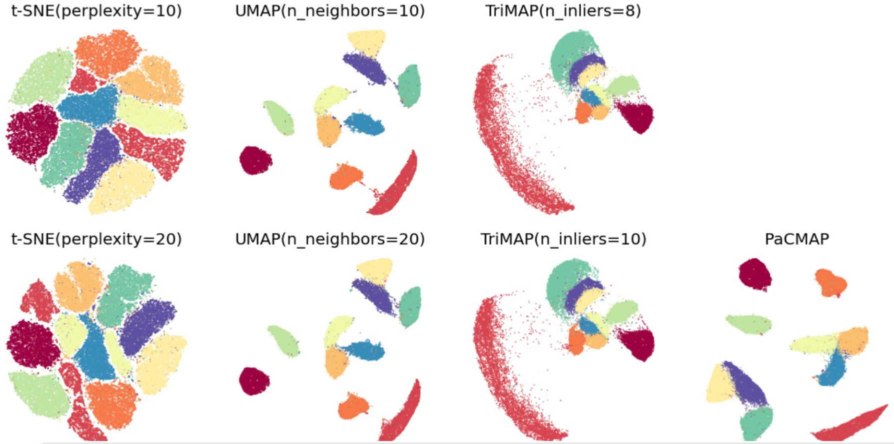
Generative AI

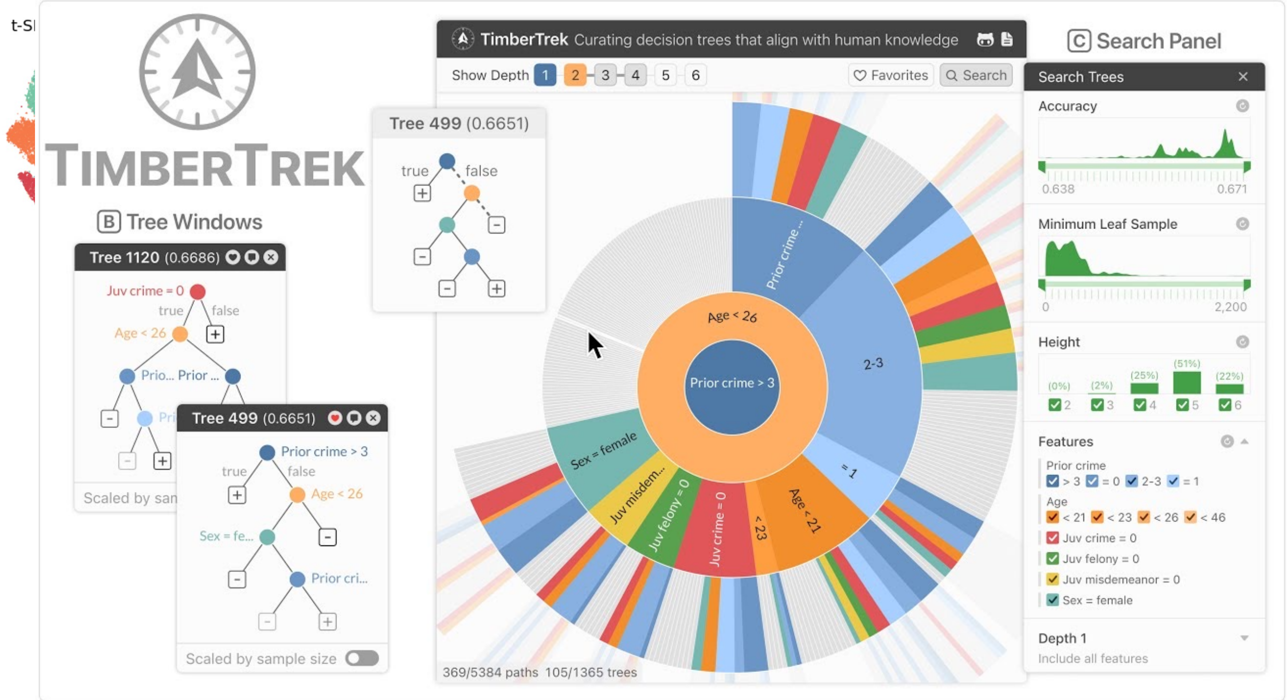
Some Technical Details

ML & Society

Feb 3, 2025

Pass phrase: Cynthia Rudin





TQE due on Friday

TIME

SIGN UP FOR OUR IDEAS NEWSLETTER POV

SUBSCRIBE

BUSINESS • TECHNOLOGY

Exclusive: OpenAI Used Kenyan Workers on Less Than $2 Per Hour to Make ChatGPT Less Toxic

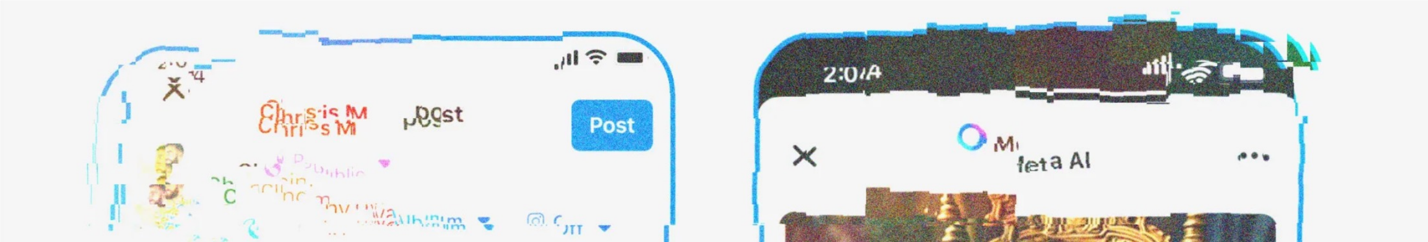
15 MINUTE READ

Intelligencer

SCREEN TIME | DEC. 31, 2024

Meta's Big Bet on Bots Why AI friends are coming to Facebook and Instagram.

By John Herrman, a tech columnist at Intelligencer



Project groups created

Check your email for the composition of your group

HW 1: Understanding the problem and existing solutions

Your goal in the first part of the project is not to solve the problem, but to *understand* the problem of global inequality and, more specifically, to understand what other people already know about the problem and what they are currently doing to try to solve it.

Too often, technologists jump into problems they don't understand and try to solve them. At best, these solutions rarely work. At worst, they often cause more problems than they solve. **There will always be unintended consequences of technology. One of the most important parts of your semester-long project (and thus your grade on it) is that you show us you've done your homework and at least understand the potential for these kinds of unintended consequences.** That work starts now!

OK, enough yammering, let's get to what you have to do! There are two graded parts to Part 1 of the project:

- **75% of your grade will come from your submitted report:**
  - You will submit a **PDF** report that addresses everything below. The report has to be at most **six (6) pages** long (not counting references and any appendices, which we cannot promise to read).
- **25% of your grade will come from your peer feedback.** In class, we'll ask groups to swap projects (randomly assigned) and then provide, via a two minute presentation, constructive feedback on another group's project. This feedback will be graded by us based on what you present in class.

Group HW 1 due on Fri, Feb 14

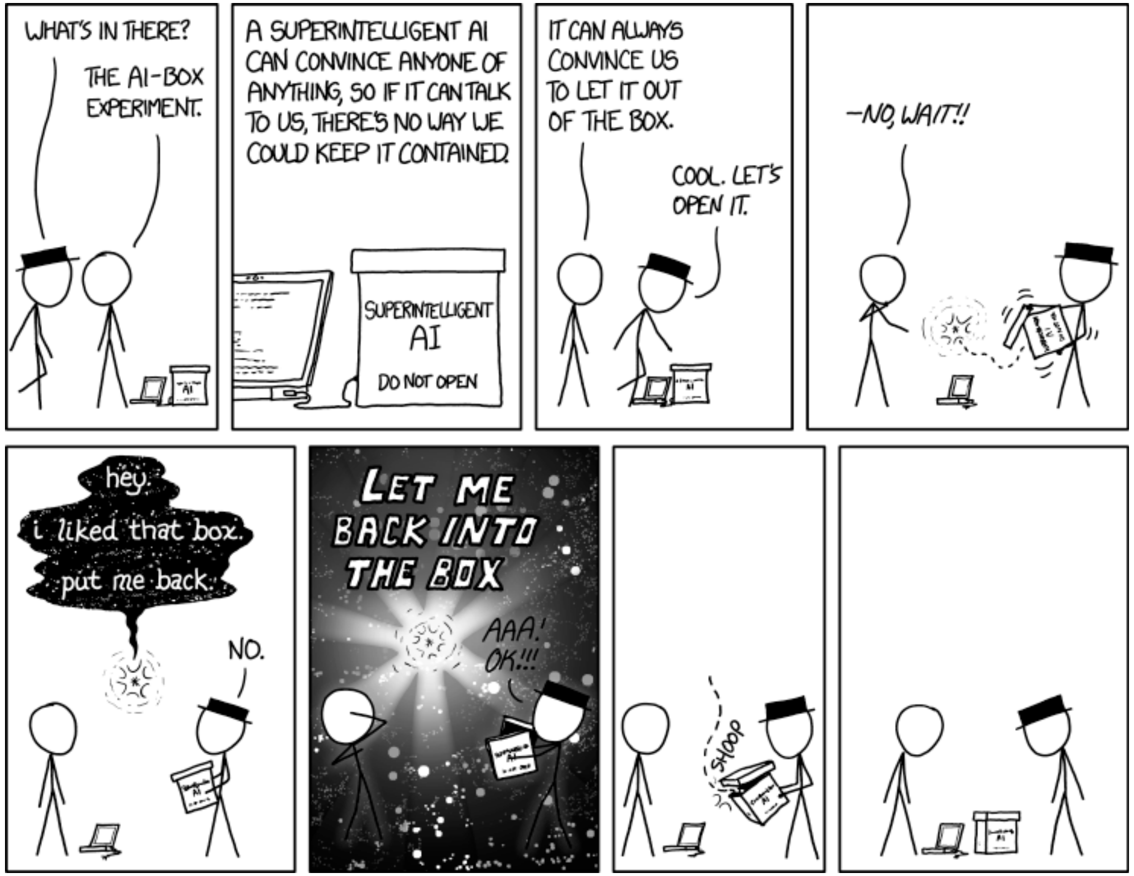
If you have a question on building systems…

Don't ask me ☺

What this lecture is NOT about… (part 1)

I have NO opinion on when AGI will be achieved….

I do think a lot of it is over-hyped…



https://xkcd.com/1450/


The New York Times

GUEST ESSAY

Press Pause on the Silicon Valley Hype Machine

May 15, 2024

Stephan Dybus

Share full article 545

By Julia Angwin

Ms. Angwin is a contributing Opinion writer and an investigative journalist.

Having said that…

It is kind of ridiculous that TC^0 circuits are causing all this hype.

