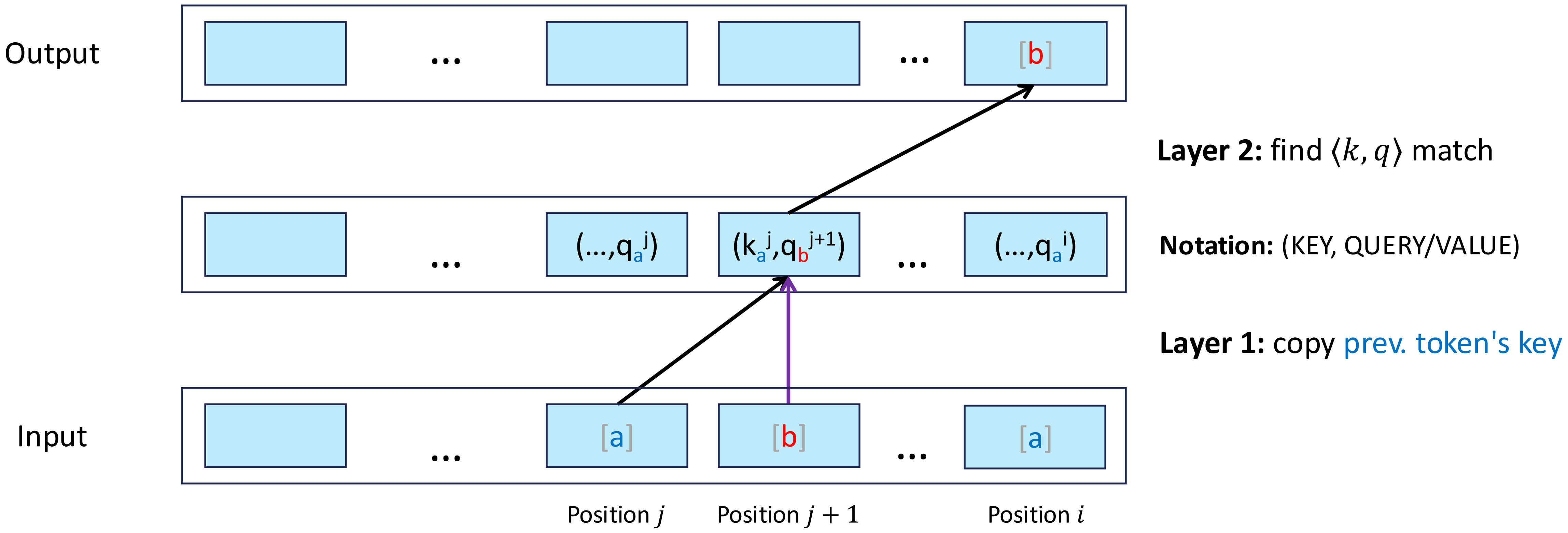
- i^{th} output: Find last position $j < i$ where x_i occurs, output x_{j+1}



Solution using two layer transformer

Composition of two "small" self-attention heads [e.g., Bietti et al, 2023]

Token embedding dimension $O(\log N)$ suffices



Necessity of two layers

Theorem [SHT'24b]:
Single self-attention head* (one layer) with embedding dimension D cannot compute associative recall for length N sequences unless

$$D \geq \tilde{\Omega}(N)$$

Exponentially larger than what's sufficient with *two* layers

Corroborates prior empirical findings
[Elhage et al, 2021; Olsson et al, 2022; Bietti et al, 2023]

*Using polylog N bits of numerical precision, even for $O(1)$-size input alphabet, allowing arbitrary size MLPs

Proof (by reduction from Index)

Index problem:

- Alice is given $(f_1, \ldots, f_T) \in \{0,1\}^T$
- Bob is given $i^* \in [T]$
- Goal: Message that Alice can send to Bob that lets Bob determine f_{i^*}

Lower bound (by counting): Alice must send T bits

Claim:

Self-attention head for Associative Recall (for N token seqs.) with embedding dim. D	$\Rightarrow$	$\tilde{O}(D)$ bit messaging strategy for Index (for $T = \Omega(N)$)

Proof of claim

- Index instance $(f_1, \dots, f_T, i^*) \mapsto$ Associative Recall instance (over alphabet $\{0,1,?,\perp\}$)

$$(x_1, x_2, \dots, x_N) = (e_1, f_1, e_2, f_2, \dots, e_T, f_T, ?)$$

where $N = 2T + 1$ and

$$e_i = \begin{cases} ?, & \text{if } i = i^* \\ \perp, & \text{if } i \neq i^* \end{cases}$$

- N^{th} output y_N of a self-attention head for Associative Recall must encode f_{i^*}:

Alice can send $O(D \log N)$ bit message to Bob that lets Bob evaluate y_N

$$y_N = \frac{\sum_{i=1}^{N} e^{\langle q^{(N)}, k^{(i)} \rangle} v^{(i)}}{\sum_{i=1}^{N} e^{\langle q^{(N)}, k^{(i)} \rangle}}$$

Known to Alice

$$= \frac{\sum_{j=1}^{T} e^{\langle q^{(N)}, k^{(2j-1)} \rangle} v^{(2j-1)} + \sum_{j=1}^{T} e^{\langle q^{(N)}, k^{(2j)} \rangle} v^{(2j)} + e^{\langle q^{(N)}, k^{(N)} \rangle} v^{(N)}}{\sum_{j=1}^{T} e^{\langle q^{(N)}, k^{(2j-1)} \rangle} + \sum_{j=1}^{T} e^{\langle q^{(N)}, k^{(2j)} \rangle} + e^{\langle q^{(N)}, k^{(N)} \rangle}}$$

Known to Bob

Beyond two layers?

Multi-step reasoning [Weston, Chorpa, Bordes, 2014; Peng, Narayanan, Papadimitriou, 2024]:

Prompt:

Jane is a teacher. Helen is a doctor. [...]

The mother of John is Helen. The mother of Charlotte is Eve. [...]

What's the profession of John's mother?"

Answer: doctor

b a c b c a b d b a

2-hop induction head

k-hop induction head

Theorem [SHT'24a]:
There is a $2 + \lceil \log_2 k \rceil$ layer transformer* that implements k-hop …

<u>Main idea</u>: Each additional layer *doubles* the "reach"
(Cf. [Liu, Ash, Goel, Krishnamurthy, Zhang, 2023] simulating finite automata)

… & under plausible conjecture about **massively parallel computation**,
$\Omega(\log k)$ layers are necessary (under similar size constraints)

*Using one self-attention head per layer, $\log N$ dimensional embeddings, $\log N$ bits of numerical precision, assuming $\mathrm{poly}(N)$-size input alphabet, causal masking

2. Transformers & Massively Parallel Computation

Massively Parallel Computation (MPC)

MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

Google, Inc.

