

Large Language Models

how they work
&
what we do not understand about them (*yet*)

Prof. Constantine Dovrolis
CaSToRC Director



Have you used ChatGPT yet?

co

Write a welcome note (about 200 words) for the attendees of the following training event:
<https://castorc.cyi.ac.cy/events/data-analytics-in-the-era-of-large-scale-machine-learning>



Dear Esteemed Participants,

A warm welcome to each of you as we embark on our journey at the "Data Analytics in the Era of Large-Scale Machine Learning" training event, hosted by the Cyprus Institute. This gathering of bright minds symbolizes our shared commitment to innovation and lifelong learning.

In the coming days, expect a blend of engaging presentations, hands-on sessions, and stimulating discussions. Our industry-leading speakers will guide you through the realm of machine learning and data analytics, showcasing the latest trends and methodologies.

This event provides a unique platform to deepen your understanding, challenge assumptions, and enhance your skills. As we dive into this rich world of data and machine learning, our goal is to empower you to harness these tools effectively in your respective fields.

Let us embrace this opportunity for growth and learning. May your participation be both productive and enjoyable.

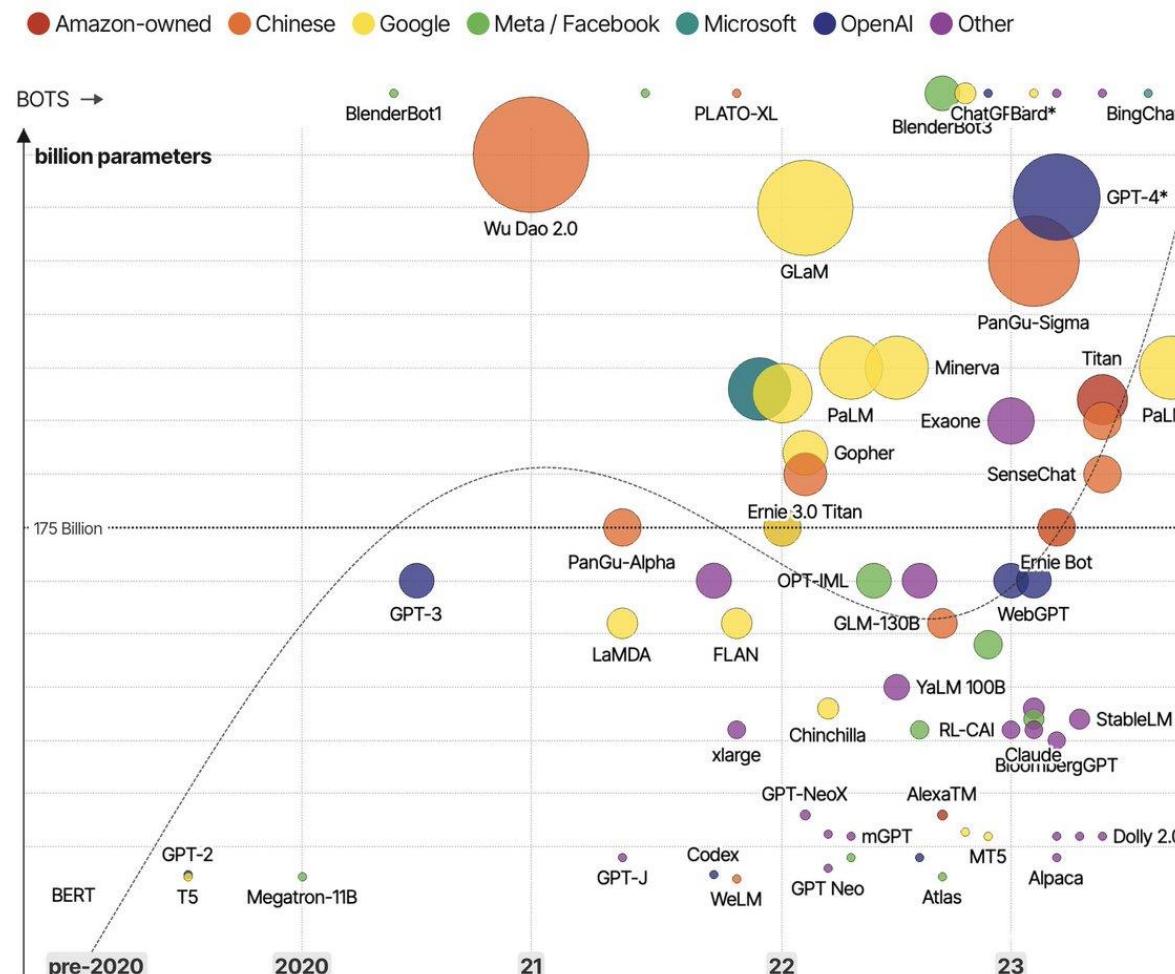
We are thrilled to have you on board and look forward to the enlightening days ahead.

Warm Regards,



Frantic Progress, Excitement, Fear, and Misunderstandings

The Rise and Rise of A.I. Large Language Models (LLMs) & their associated bots like ChatGPT



David McCandless, Tom Evans, Paul Barton
Information is Beautiful // May 2023

MADE WITH VIZsweet

INCREASED USE OF AI

IN DAILY LIFE

MORE CONCERNED THAN EXCITED

38%

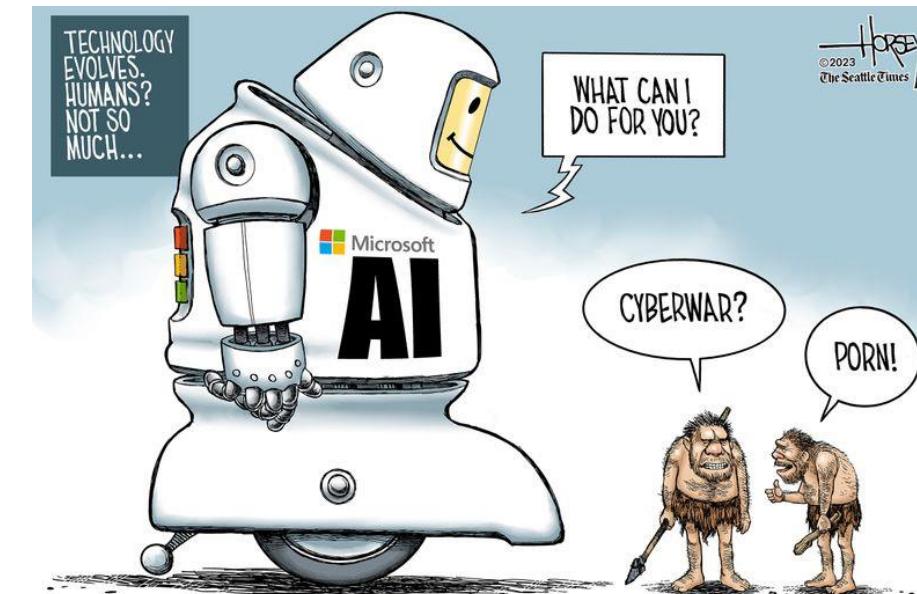
MORE EXCITED THAN CONCERNED

15%

EQUALLY CONCERNED AND EXCITED

46%

NBC News poll: Feb 19, 2023





This talk is about what is “under the hood”

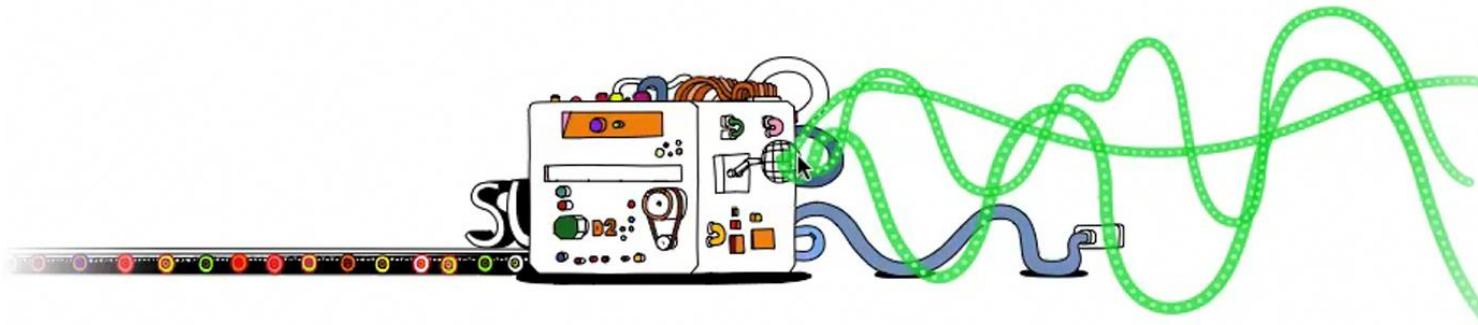


Source: <https://www.etftrends.com/etfs-the-importance-of-looking-under-the-hood/>