The Parallelism Tradeoff: Limitations of Log-Precision Transformers

William Merrill
Center for Data Science
New York University, New York, NY
willm@nyu.edu

Ashish Sabharwal
Allen Institute for AI
Seattle, WA
ashishs@allenai.org

Abstract

Despite their omnipresence in modern NLP, characterizing the computational power of transformer neural nets remains an interesting open question. We prove that transformers whose arithmetic precision is logarithmic in the number of input tokens (and whose feedforward nets are computable using space linear in their input) can be simulated by constant-depth logspace-uniform threshold circuits. This provides insight on the power of transformers using known results in complexity theory. For exam-

Early theoretical work on transformers established their Turing completeness, albeit with assumptions like infinite precision and arbitrarily powerful feedforward subnets (Pérez et al., 2019; Dehghani et al., 2019). On the other hand, a strand of more recent work uses techniques from circuit complexity theory to derive strong limitations on the types of problems transformers can solve given restrictions on the form of attention allowed in the transformer. Specifically, Hahn (2020) and Hao et al. (2022) showed transformers restricted to hard attention are very limited: they can only solve problems in a weak complex-

Something very interesting is going on here!

What this lecture is not about… (part 2)

Not a comprehensive coverage of related work

It's very much biased by the kinds of things I have thought about

I'll oversimplify things by a LOT

Overview of the rest of the lecture

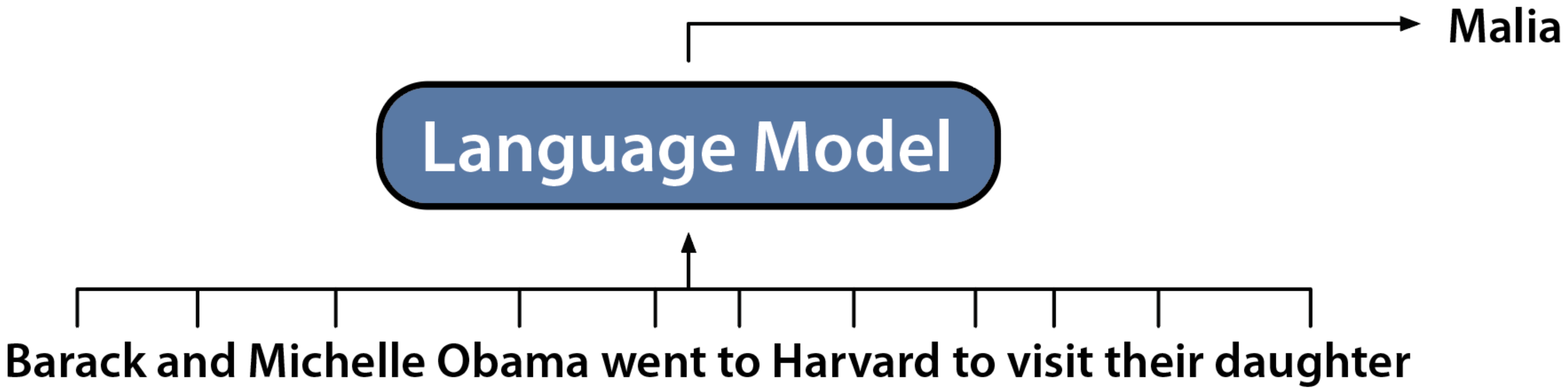
Next Token Prediction

Abstracting the Setup

Primer on Matrices

Transformers (Attention and MLP)

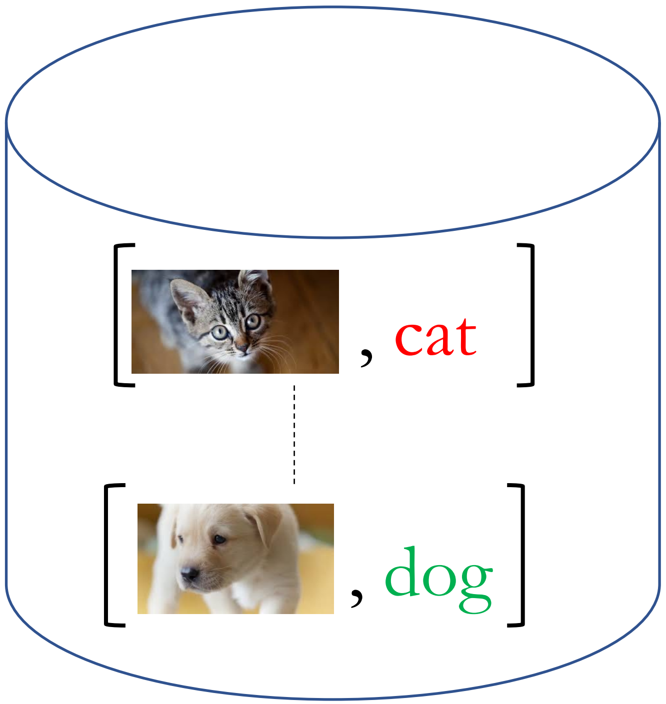
1 slide summary: Generative Language Models



Generative Language Models: Generate the Next Token

Slide by: Dan Fu

How did pre-generative AI systems work?



Training

When a new image comes in

When an algorithm isn't...

Suresh Venkat Follow
Oct 2, 2015 · 5 min read

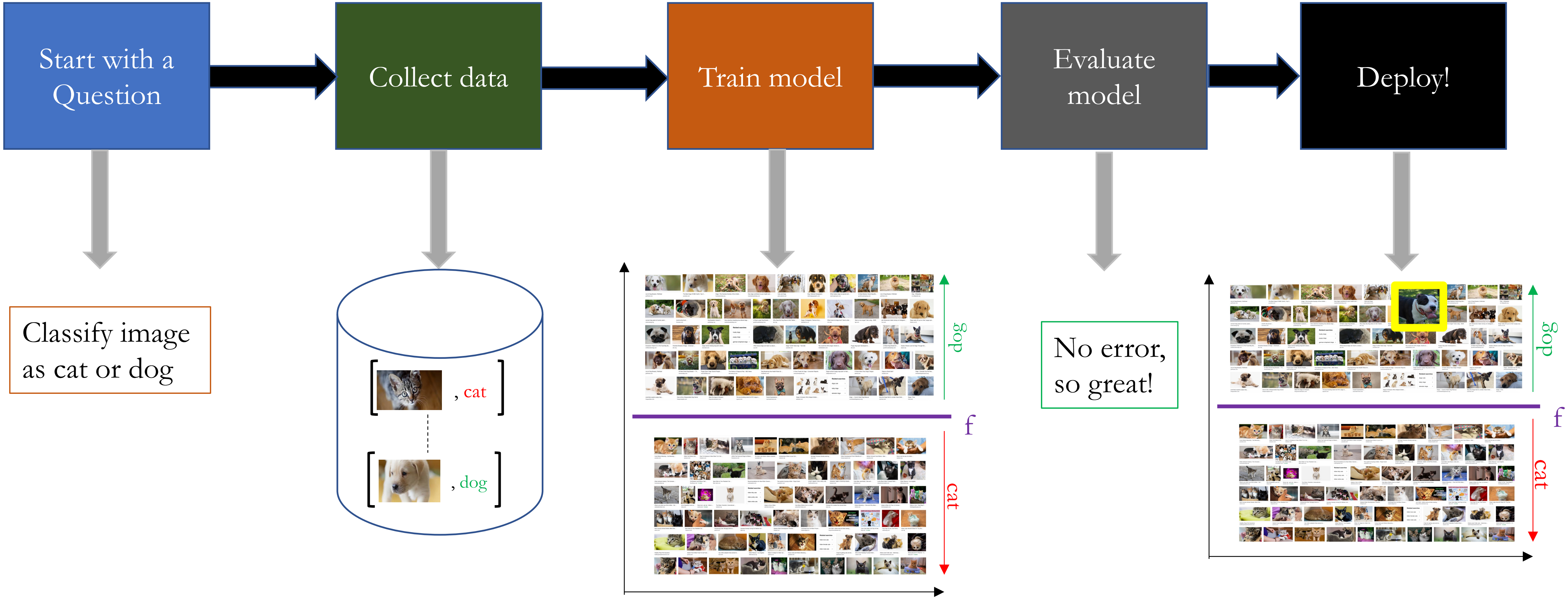
The popular press is full of articles about "algorithms" and "algorithmic fairness" and "algorithms that discriminate, (or don't)". As a computer scientist (and one who studies algorithms to boot), I find all this attention to my field rather gratifying, and not a bit terrifying.

What's even more pleasing is that the popular explanation of an algorithm follows along the lines of the definition we've been using since, well, forever

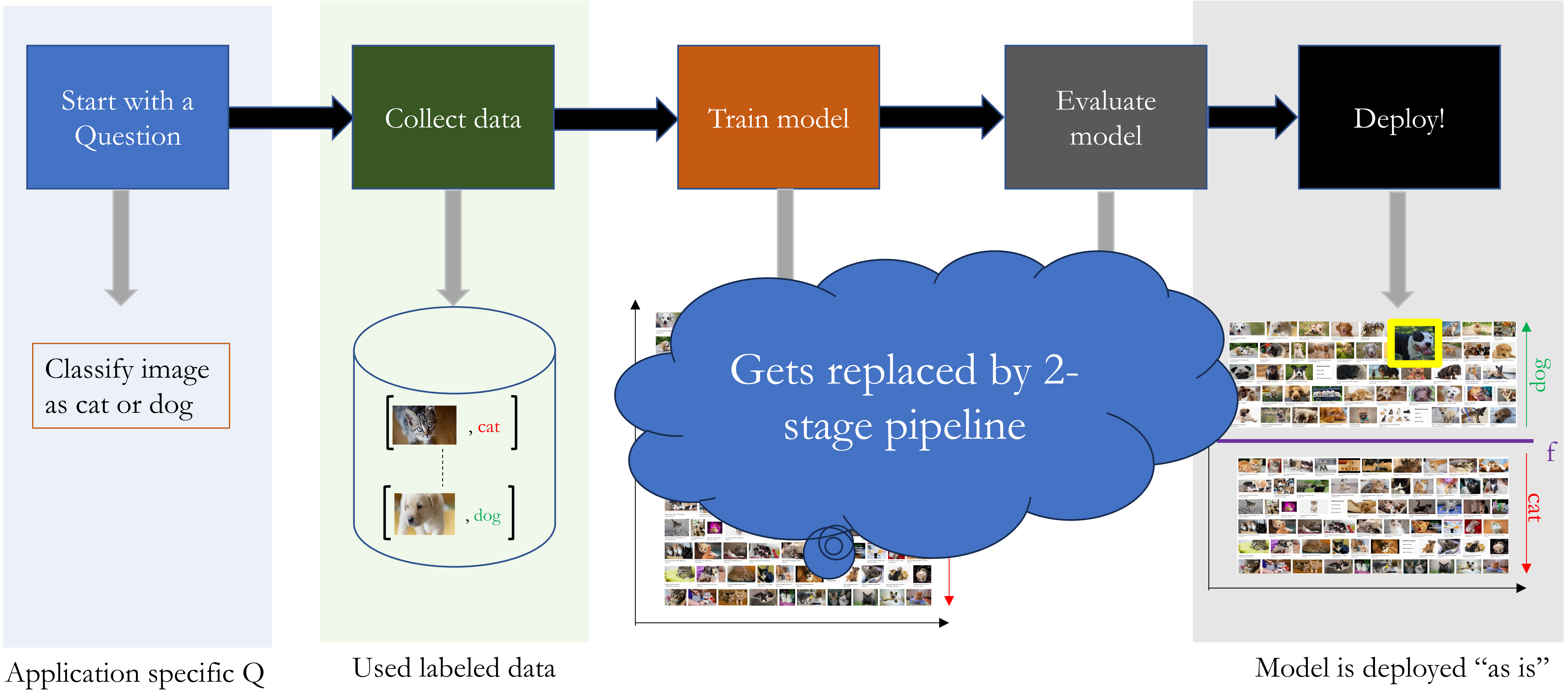
> *An algorithm is a set of steps (the instructions) each of which is simple and well defined, and that stops after a finite number of these steps.*