A Model of Computation for MapReduce

Howard Karloff* Siddharth Suri† Sergei Vassilvitskii‡

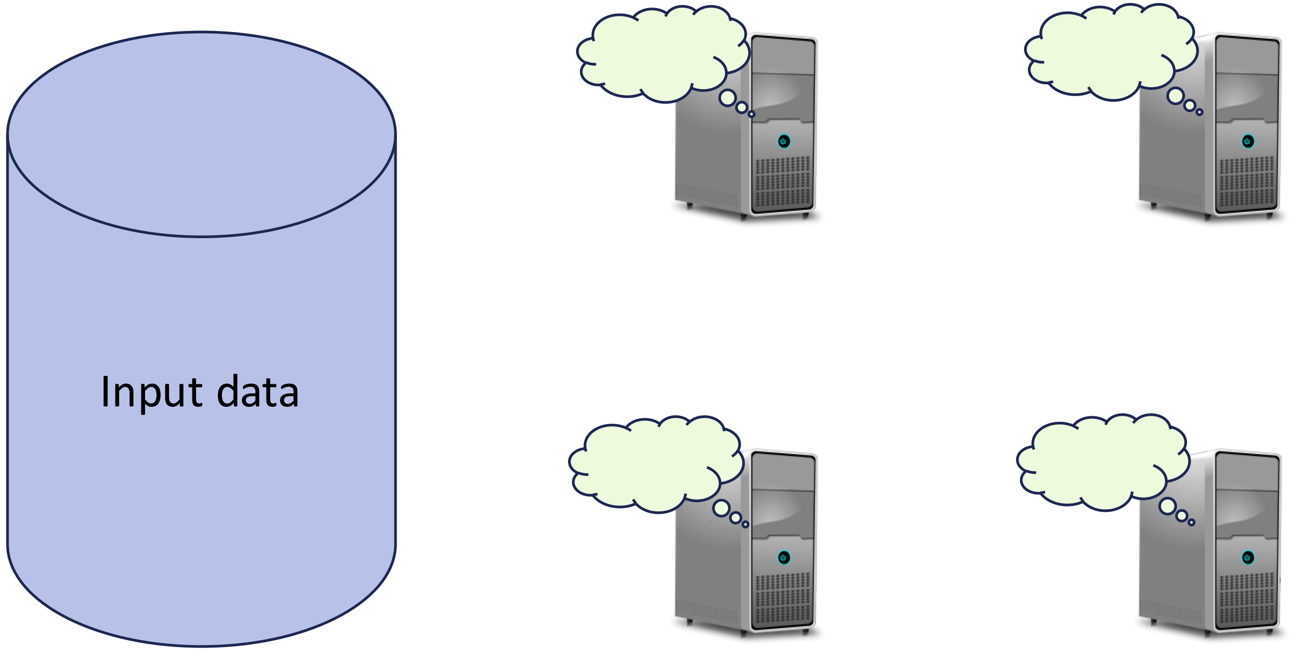
[Karloff et al, 2010; Goodrich et al, 2011; Beame et al, 2013; Andoni et al, 2014]

MPC model of computation

Input data size: N words [$N \leq M \times S$]

Number of machines: M

Memory size per machine: S words [$S = \Theta(N^{\delta})$ for small $\delta \in (0,1)$]



Communication constraints per "shuffle" round:

Each machine sends $\leq S$ words

Each machine receives $\leq S$ words

Between "shuffle" rounds:

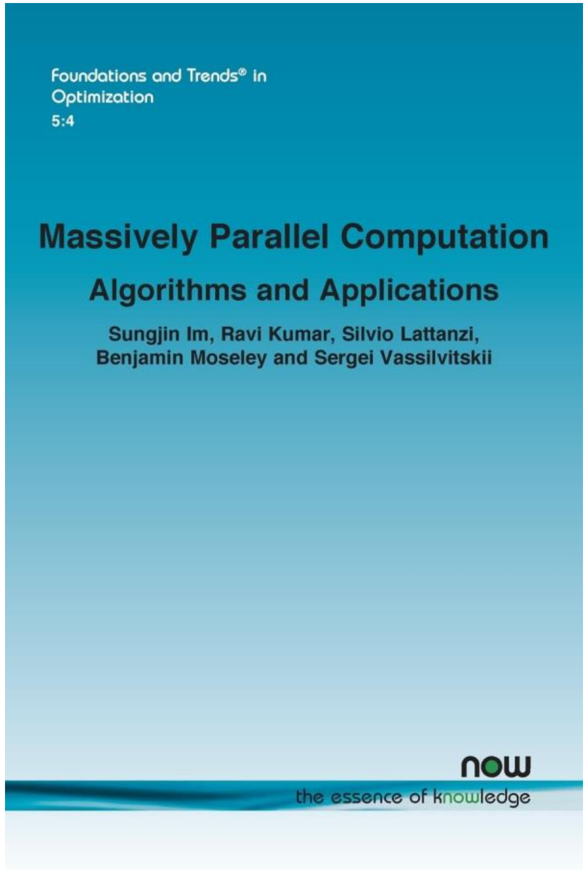
Each machine performs arbitrary computation on local memory

Main question: How many rounds R are needed?