The Sequence Transduction problem

- Sequence to sequence mapping



Source: Alex Graves ([ICML'12 tutorial](#))

Let $\mathbf{x} = \{x_1, x_2, \dots, x_T\}$ be the input sequence and $\mathbf{y} = \{y_1, y_2, \dots, y_U\}$ be the output sequence, where $x_t \in \mathcal{S}_x$, $y_u \in \mathcal{S}_y$, and \mathcal{S}_x and \mathcal{S}_y are the sets of possible things that each x_t and y_u can be, respectively.

$$\begin{aligned} p_{\theta}(\mathbf{y}|\mathbf{x}) &= p_{\theta}(y_1, y_2, y_3, \dots, y_U | \mathbf{x}) \\ &= \prod_{u=1}^U p_{\theta}(y_u | y_{u-1}, y_{u-2}, \dots, y_1, \mathbf{x}) \end{aligned}$$



Learning tasks in LLMs

A Pretraining



Large corpus
(unlabeled text)

"Would you tell me, please, which way I ought to go from here?"
"That depends a good deal on where you want to get to," said the Cat.
"I don't much care where—" said Alice.
"Then it doesn't matter which way you go," said the Cat.
"—so long as I get *somewhere*," Alice added as an explanation.
"Oh, you're sure to do that," said the Cat, "if you only walk long enough."

Original text

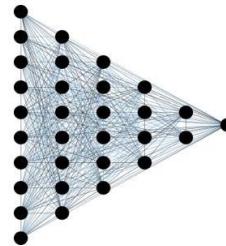
Masking



Source: A. Radford et al., "Improving Language Understanding by Generative Pre-training"

Masked text

Language model



"Would you tell me, **sir**, which way I **need** to go from here?"
"That **depends** a **good** deal on where you want to get to," said the Cat.
"I **don't** much care where—" **said** Alice.
"Then it doesn't matter **which way** you go," said the Cat.
"—so long as I get *somewhere*," Alice **added** as an explanation.
"Oh, **no need** to do that," said the Cat, "if **one** only **waits** long enough."

Predicted text

Loss

B Fine-tuning

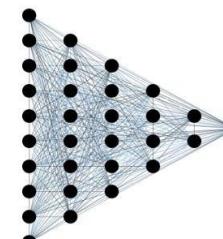


Small labeled
dataset

We wish to suggest a structure for the salt of deoxyribose nucleic acid (D.N.A.). This structure has novel features which are of considerable biological interest.

Text

Fine-tuned model



Topic: Biology (97%)

Prediction

Source: D.Ofer et al., "The language of proteins: NLP, machine learning & protein sequences"

CaSToRC - Constantine Dovrolis



- **Rule-based methods (1950s-80s):** struggled with ambiguity & complexity of human language - also not much data to work with
- **Statistical methods (till 1990s):** n-gram based methods, Hidden Markov Models (HMMs), use of Viterbi algorithm, progress in Named Entity Recognition tasks
- **Corpus linguistics (late 1980s-early 2000s):** availability of large language datasets from Internet resources - development of methods such as tf-idf for info retrieval from documents
- **Machine learning (1990s till 2010s):** Support Vector Machines (SVMs), Bayesian methods, Latent Semantic Analysis (LSA) - but required "feature engineering"
- 2003: "**A neural probabilistic language model**" by Bengio et al.
 - Introduced the power of neural networks and word embeddings in NLP, started current era of NLP