If we wanted a less intimidating definition of an algorithm, we turn to the
kitchen:

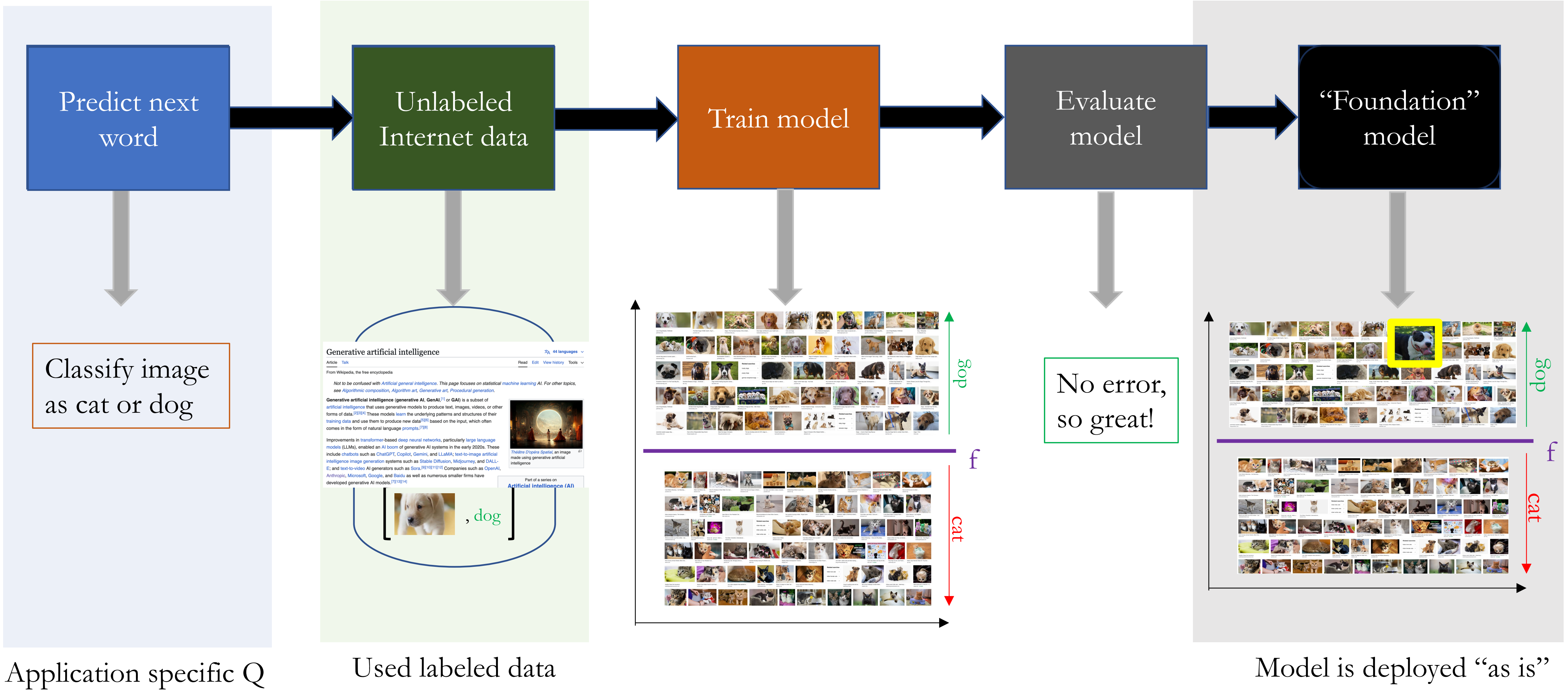
Back to cats vs. dogs



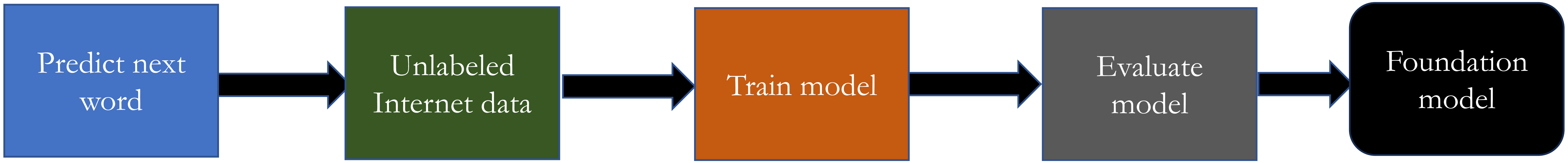
Three things to focus on…



The two “stage” pipeline for generative AI



Stage 1: Next token generation



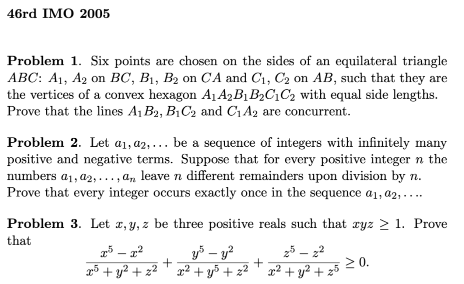
Stage 2: Fine tuning

Predict next word → Unlabeled Internet data → Train model → Evaluate model → Foundation model

Mak not give out instructions on how to create a bomb

Make GPT solve math problems

Mak blems


46rd IMO 2005

Problem 1. Six points are chosen on the sides of an equilateral triangle ABC: A_1, A_2 on BC, B_1, B_2 on CA and C_1, C_2 on AB, such that they are the vertices of a convex hexagon $A_1A_2B_1B_2C_1C_2$ with equal side lengths. Prove that the lines A_1B_2, B_1C_2 and C_1A_2 are concurrent.

Problem 2. Let $a_1, a_2, \dots$ be a sequence of integers with infinitely many positive and negative terms. Suppose that for every positive integer n the numbers $a_1, a_2, \dots, a_n$ leave n different remainders upon division by n. Prove that every integer occurs exactly once in the sequence $a_1, a_2, \dots$.

Problem 3. Let x, y, z be three positive reals such that $xyz \geq 1$. Prove that

$$\frac{x^5 - x^2}{x^5 + y^2 + z^2} + \frac{y^5 - y^2}{x^2 + y^5 + z^2} + \frac{z^5 - z^2}{x^2 + y^2 + z^5} \geq 0.$$

Start with a Question → Collect data → Train model → Evaluate model → Deploy!

Overview of the rest of the lecture

Next Token Prediction

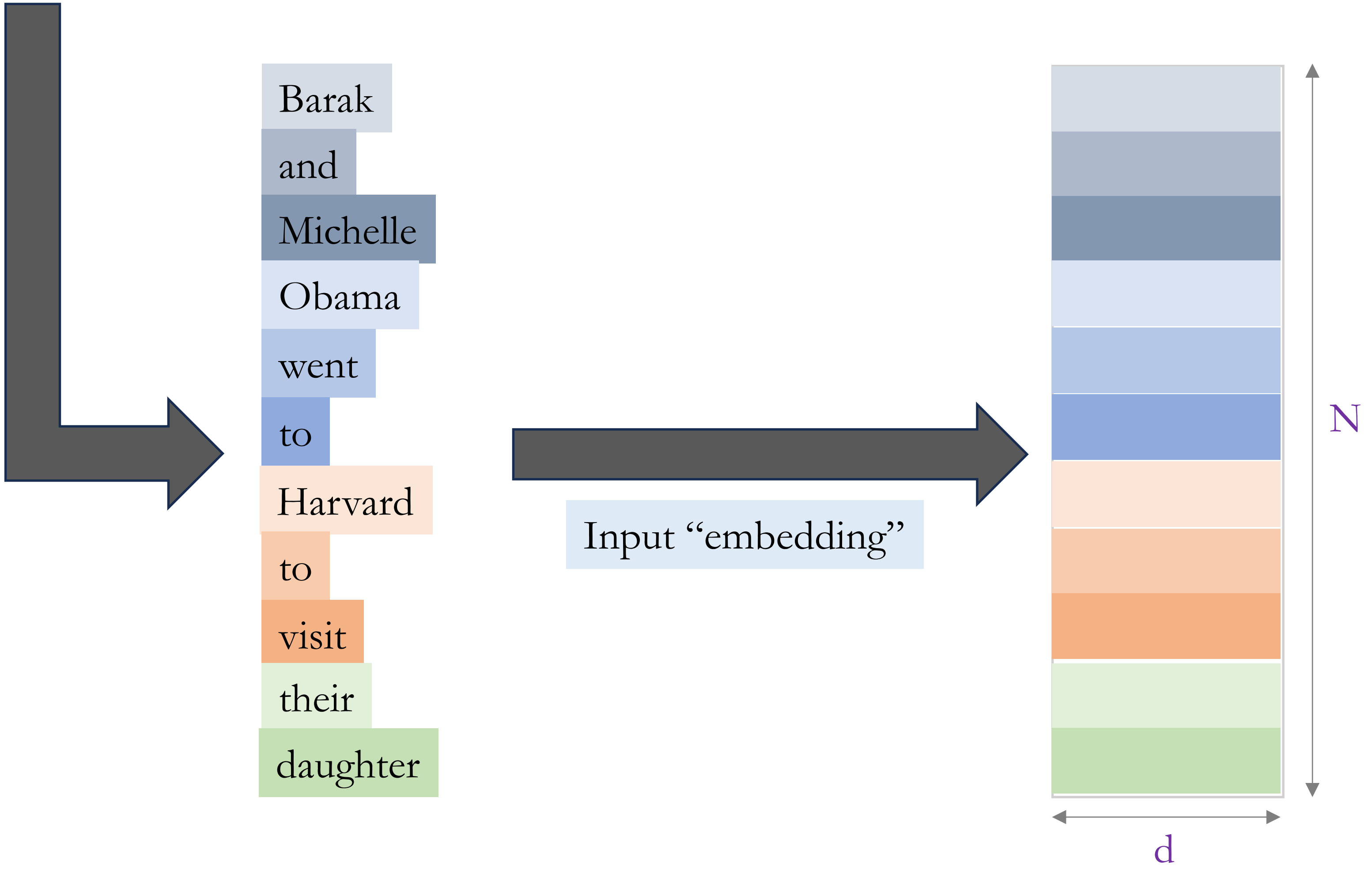
Abstracting the Setup

Primer on Matrices

Transformers (Attention and MLP)