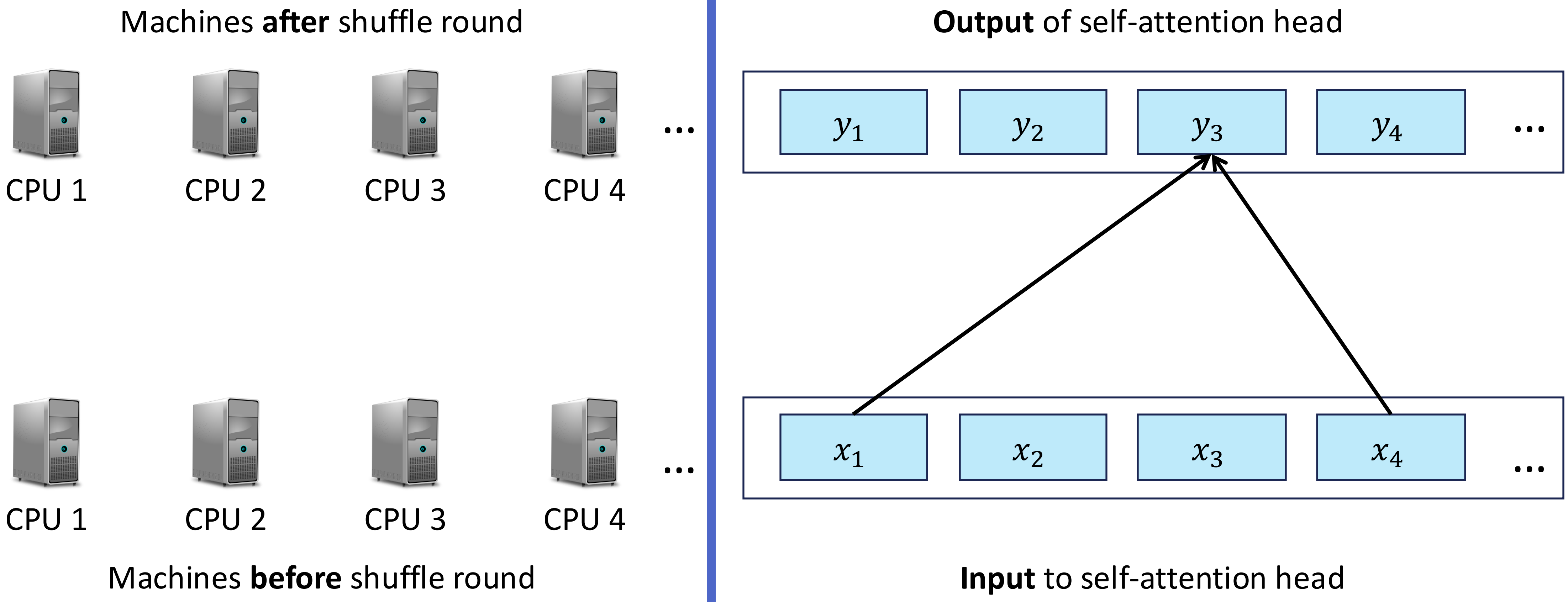
MPC algorithms for many tasks

- Broadcast $R = O(1)$
- Sorting $R = O(1)$
- Prefix sum $R = O(1)$
- …

- Open question:
  $R = o(\log N)$ rounds for graph connectivity?