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

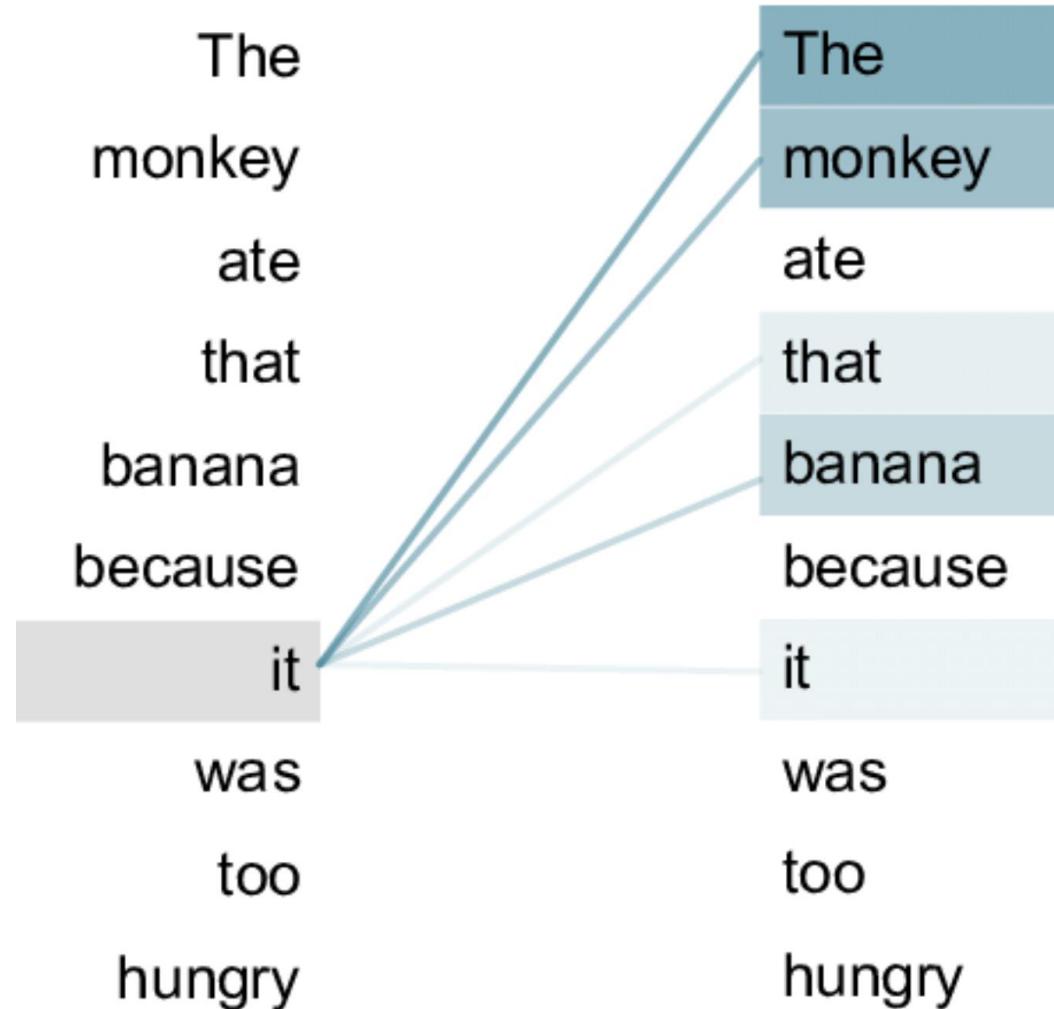
lukaszkaiser@google.com

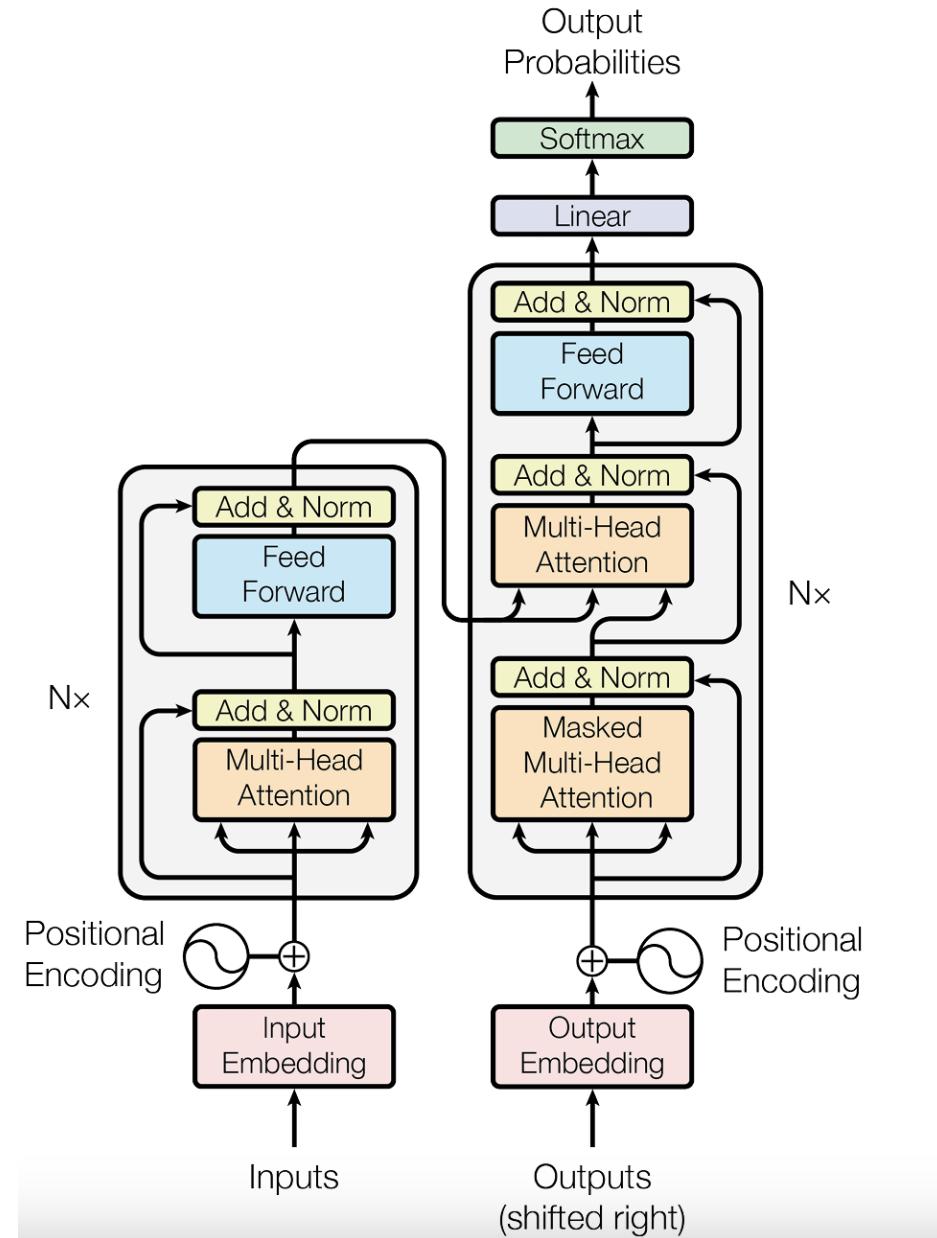
Illia Polosukhin* ‡

illia.polosukhin@gmail.com



Fundamental idea behind all current LLMs: self-attention

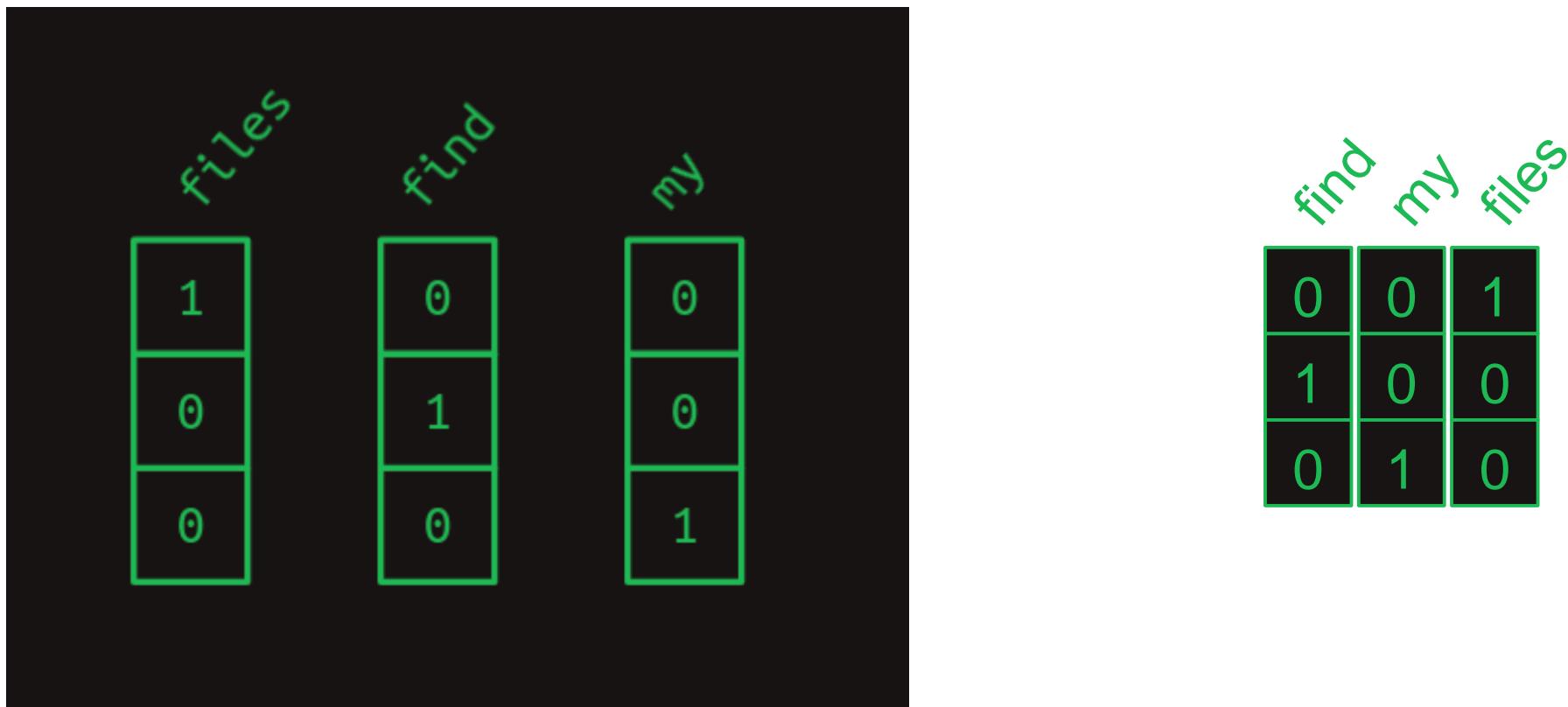






How to represent words as vectors?

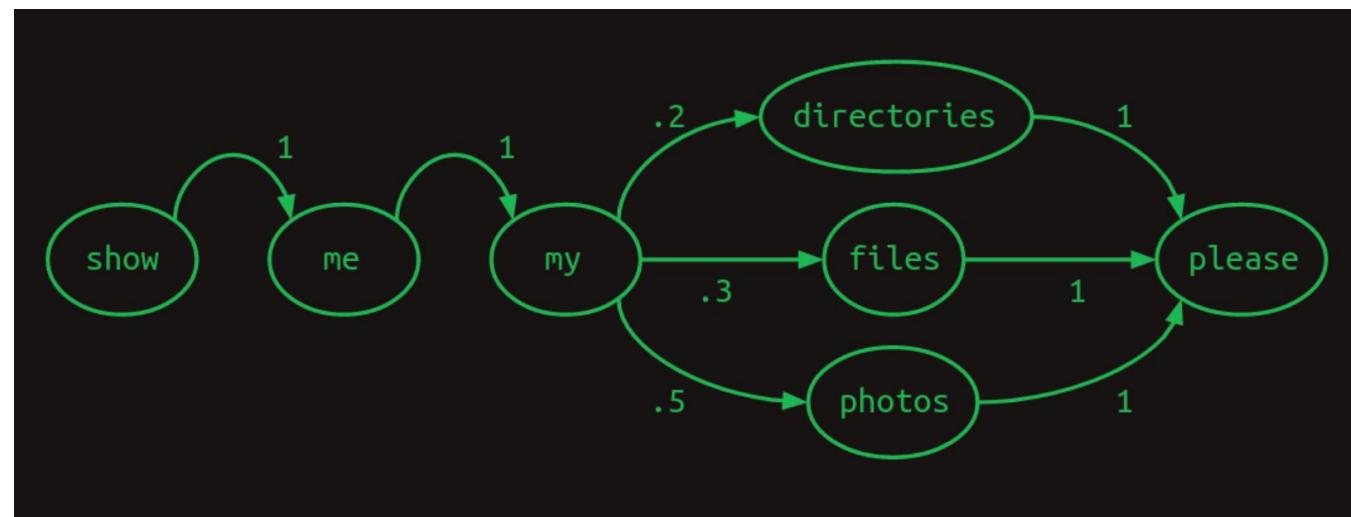
- Simplest approach: **one-hot encoding**
 - Just a “pedagogical” starting point – this is NOT how it is really done!





Modeling sequences of words: first-order Markov model

- Vocabulary: {directories, files, me, my, photos, please, show}
- Represent possible sequences of words with a Markov Chain



Show me my directories please.
Show me my files please.
Show me my photos please.

	directories	files	me	my	photos	please	show
directories	0	0	0	0	0	1	0
files	0	0	0	0	0	1	0
me	0	0	0	1	0	0	0
my	.2	.3	0	0	.5	0	0
photos	0	0	0	0	0	1	0
please	0	0	0	0	0	0	0
show	0	0	1	0	0	0	0



Predicting the next word in first-order model

- Inner product of a word encoding with transition matrix
 - Inner product "selects" the corresponding row of the transition matrix

The diagram illustrates the prediction process using a word embedding vector and a transition matrix.