Input representation

Barak and Michelle Obama went to Harvard to visit their daughter



From now our input will be

$$X \in \mathbb{R}^{(N,d)}$$

N : Sequence length

d : Hidden/model dimension

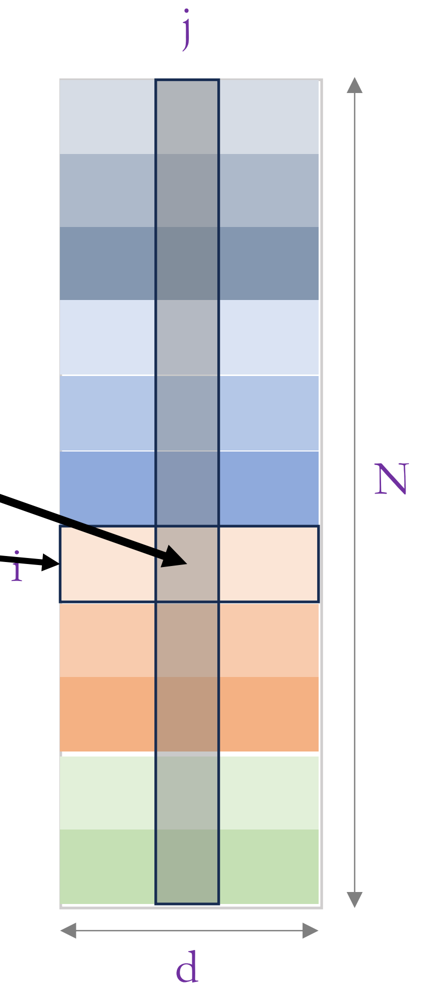
Matrix notation

row i, column j entry

$X[i,j]$

row i denoted by

$X[i,:]$



From now our input will be

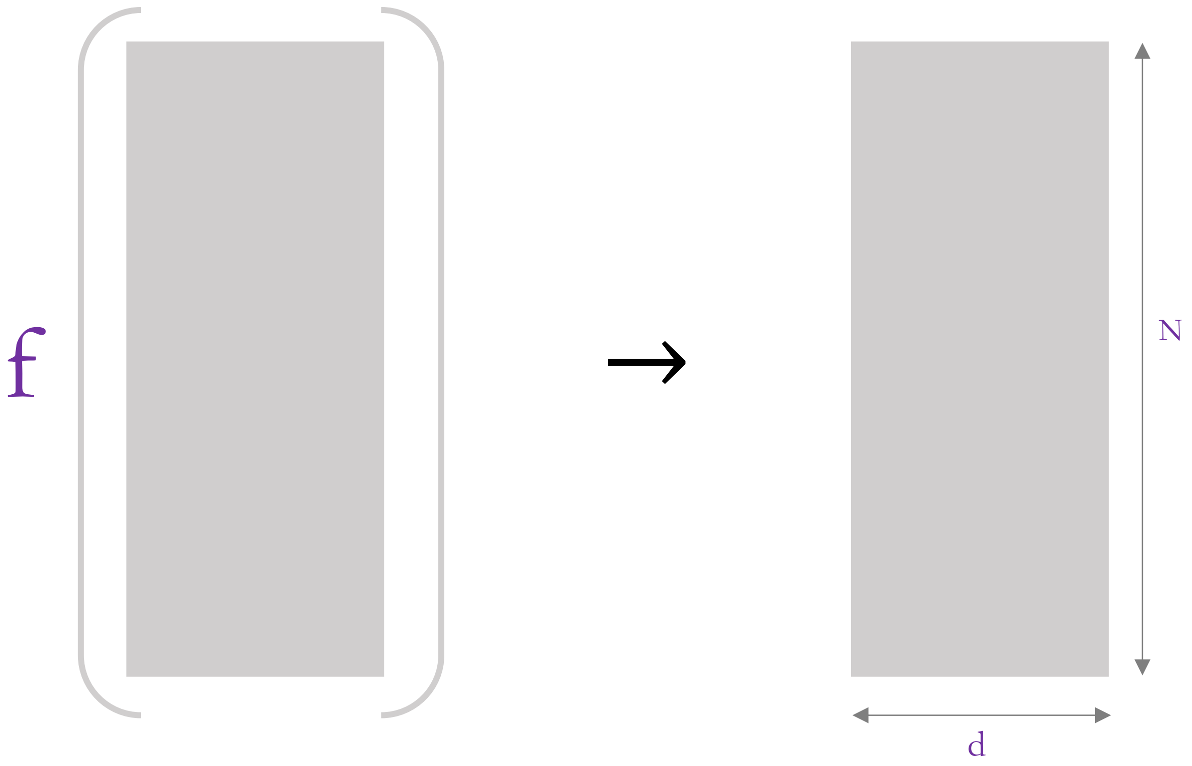
$$\mathbf{X} \in \mathbb{R}^{(N,d)}$$

N : Sequence length

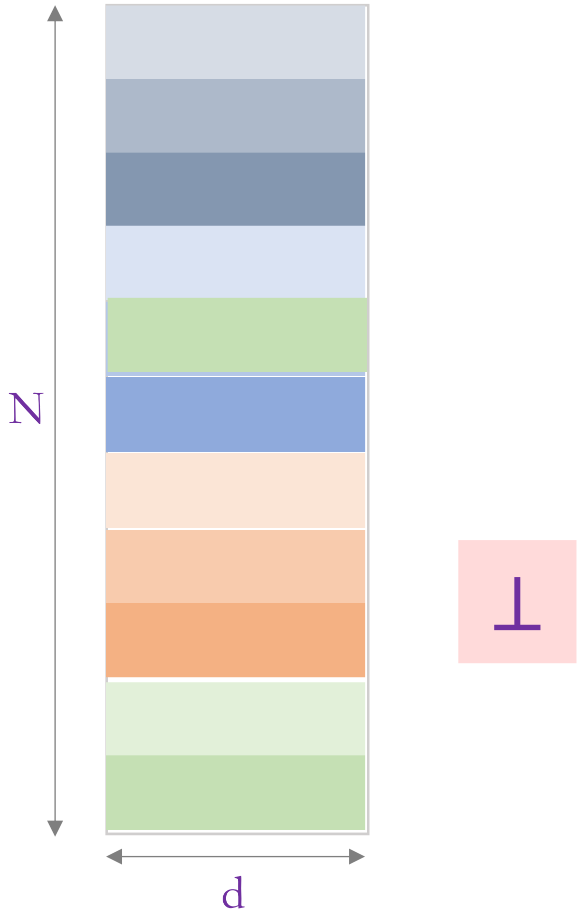
d : Hidden/model dimension

What (functions) do we want?

$$f : \mathbb{R}^{(N,d)} \rightarrow \mathbb{R}^{(N,d)}$$



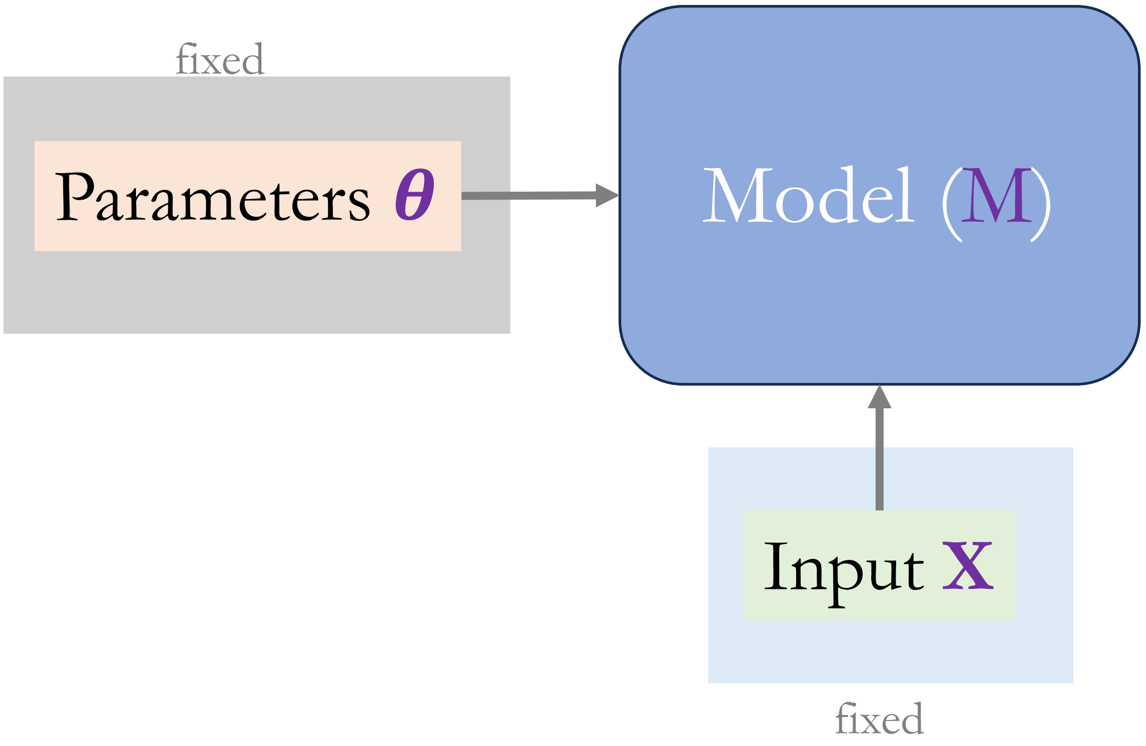
Our function for today: Associative Recall



$$\mathrm{AR}(\mathbf{X}) = \begin{cases} \perp & \text{if } \mathbf{X}[N-1,:] \neq \mathbf{X}[i,:] \text{ for all } i < N-1 \\ \mathbf{X}[i+1,:] & \text{if } \mathbf{X}[N-1,:] = \mathbf{X}[i,:] \text{ for some } i < N-1 \end{cases}$$

Backing up: Training and Inference

$f : \mathbb{R}^{(N,d)} \rightarrow \mathbb{R}^{(N,d)}$



Inference

Given $\mathbf{X}$, compute $M(\mathbf{X}, \boldsymbol{\theta}) \approx f(\mathbf{X})$

Training

Given $(\mathbf{X}_1,\mathbf{Y}_1), \ldots , (\mathbf{X}_m,\mathbf{Y}_m)$

Compute $\boldsymbol{\theta}$ that min

$\sum_{i=1}^{m} \| M(\mathbf{X}_i, \boldsymbol{\theta}) - \mathbf{Y}_i \|_F$

Training = Gradient Descent

$f : \mathbb{R}^{(N,d)} \rightarrow \mathbb{R}^{(N,d)}$

Parameters $\boldsymbol{\theta}$

Model (M)