Simulating MPC shuffle round with self-attention



MPC algorithms ⇄ transformers

Theorem [SHT'24a; Sanford et al, 2024] (informal version):

can be simulated by*

$\Theta(R)$-round MPC algorithm with local memory $\approx \Theta(N^{\delta})$

$\Theta(R)$-layer transformer with embedding dimension $\approx \Theta(N^{\delta})$

can be simulated by**

Easy for MPC ⇒ Easy for transformer

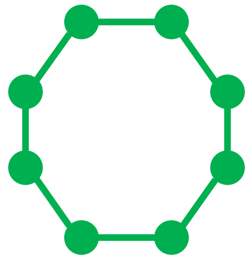
Hard for MPC ⇒ Hard for transformer

*Embedding dimension needed is actually $O(N^{\delta+\epsilon})$ for any constant $\epsilon > 0$

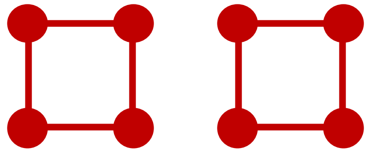
**With additional $\Theta(N^2)$ machines

What is hard for MPC?

1-vs-2 cycle problem: Given graph G that is promised to be either cycle on N vertices or union of two cycles on $N/2$ vertices each …



versus



… decide if G is connected

1-vs-2 cycle hypothesis (informal version) [e.g., Im et al, 2023]:

Every "efficient" MPC algorithm must use $R = \Omega(\log N)$ rounds

Theorem [SHT'24a]: 1-vs-2 cycle hypothesis implies necessity of $\Omega(\log k)$ layers in "small size" transformers for k-hop

Cf. Lower bounds via containment in constant depth circuit classes
[Liu et al, 2023; Merrill & Sabharwal, 2024; Li, Liu, Zhou, Ma, 2024; …]

More from the MPC ⇄ transformers connection

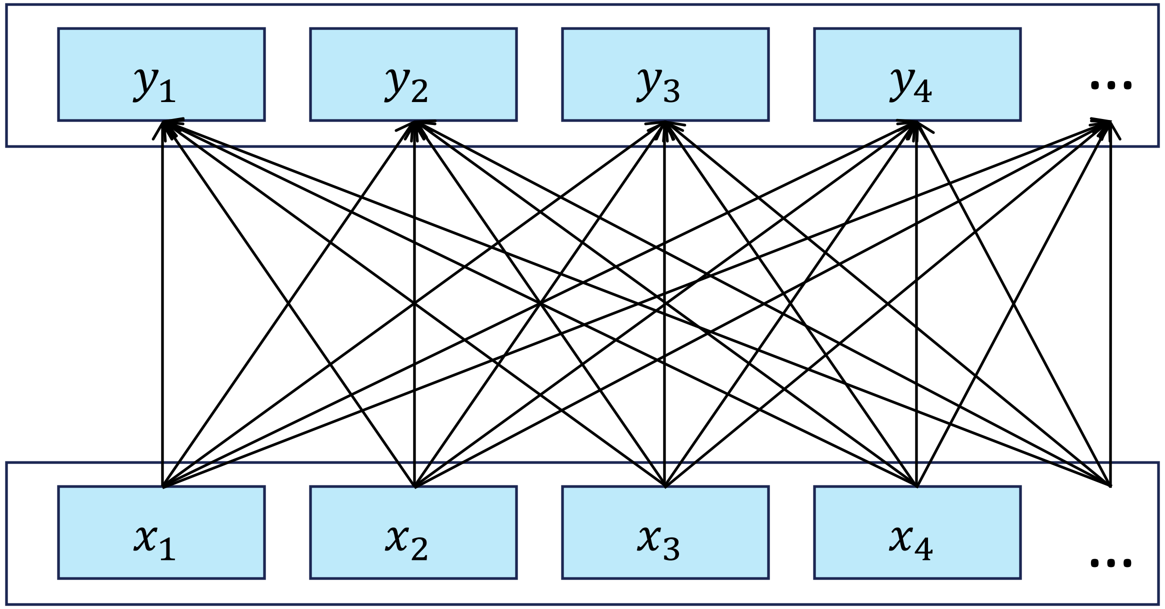
- 3-SUM: Given integers $x_1, \ldots, x_N \in [-M, M]$ (for some $M = \mathrm{poly}(N)$), determine if there exists $i, j, k \in [N]$ such that $x_i + x_j + x_k = 0$
  - Can solve in $O(N^2)$ time; conjectured to be (essentially) optimal
  - Theorem [SHT'23]: $\exists$ $O(1)$-layer transformer for 2-SUM using embedding dimension $D = O(\log N)$
  - Conjecture [SHT'23]: Every transformer for 3-SUM with $D = O(\log N)$ needs $\Omega(N^c)$ layers for some $c > 0$
- Theorem [HajiAghayi et al, 2019]: $\exists$ MPC algo. for 3-SUM using $R = O(1)$ rounds and space $S = O(N^{0.51})$ on each of $N^{0.99}$ machines
- Corollary: $\exists$ $O(1)$-layer transformer for 3-SUM using embedding dimension $D = O(N^{0.52})$