Word Embedding Vector:

0	0	0	1	0	0	0
directories	files	me	ny	photos	please	show

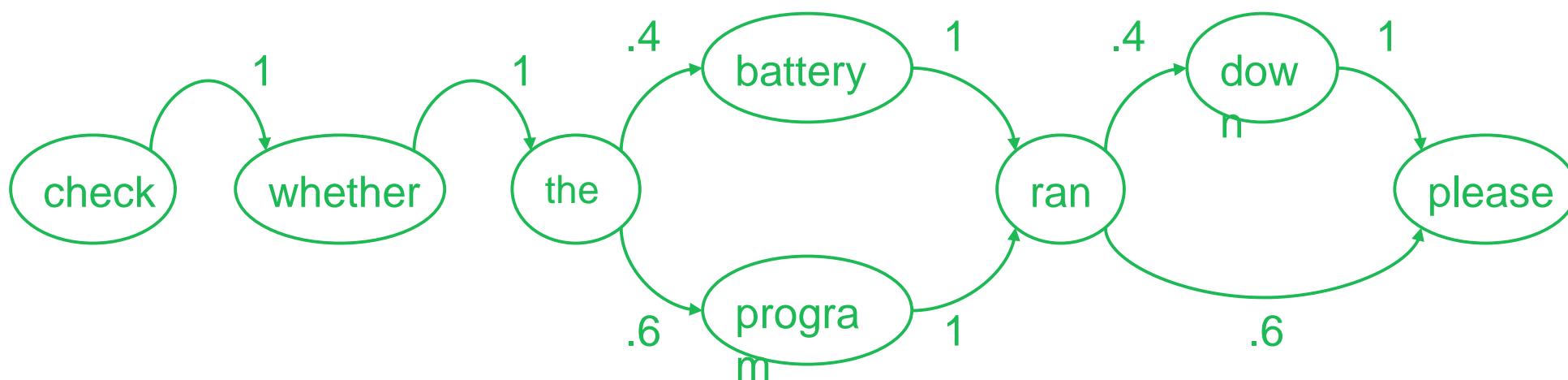
Transition Matrix:

0	0	0	0	0	1	0
0	0	0	0	0	1	0
0	0	0	1	0	0	0
.2	.3	0	0	.5	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	0
0	0	1	0	0	0	0

Result:

.2	.3	0	0	.5	0	0
directories	files	me	ny	photos	please	show

- Suppose the only sentences in our text input are:
 - "Check whether the battery run down please" (40% occurrence)
 - "Check whether the program run please" (60% occurrence)
- What is the problem with the next Markov chain?





Second-order model

- Predict next word from combination of last two words
 - But the transition matrix now has N^2 rows (N : number of words)

	battery	check	down	please	program	ran	the	whether
battery ran	0	0	1	0	0	0	0	0
check whether	0	0	0	0	0	0	1	0
program ran	0	0	0	1	0	0	0	0
the battery	0	0	0	0	0	1	0	0
the program	0	0	0	0	0	1	0	0
ran down	0	0	0	1	0	0	0	0
whether the	.4	0	0	0	.6	0	0	0
.	0	0	0	0	0	0	0	0

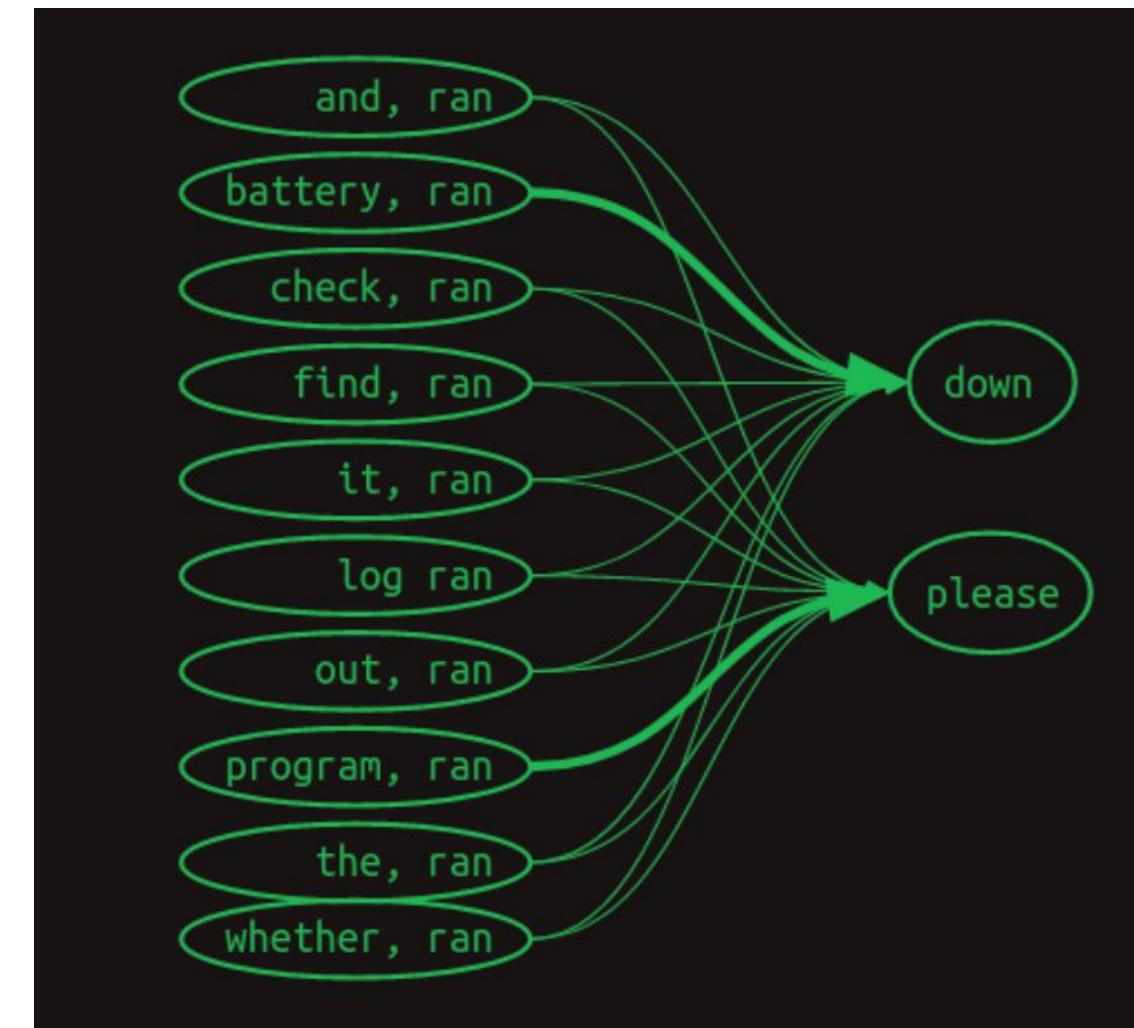


What about long-range dependencies?

- "Check the **program** log and find out whether it **ran** please."
- "Check the **battery** log and find out whether it **ran** down please."

- Need **to abandon Markov models**

- Predict next word based on different combinations of current+earlier words
- Think of each word-combination as a "weighted vote" for a particular next word
- Word-combinations: features
 - Some informative, some not





New issue: lots of uninformative word combinations

- "Check the **program** log and find out whether it **ran** please."
- "Check the **battery** log and find out whether it **ran** down please."

	and	battery	check	down	find	it	log	out	please	program	ran	the	whether
and, ran				.5					.5				
battery, ran					1						0		
check, ran				.5					.5				
down, ran													
find, ran				.5					.5				
it, ran				.5					.5				
log, ran				.5					.5				
out, ran				.5					.5				
please, ran													
program, ran					0						1		
ran, ran													
the, ran				.5					.5				
whether, ran				.5					.5				



How to identify informative features? **Attention**



- A “masking vector” identifies the only features that we should pay **attention** to

$$\begin{array}{l} \text{feature} \\ \text{activities} \end{array} \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ \hline \end{array} \quad x \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline \end{array} \\ \text{mask} \\ \quad = \\ \begin{array}{l} \text{masked} \\ \text{feature} \\ \text{activities} \end{array} \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$



What do we gain by using Masking (or Attention)?