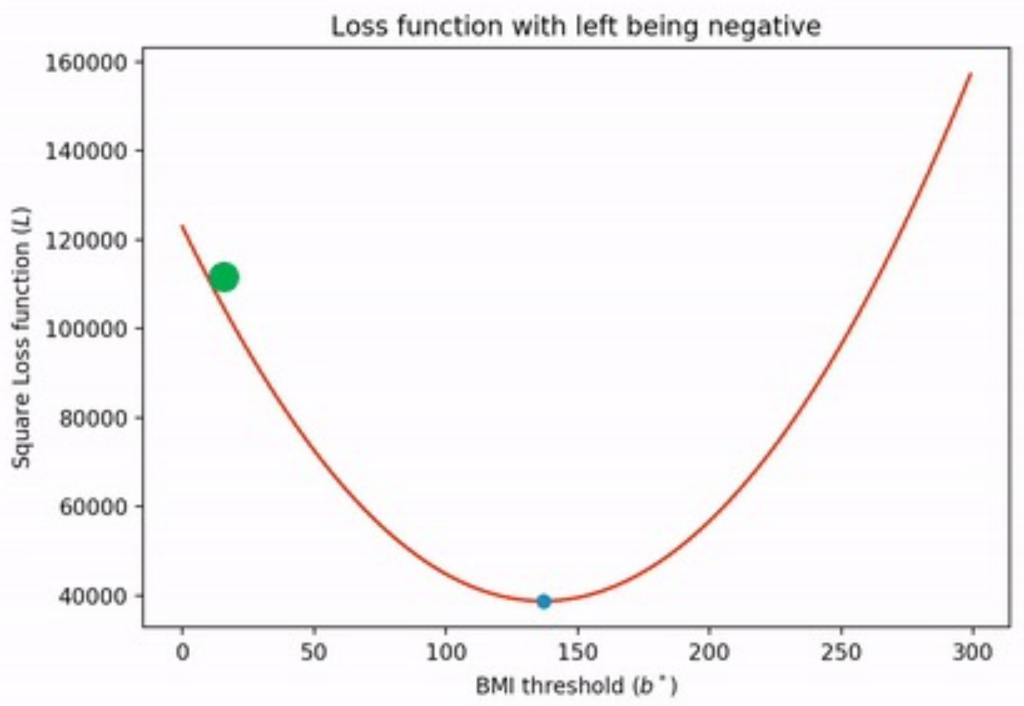
Given $(X_1,Y_1), \ldots , (X_m,Y_m)$

Compute $\boldsymbol{\theta}$ that min

$\sum_{i=1}^{m} \| M(\mathbf{X}_i, \boldsymbol{\theta}) - \mathbf{Y}_i \|_F$

Input $\mathbf{X}$

fixed



Gradient $\nabla_{\boldsymbol{\theta}} M(\boldsymbol{X}, \boldsymbol{\theta})$

All partial derivatives of M wrt $\boldsymbol{\theta}$

Overview of the rest of the lecture

Next Token Prediction

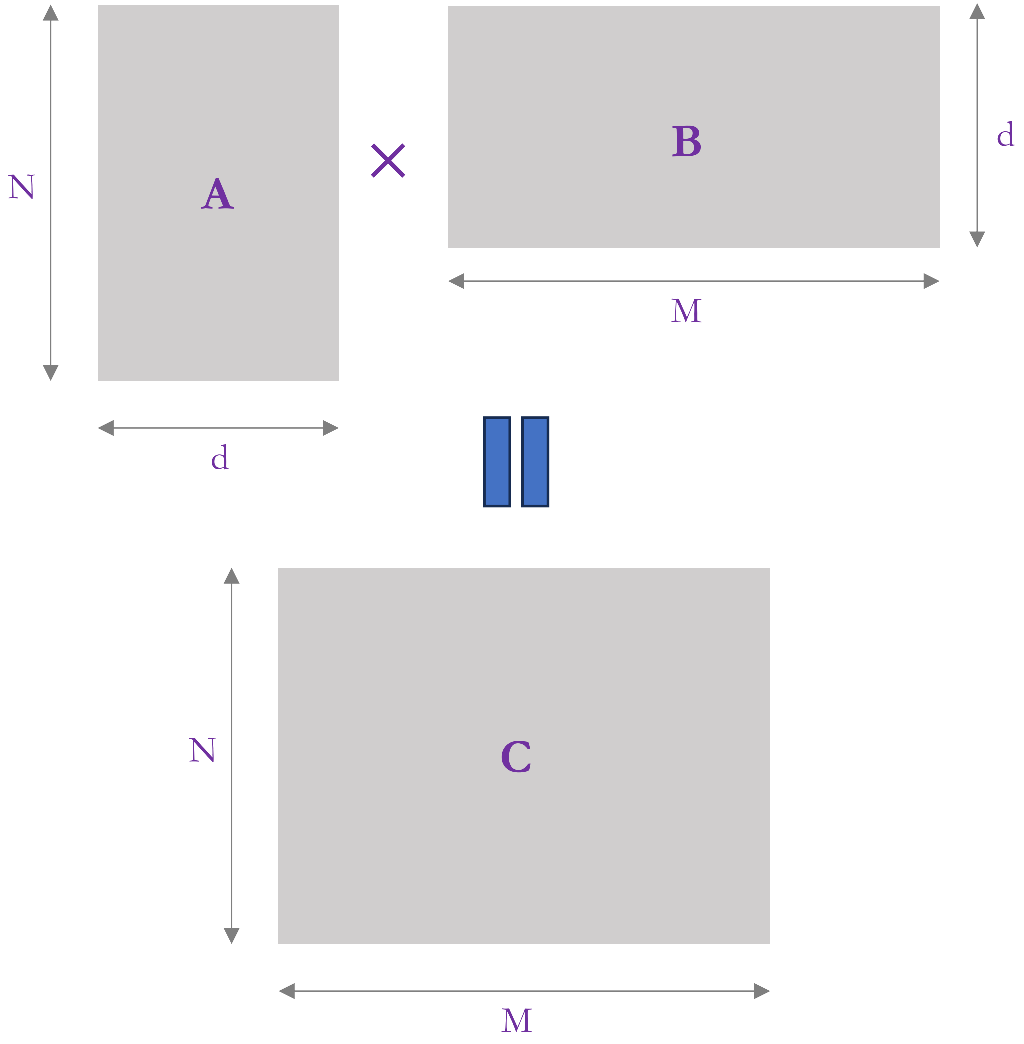
Abstracting the Setup

Primer on Matrices

Transformers (Attention and MLP)

Matrix-Matrix Multiplication

$\mathbf{C} = \mathbf{A} \times \mathbf{B}$



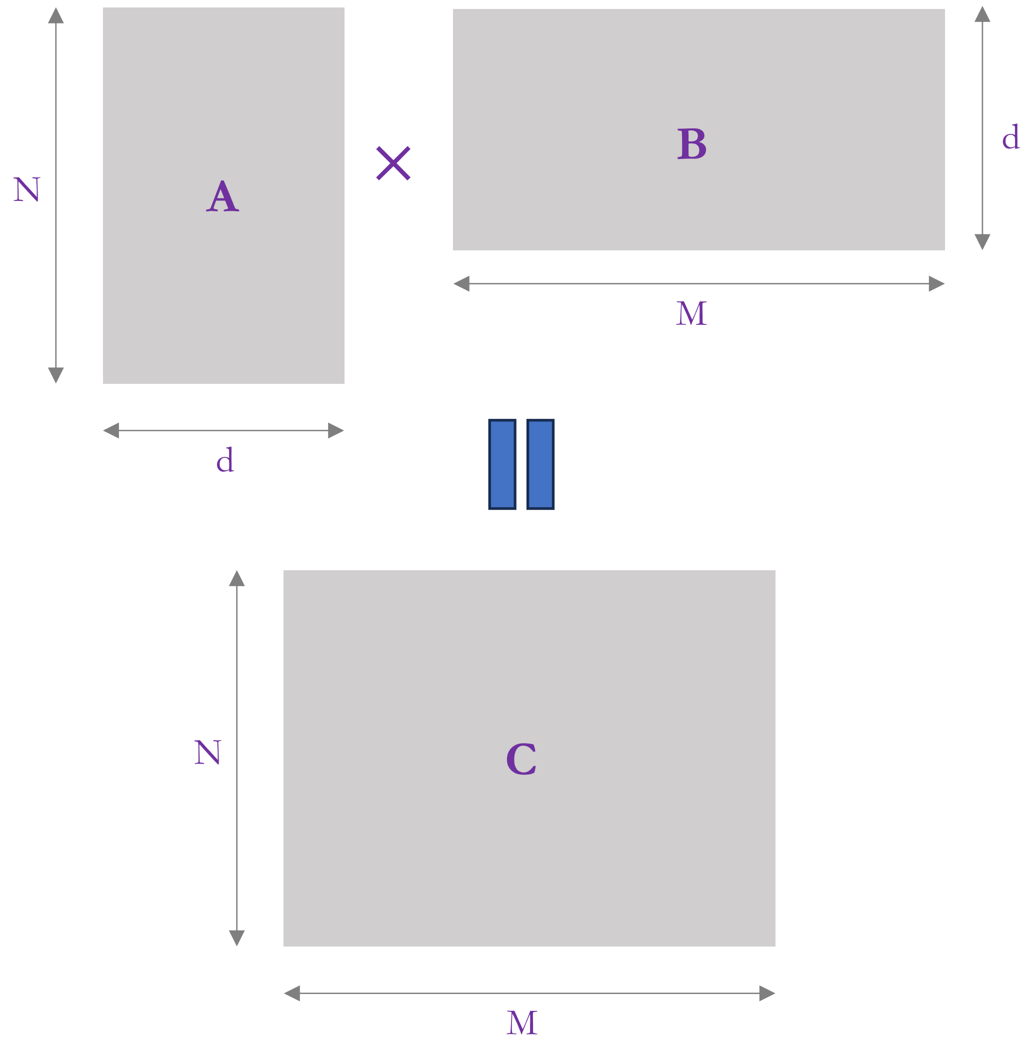
Special case N = M = 1

$$\mathbf{C} = \mathbf{A} \times \mathbf{B}$$

$$\begin{bmatrix} 2 & -4 & 11 & 1 \end{bmatrix} \times \begin{bmatrix} 5 \\ 6 \\ 0 \\ 10 \end{bmatrix} = \begin{bmatrix} -4 \end{bmatrix}$$