3. Limitations of sequential neural architectures

[If time permits...]

Computational cost of transformers

For self-attention, quadratic time computation appears to be inherent
[e.g., Alman & Song, 2023; Alman & Yu, 2025]



Are there sub-quadratic alternatives to self-attention?

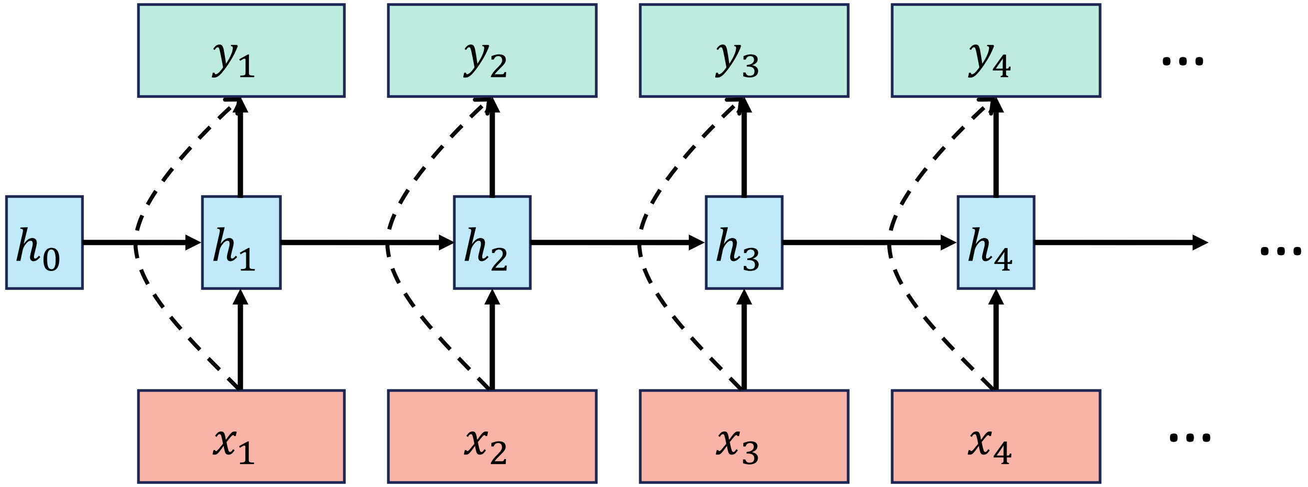
Sequential neural architectures

Recurrent neural network (RNN):