- The transition matrix becomes sparse - only few word combinations matter

	and	battery	check	down	find	it	log	out	please	program	ran	the	whether
and, ran			.5					.5					
battery, ran				1					0				
check, ran				.5					.5				
down, ran													
find, ran			.5						.5				
it, ran			.5						.5				
log, ran			.5						.5				
out, ran			.5						.5				
please, ran													
program, ran				0					1				
ran, ran													
the, ran			.5						.5				
whether, ran			.5						.5				

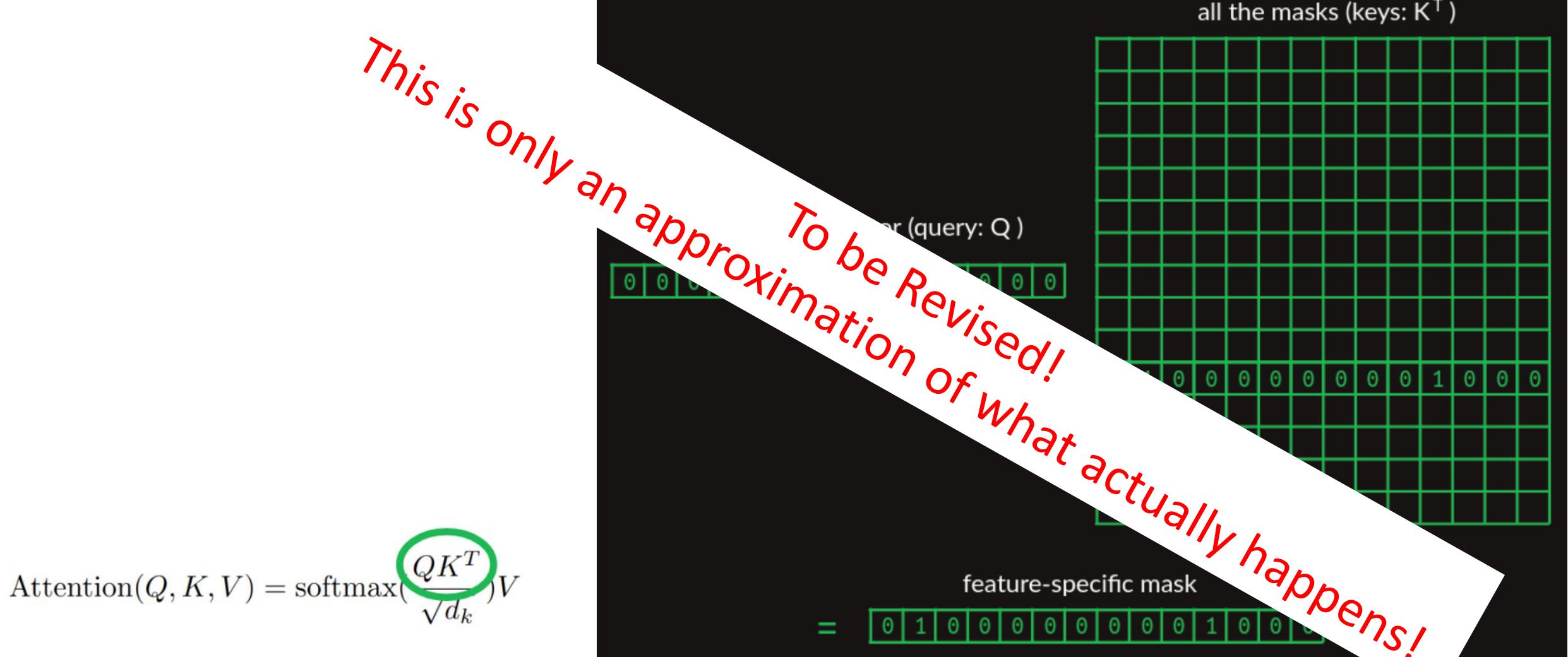
	and	battery	check	down	find	it	log	out	please	program	ran	the	whether
and, ran													
battery, ran									1				0
check, ran													
down, ran													
find, ran													
it, ran													
log, ran													
out, ran													
please, ran													
program, ran										0			1
ran, ran													
the, ran													
whether, ran													

W2



How to select the Attention ("mask") vector for a given word ("query")

- "Check the **program** log and find out whether it **ran** please."
- "Check the **battery** ↗ and find out whether it **ran** down please."

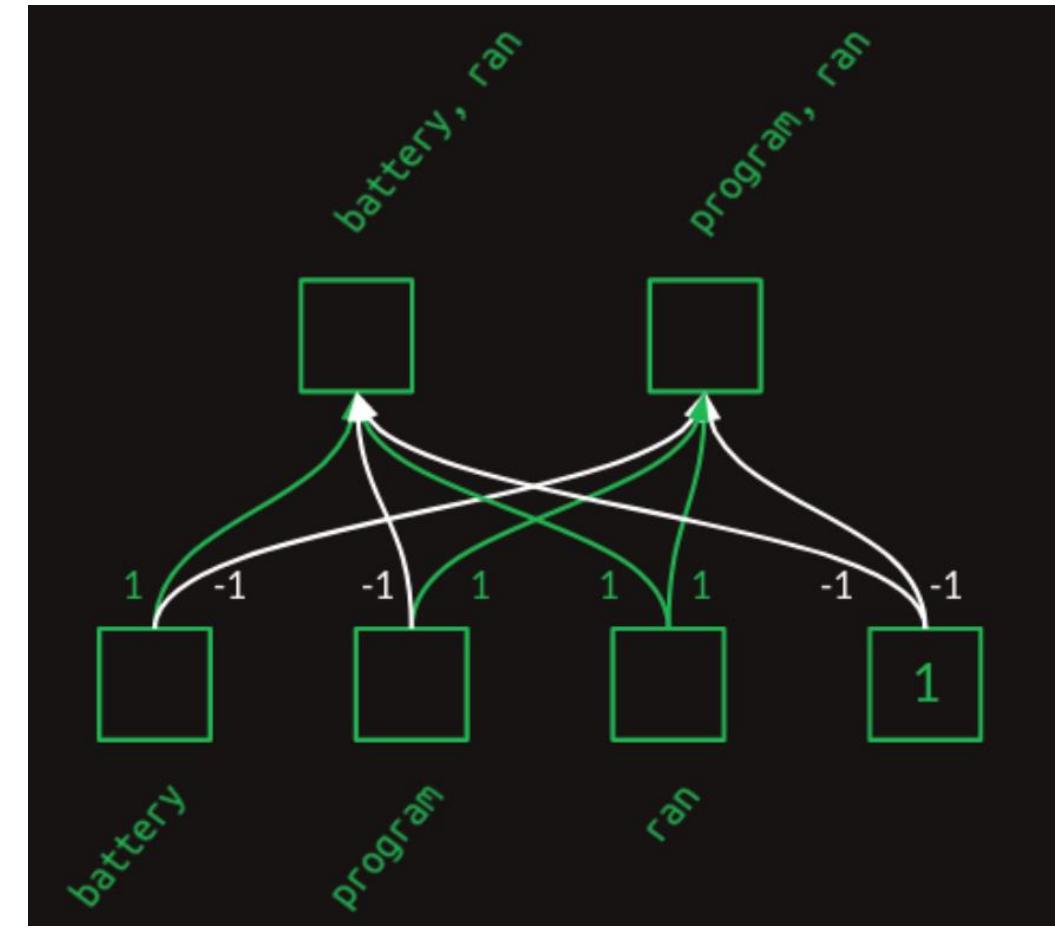




And how do we get the feature vectors from the attention vector?