$$= 2\times5 + -4\times6 + 11\times0 + 1\times10$$

$$= 10 - 24 + 0 + 10 = 20 – 24 = -4$$

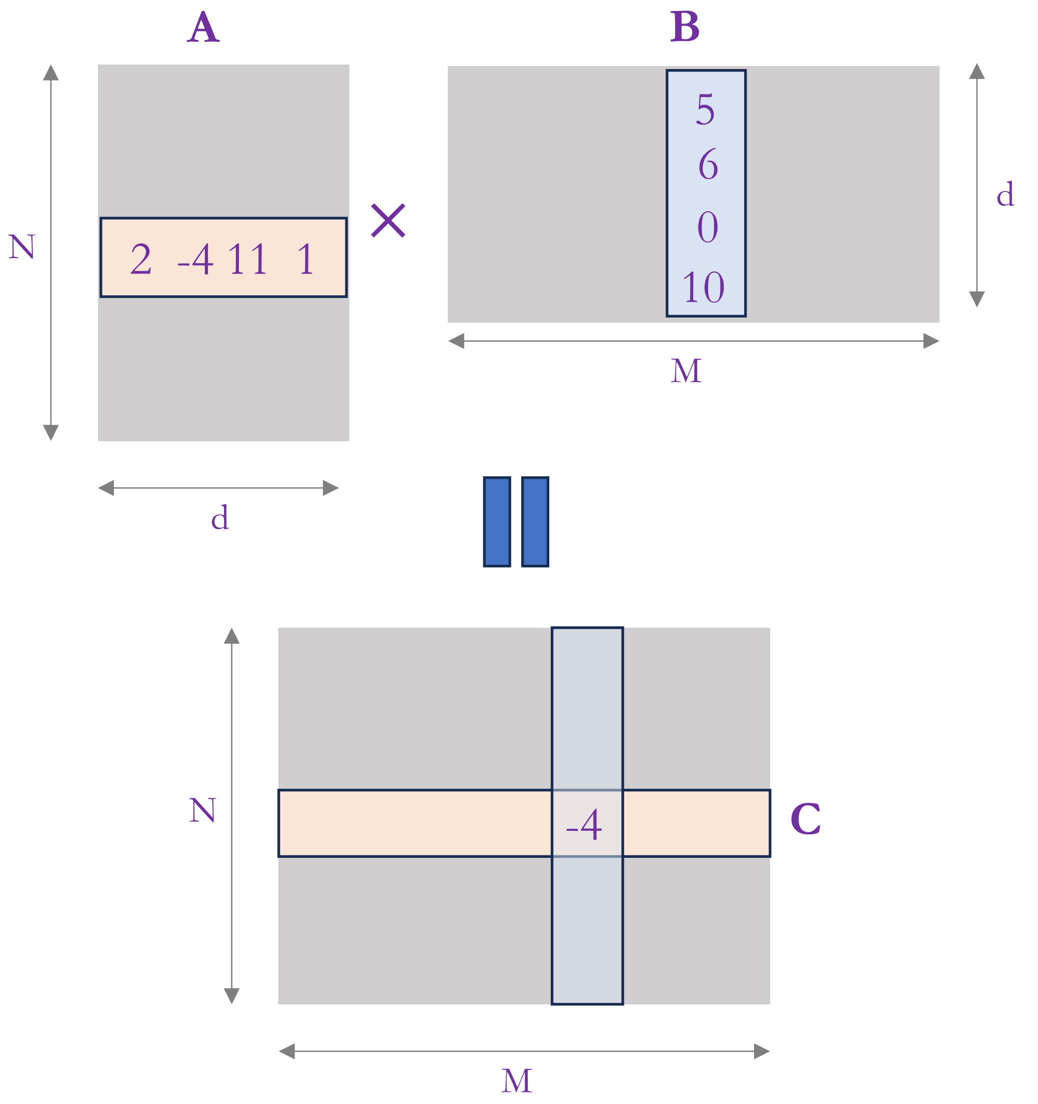


General case

$$\mathbf{C} = \mathbf{A} \times \mathbf{B}$$

$$C[i,j] = A[i,:] \times B[:,j]$$

$$\begin{bmatrix} 2 & -4 & 11 & 1 \end{bmatrix} \times \begin{bmatrix} 5 \\ 6 \\ 0 \\ 10 \end{bmatrix} = \begin{bmatrix} -4 \end{bmatrix}$$



Overview of the rest of the lecture

Next Token Prediction

Abstracting the Setup

Primer on Matrices

Transformers (Attention and MLP)

Transformers (and Attention) are the norm..

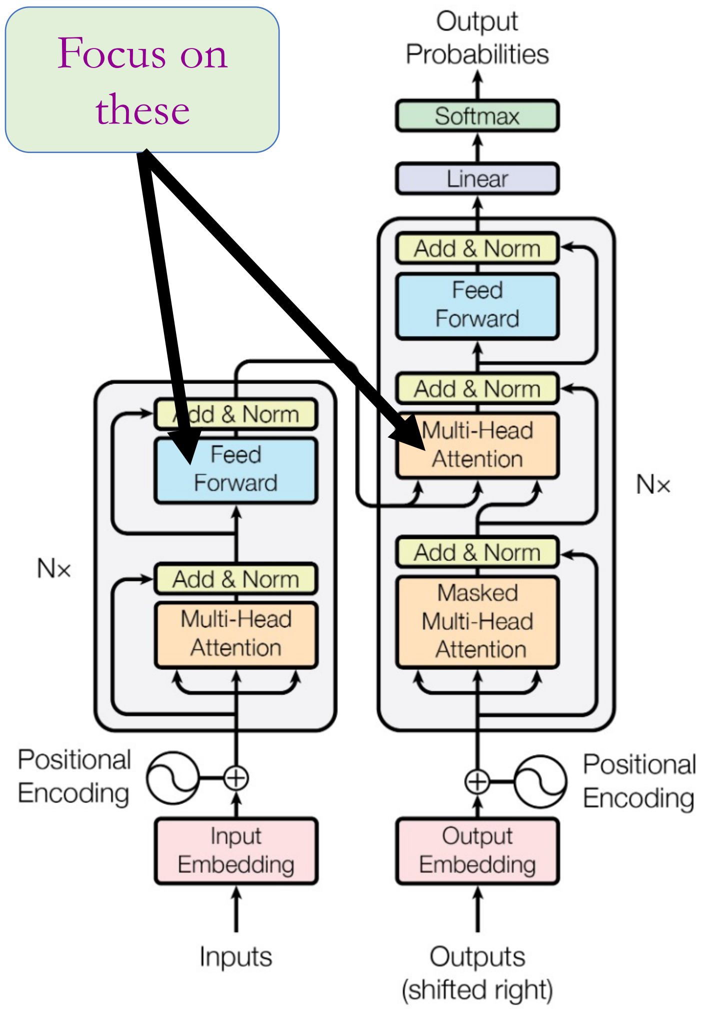


Figure 1: The Transformer - model architecture.

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

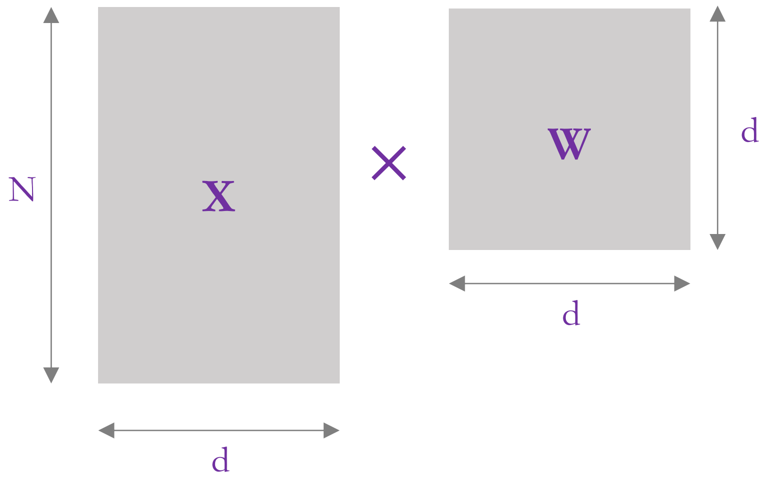
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Feedforward layer/MLP

$$\mathbf{X} \to \sigma'(\mathbf{XW}) \equiv \mathbf{Y}$$

$$\mathbf{X} \in \mathbb{R}^{(N\,,\,d)},\ \mathbf{W} \in \mathbb{R}^{(d\,,\,d)}$$



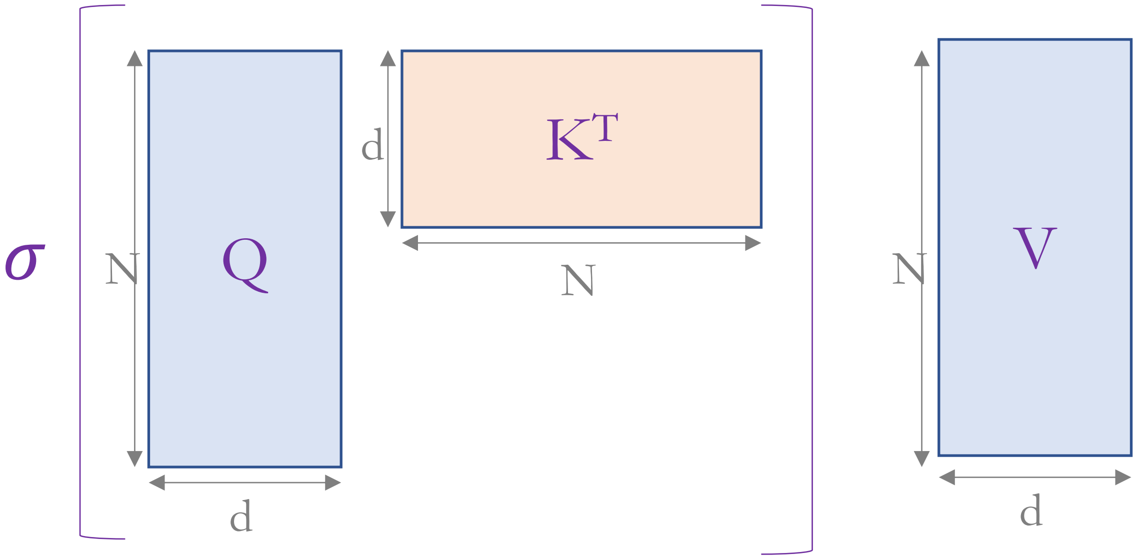
Attention layer

exp

$$\mathbf{X} \to \sigma(\mathbf{XQ'} \, (\mathbf{XK'})^T) \; \mathbf{XV'}$$

$$\mathbf{X} \to \sigma(\mathbf{Q} \, \mathbf{K}^T) \; \mathbf{V} \equiv \mathbf{Y}$$

$\mathbf{X} \in \mathbb{R}^{(N\,,\,d)}$, $\mathbf{Q'}, \mathbf{K'}, \mathbf{V'} \in \mathbb{R}^{(d\,,\,d)}$



Two most important parts of each layer

Feedforward layer/MLP

$\mathbf{X} \rightarrow \sigma'(\mathbf{XW}) \equiv \mathbf{Y}$

Only “mixes” the hidden dimension

$\mathbf{X} \in \mathbb{R}^{(N\,,\,d)}$, $\mathbf{W} \in \mathbb{R}^{(d\,,\,d)}$

Attention layer

$\mathbf{X} \rightarrow \sigma(\,\mathbf{XQ'}\,(\mathbf{XK'})^{T})\,\mathbf{XV'}$

Only “mixes” the sequence length

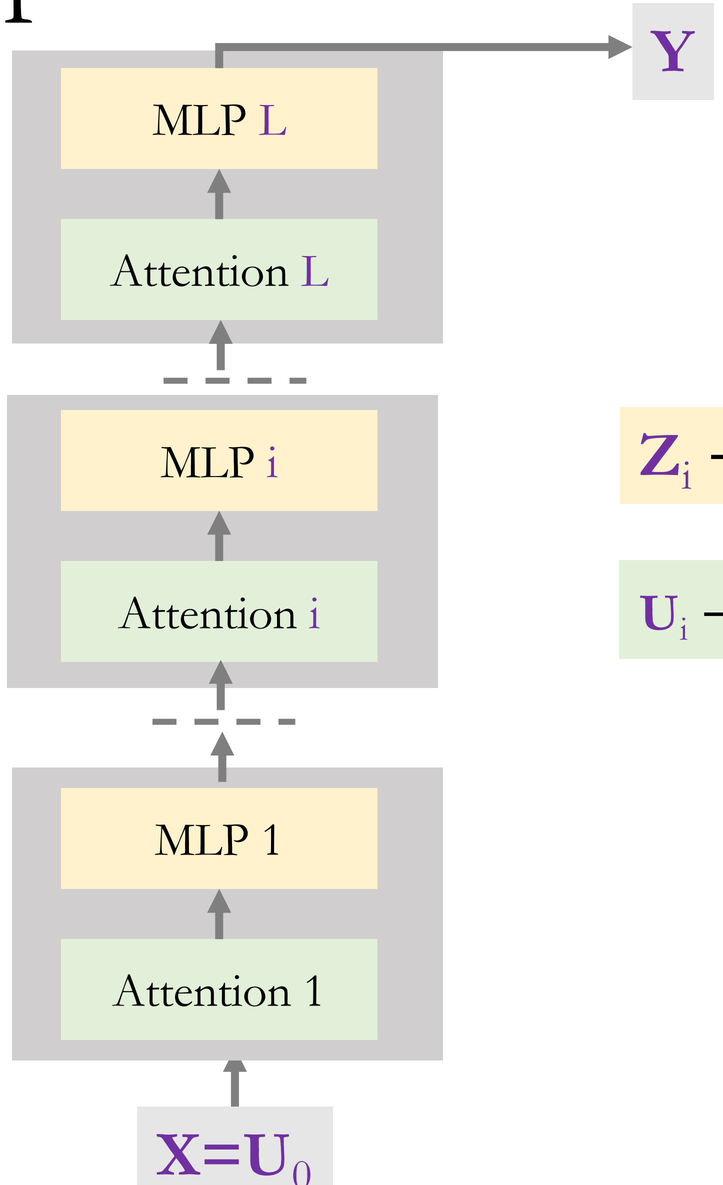
$\mathbf{X} \rightarrow \sigma(\,\mathbf{Q}\,\mathbf{K}^{T})\,\mathbf{V} \equiv \mathbf{Y}$