Initialize "hidden state" h_0

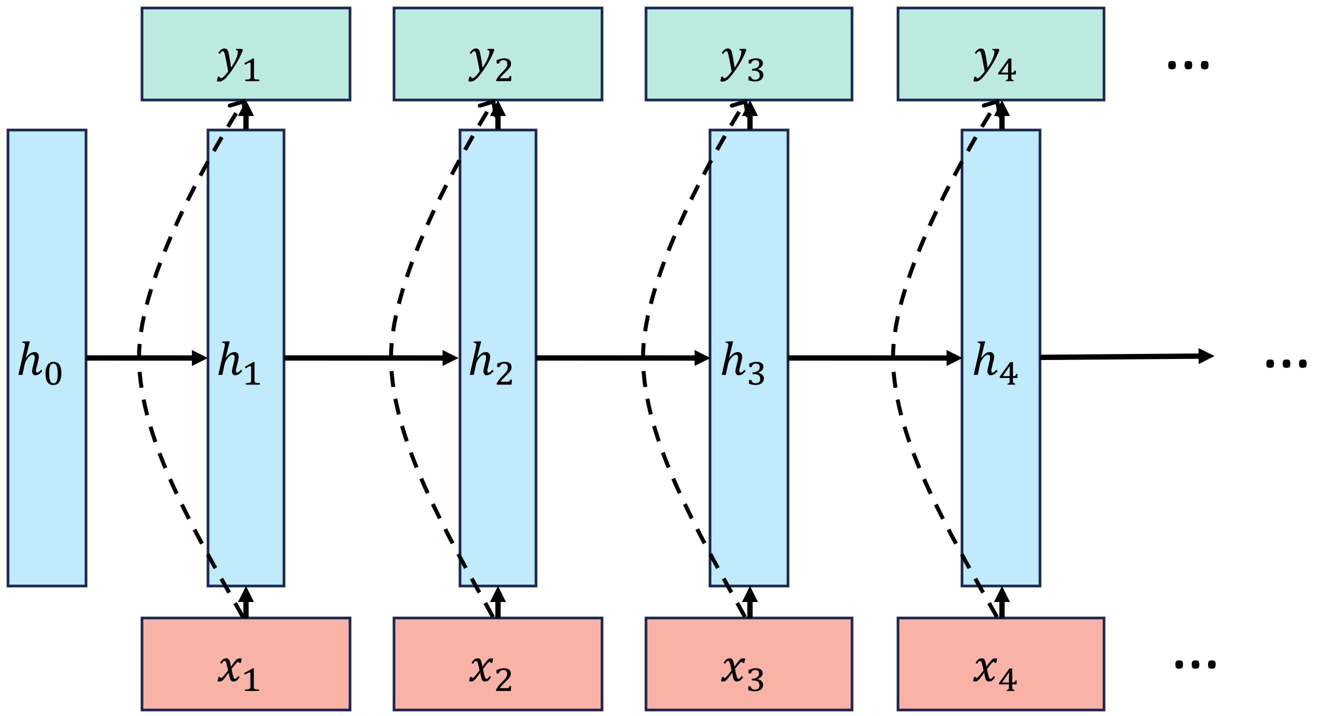
For $t = 1,2,\dots,N$:

$$h_t = \mathrm{update}_t(h_{t-1}, x_t)$$
$$y_t = \mathrm{output}_t(h_t, x_t)$$



Memory bottlenecks in RNNs

Theorem [SHT'23]:
Any RNN that computes N^{th} output of Associative Recall must use a $\Omega(N)$-bit hidden state



Further limitations for sequential architectures

Theorem [SHT'24a] (informal version):

For k-fold composition, "sequential architectures" require

$$\text{"\# sequential steps"} \geq k \quad \text{or} \quad \text{"size"} = \Omega(N/k^6)$$

(Applies to multi-layer RNNs, shallow TF with "chain-of-thought", …)

(Recall: For standard transformer, $\text{depth} = \mathrm{O}(\log k)$, $\text{size} = O(\log N)$)

Closing

1. Role of depth in transformers
    - At least two layers are necessary for associative recall ("induction head")
    - For k-fold compositions, $\log k$ layers sufficient (and probably necessary)
    - What are important function compositions in LLMs?
2. Transformers & MPC
    - Coarse reductions between transformers and MPC
    - How to characterize power of transformer "shuffle" operation?
3. Limitations of sequential neural architectures
    - How do we get around these limitations?

Thank you!