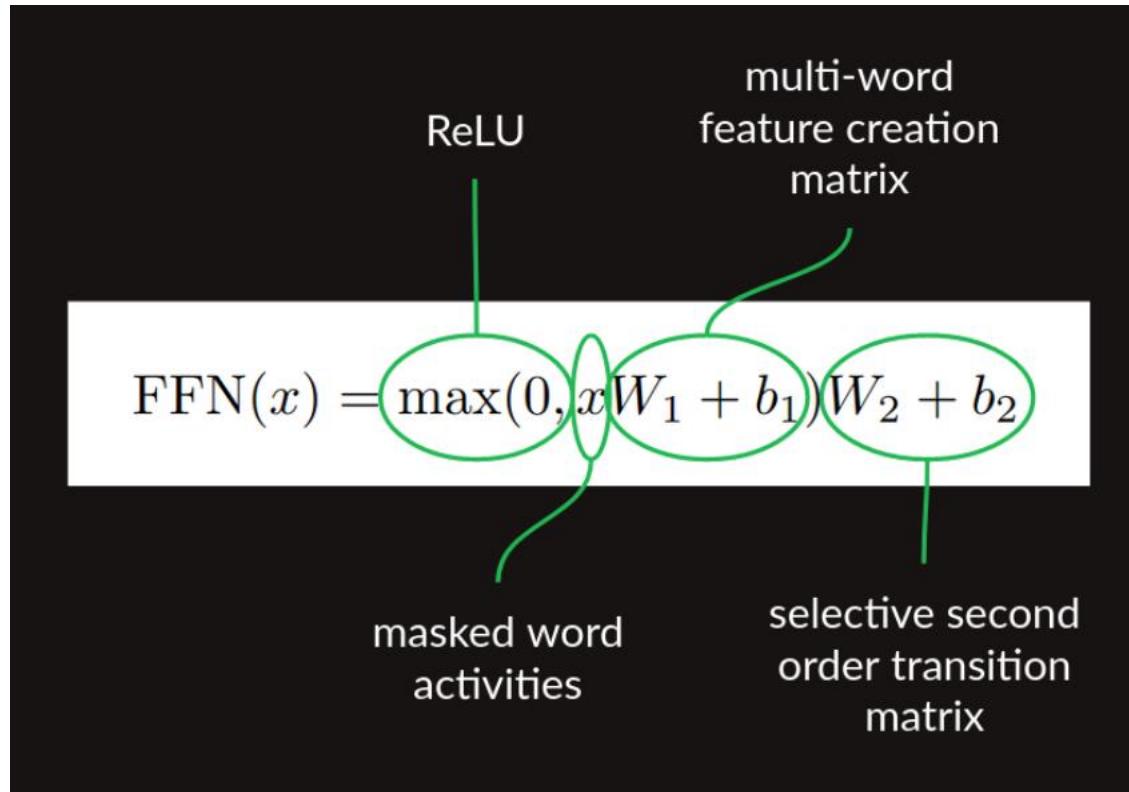
$$\begin{matrix} \text{battery} \\ \text{program} \\ \text{run} \\ \text{bias} \end{matrix} \quad \begin{matrix} 0 & 1 & 1 & 1 \end{matrix}$$
$$\begin{matrix} 1 & -1 \\ -1 & 1 \\ 1 & 1 \\ -1 & -1 \end{matrix} = \begin{matrix} \text{battery, run} \\ \text{program, run} \end{matrix}$$

- Use a single-layer fully-connected network
 - Ex: two-word features

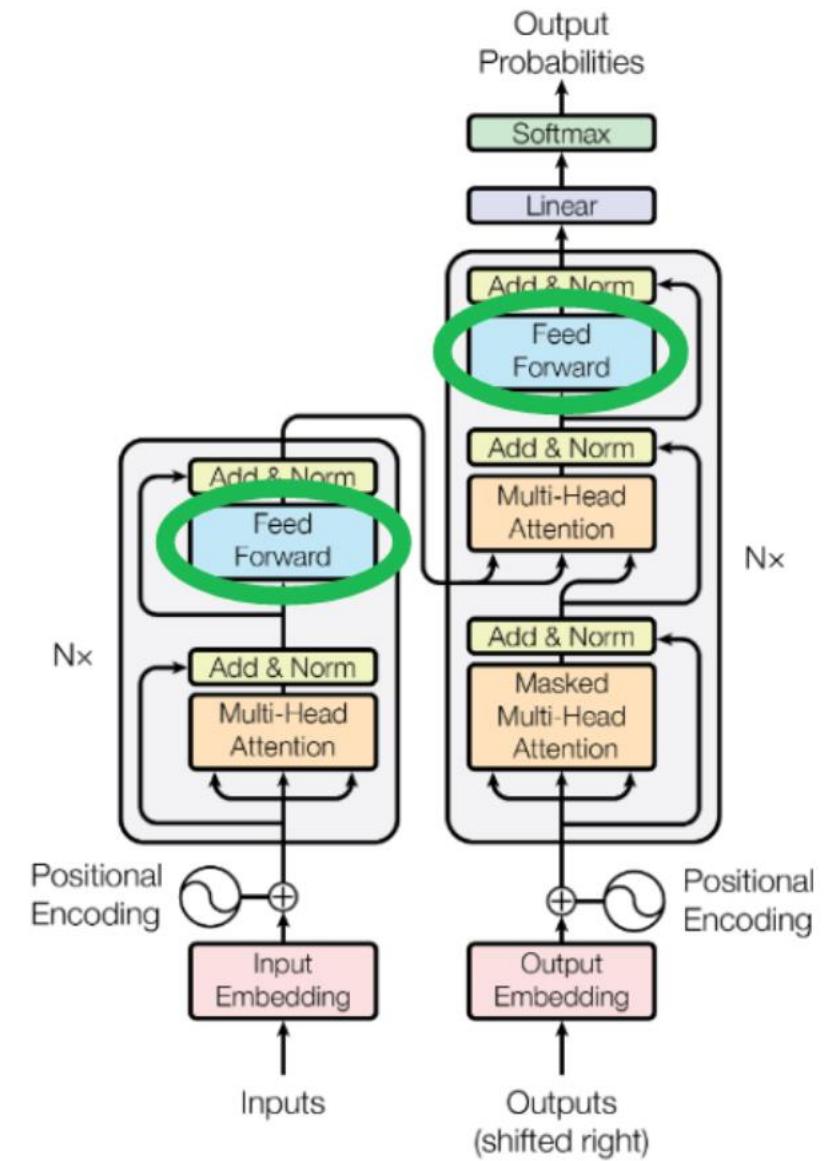




How does this map to the Transformer architecture?



feature creation matrix multiplication,
ReLU nonlinearity, and
transition matrix multiplication



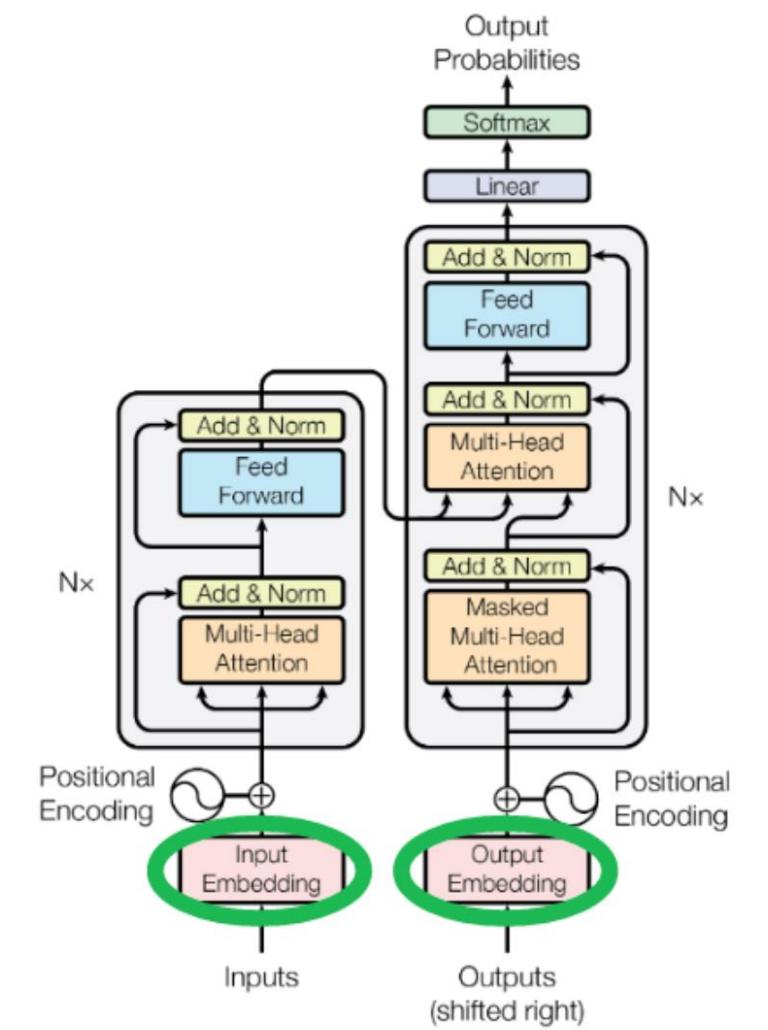
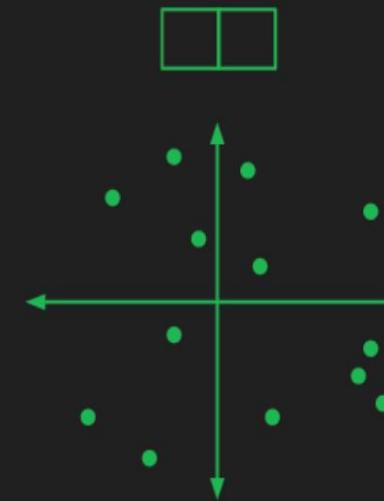


What's the problem with one-hot encodings?