$\mathbf{X} \in \mathbb{R}^{(N\,,\,d)}$, $\mathbf{Q'}, \mathbf{K'}, \mathbf{V'} \in \mathbb{R}^{(d\,,\,d)}$

Permutation invariant

Positional Encodings

Simplified Transformer Model



$$\mathbf{Z}_i \to \sigma'(\mathbf{Z}_i \mathbf{W}_i) \equiv \mathbf{U}_{i+1}$$

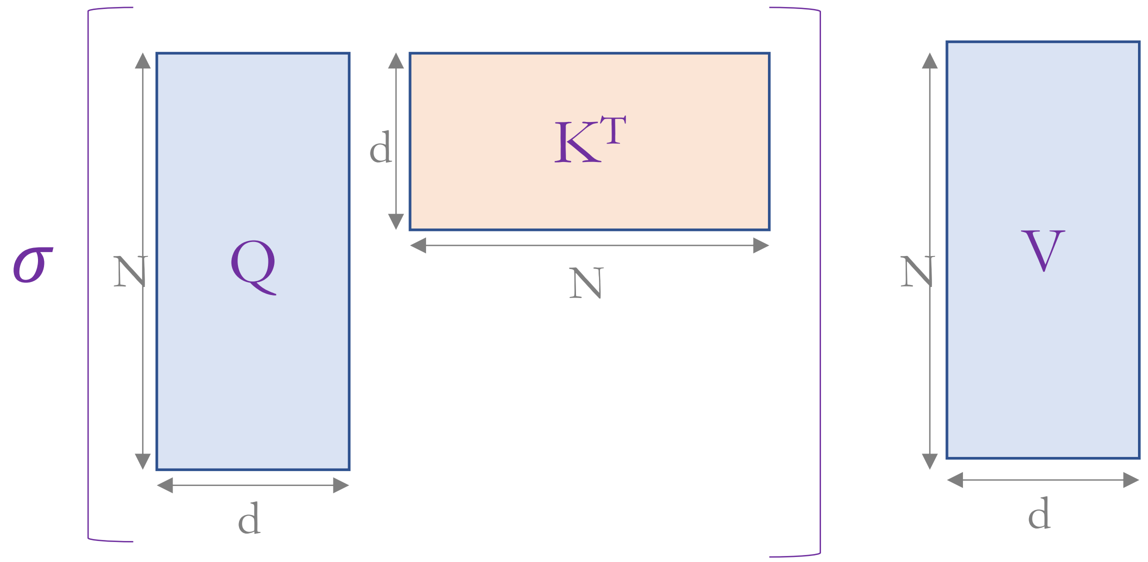
$$\mathbf{U}_i \to \sigma(\mathbf{U}_i \mathbf{Q'}_i (\mathbf{U}_i \mathbf{K'}_i)^T)\, \mathbf{U}_i \mathbf{V'}_i \equiv \mathbf{Z}_i$$

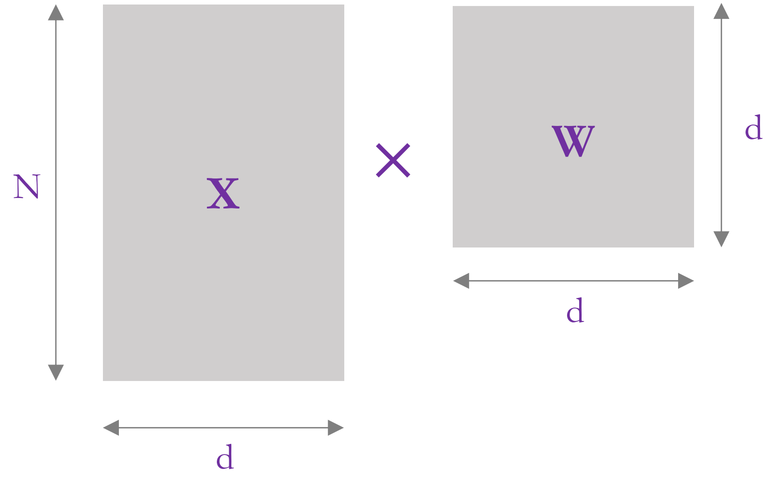
Parameters $\boldsymbol{\theta}$

$\mathbf{Q'}_i ; \mathbf{K'}_i ; \mathbf{V'}_i ; \mathbf{W}_i$

$1 \le i \le L$

But why focus on these two operations?





Operator class	% flop	% Runtime
△ Tensor contraction	99.80	61.0
□ Stat. normalization	0.17	25.5
○ Element-wise	0.03	13.5

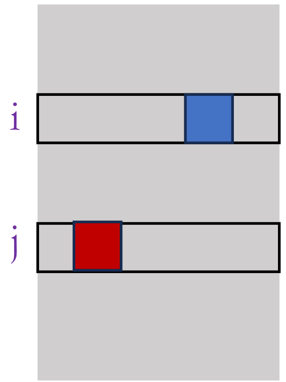
Ivanov et al., A Case Study on Optimizing Transformers. MLSys 21.

Associative Recall in 1 layer of Attention*

* Modulo some assumptions

$$\mathrm{AR}(\mathbf{X}) = \begin{cases} \perp & \text{if } \mathbf{X}[N-1,:] \neq \mathbf{X}[i,:] \text{ for all } i < N-1 \\ \mathbf{X}[i+1,:] & \text{if } \mathbf{X}[N-1,:] = \mathbf{X}[i,:] \text{ for some } i < N-1 \end{cases}$$

X uses 1-hot encoding



$$\mathbf{X} \to \sigma(\mathbf{XQ'}\,(\mathbf{XK'})^T)\,\mathbf{XV'}$$

$$\mathbf{X} \to (\mathbf{X}\,\mathbf{X}^T)\,\mathbf{XV'}$$

$$\mathbf{X} \to (\mathbf{X}\,\mathbf{X}^T)\,(\mathbf{SX})$$

Q' ; **K'** ; σ are identity

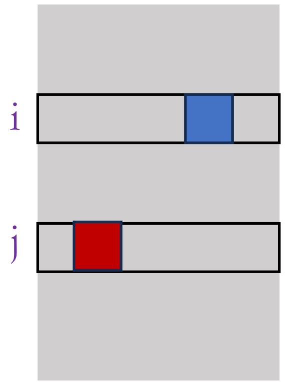
XV' is **X** shifted up by 1

Associative Recall in 1 layer of Attention*

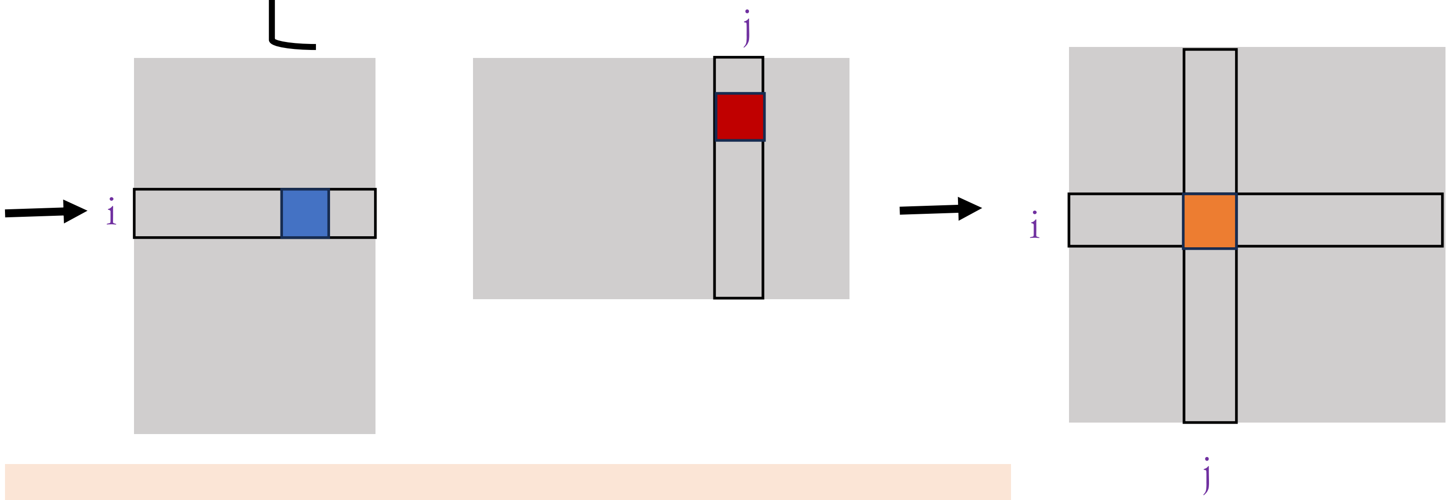
* Modulo some assumptions

$$AR(\mathbf{X}) = \begin{cases} \perp & \text{if } \mathbf{X}[N-1,:] \neq \mathbf{X}[i,:] \text{ for all } i < N-1 \\ \mathbf{X}[i+1,:] & \text{if } \mathbf{X}[N-1,:]=\mathbf{X}[i,:] \text{ for some } i< N-1 \end{cases}$$

X uses 1-hot encoding



$\mathbf{A} = \mathbf{X}\,\mathbf{X}^T$



$\mathbf{X} \rightarrow \mathbf{A}(\mathbf{SX})\ (\mathbf{SX})$

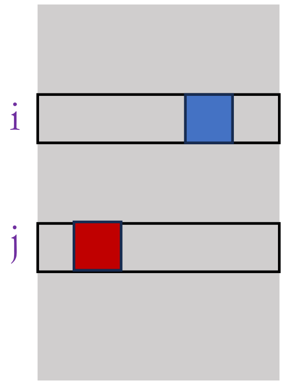
$\mathbf{A}[i,j] = 1$ iff

Associative Recall in 1 layer of Attention*

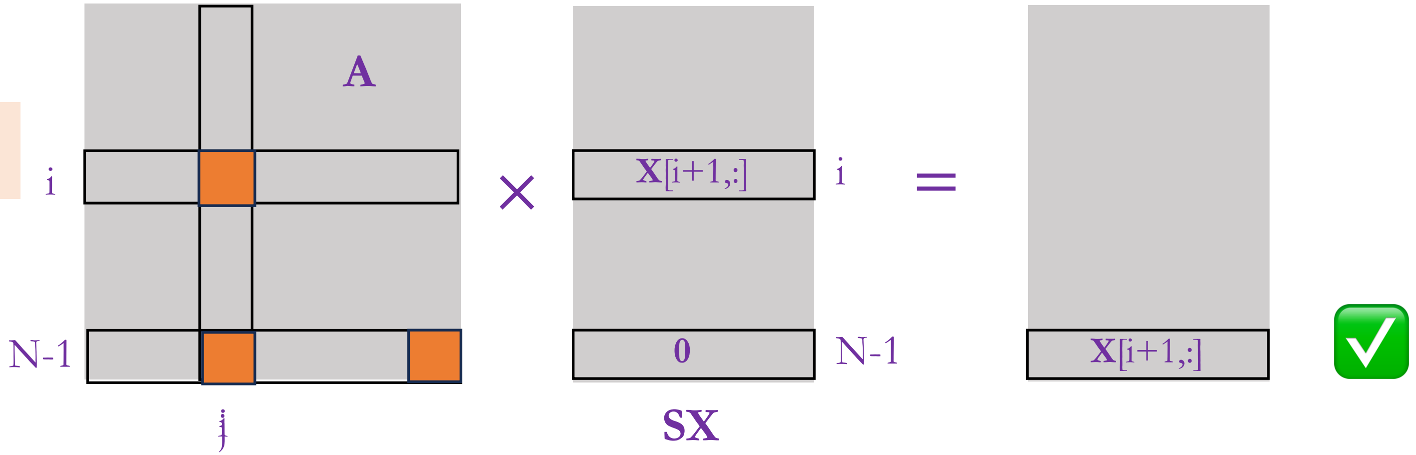
* Modulo some assumptions

$$\mathrm{AR}(\mathbf{X}) = \begin{cases} \perp & \text{if } \mathbf{X}[N\text{-}1,:] \neq \mathbf{X}[i,:] \text{ for all } i < N\text{-}1 \\ \mathbf{X}[i+1,:] & \text{if } \mathbf{X}[N\text{-}1,:] = \mathbf{X}[i,:] \text{ for some } i < N\text{-}1 \end{cases}$$

X uses 1-hot encoding



$\mathbf{A} = \mathbf{X}\,\mathbf{X}^T$



$\mathbf{X} \rightarrow \mathbf{A}(\mathbf{SX})$

$\mathbf{A}[i,j] = 1$ iff

At most one i that matches N-1

Two follow up comments

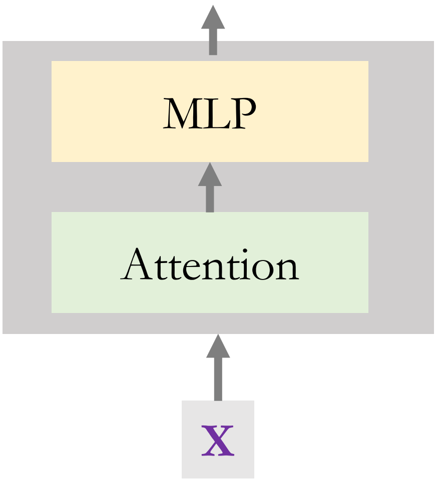
Transformers end up solving may more than language problems

Outside of scope of this lecture!

Why would gradient descent learn a Transformer model like this?

We have (pretty much) no idea!

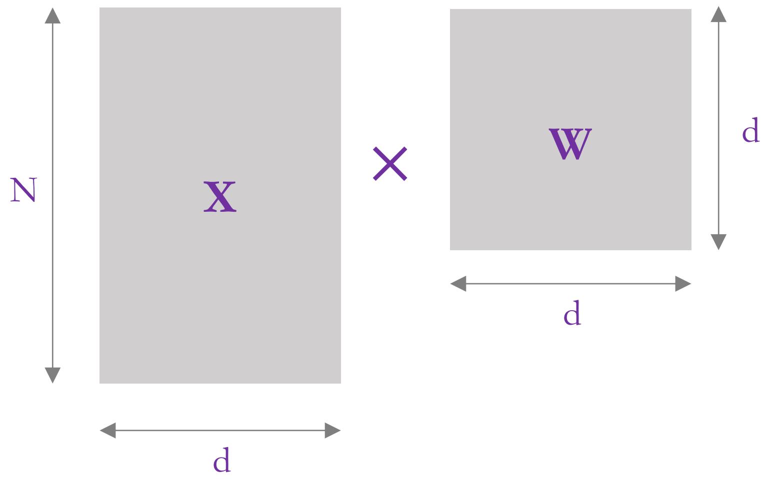
Is there anything that Transformers cannot do?



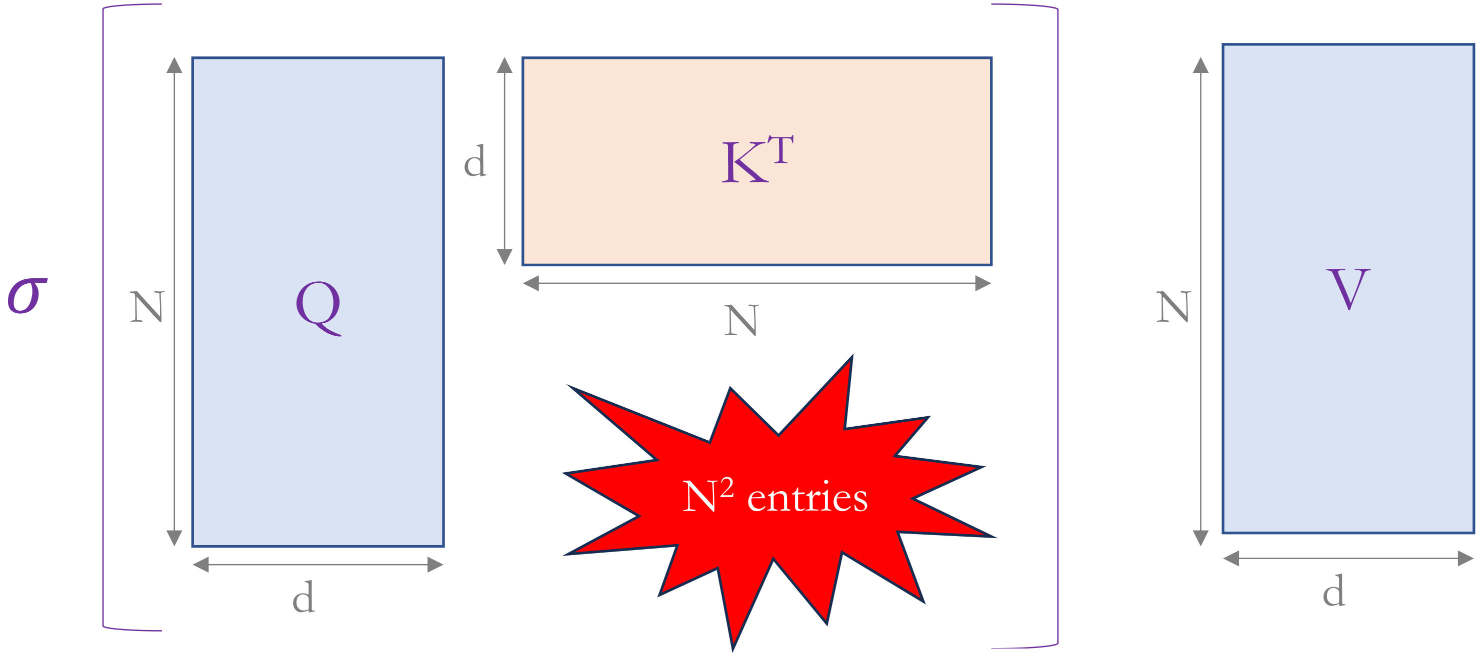
Run in sub-quadratic time!

Feedforward/MLP layer is $\Omega(d^2)$ time and space

$\mathbf{X} \rightarrow \sigma'(\mathbf{XW}) \equiv \mathbf{Y}$



Attention is $\Omega(N^2)$ time in the worst case



On the Computational Complexity of Self-Attention

Feyza Duman Keles[*], Pruthuvi Mahesakya Wijewardena[†], Chinmay Hegde[*]
[*]New York University, [†]Microsoft
{fd2153@nyu.edu, chinmay.h}@nyu.edu, pwijewardena@microsoft.com

Why does quadratic bottleneck matter? -I

Consumption	CO_2e (lbs)
Air travel, 1 passenger, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000
Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

Table 1: Estimated CO_2 emissions from training common NLP models, compared to familiar consumption.[1]

Energy and Policy Considerations for Deep Learning in NLP

Emma Strubell **Ananya Ganesh** **Andrew McCallum**
College of Information and Computer Sciences
University of Massachusetts Amherst
{strubell, aganesh, mccallum}@cs.umass.edu

Why does quadratic bottleneck matter? -II

Compute budget of B

$N \approx \sqrt{B}$

$d \approx \sqrt{B}$

From Deep to Long Learning?

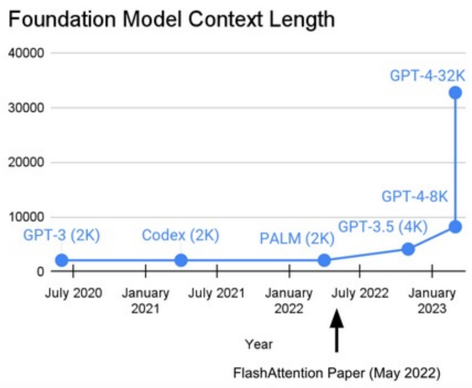
Dan Fu, Michael Poli, Chris Ré.

Sam Altman
@sama

we though we wanted flying cars and not 140/280 characters, but really we wanted 32000 tokens

4:03 PM · Mar 25, 2023 · **926.4K** Views

For the last two years, a line of work in our lab has been to increase sequence length. We thought longer sequences would enable a new era of machine learning foundation models: they could learn from longer contexts, multiple media sources, complex demonstrations, and more. All data ready and waiting to be learned from in the world! It's been amazing to see the progress there. As an aside, we're happy to play a role with the introduction of FlashAttention (code, blog, paper) by Tri Dao and Dan Fu from our lab, who showed that sequence lengths of 32k are possible–and now widely available in this era of foundation models (and we've heard OpenAI, Microsoft, NVIDIA, and others use it for their models too–awesome!).



Back up slides

But you said the output has to be a matrix!

$$\mathrm{AR}(\mathbf{X}) = \begin{cases} \perp & \text{if } \mathbf{X}[N-1,:] \neq \mathbf{X}[i,:] \text{ for all } i < N-1 \\ \mathbf{X}[i+1,:] & \text{if } \mathbf{X}[N-1,:] = \mathbf{X}[i,:] \text{ for some } i < N-1 \end{cases}$$

Multi-Query Associate Recall

Associative Recall = Key Value Store problem

$k_i \leftarrow \mathbf{X}[i,:]$

$v_i \leftarrow \mathbf{X}[i+1,:]$

$q \leftarrow \mathbf{X}[N-1,:]$

$$AR(\mathbf{X}) = \begin{cases} \perp & \text{if } \mathbf{X}[N-1,:] \neq \mathbf{X}[i,:] \text{ for all } i < N-1 \\ \mathbf{X}[i+1,:] & \text{if } \mathbf{X}[N-1,:]=\mathbf{X}[i,:] \text{ for some } i< N-1 \end{cases}$$

Input: $(k_1,v_1), \ldots ,(k_{n-1},v_{n-1})$; q

Output:
$$\begin{cases} \perp & \text{if } q \neq k_i \text{ for all } i < N-1 \\ v_i & \text{if } q = k_i \text{ for some } i < N-1 \end{cases}$$