- For N words, the second-order transition matrix would be $N^2 \times N$
 - $N=50,000$ words $\rightarrow 100$ trillion elements..



embedding
in 2 dimensions

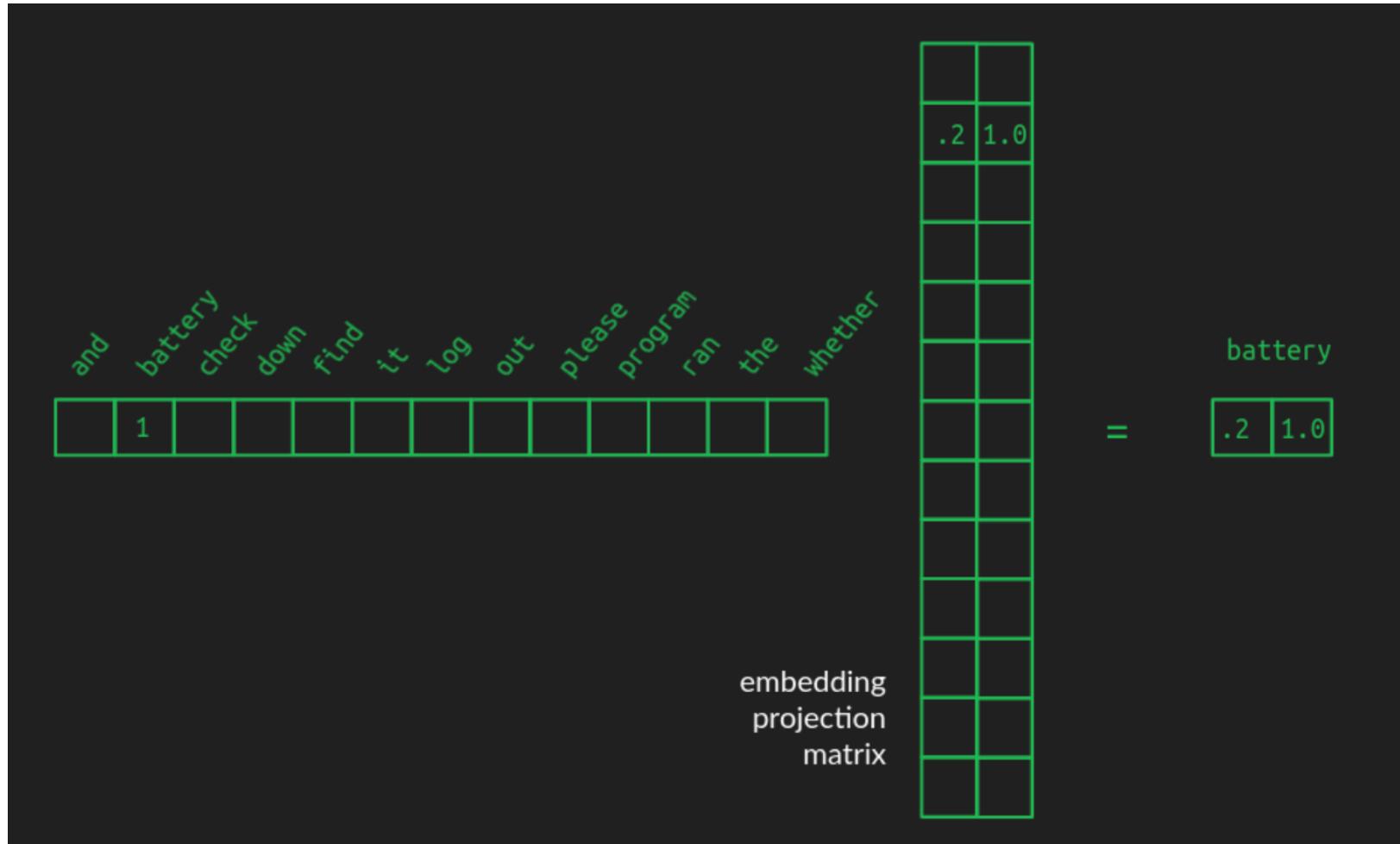




Word embeddings

- Projections from N-dimensional space to M-dimensions ($M \ll N$) so that similar words are closer

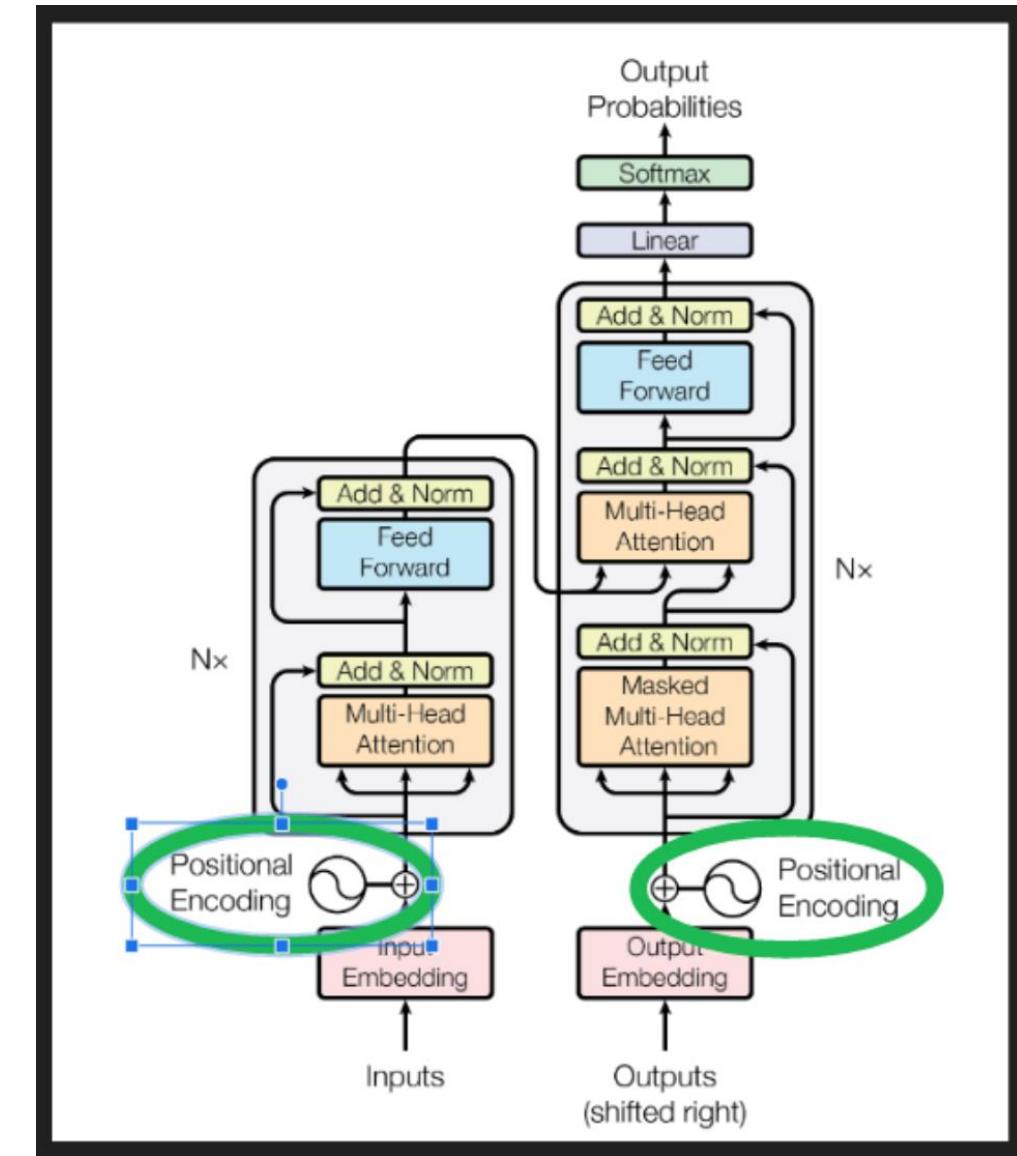
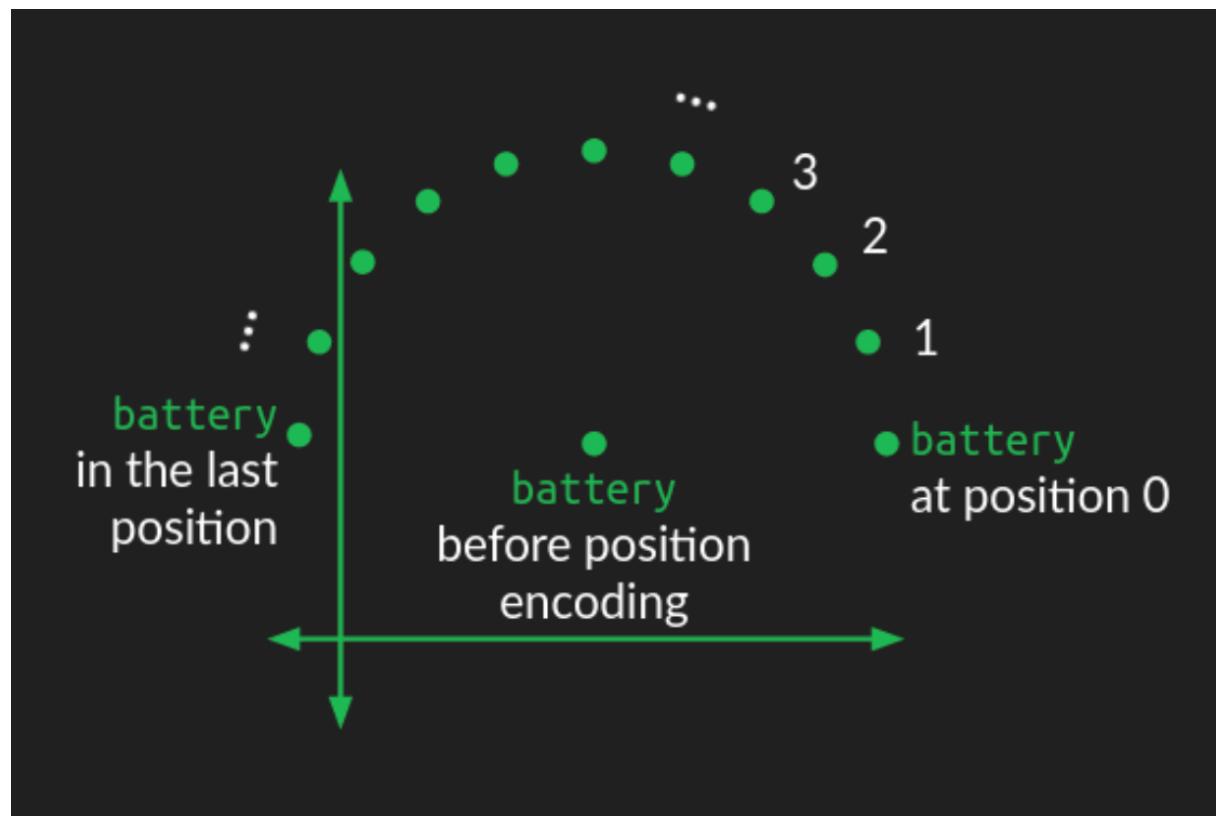
- Eg.: $M=2$





Positional encoding

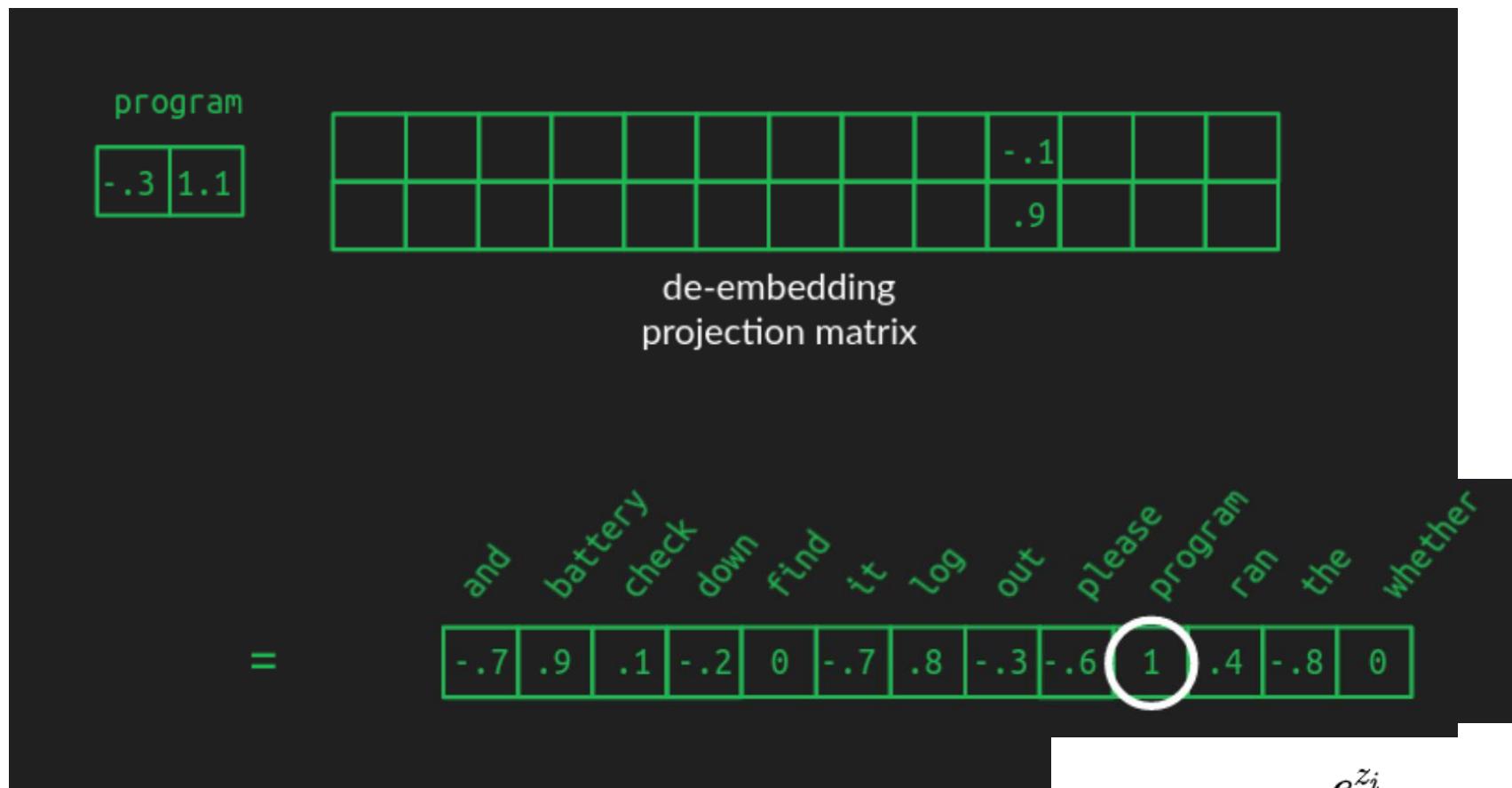
- Represent the position of a word by rotating its embedding based on its location in the sequence
 - Eg.: M=2



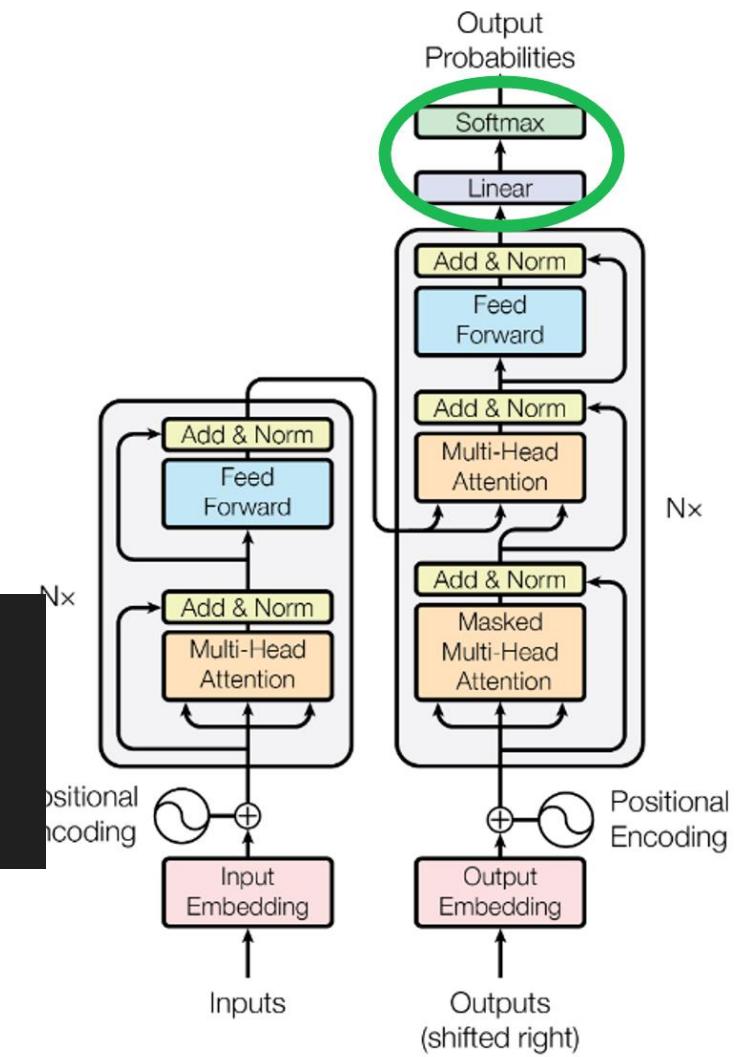


How to go from low-dimensional embeddings back to words?

- Need the opposite projection
 - but that may not result in sparse (one-shot) encoding of output word



$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

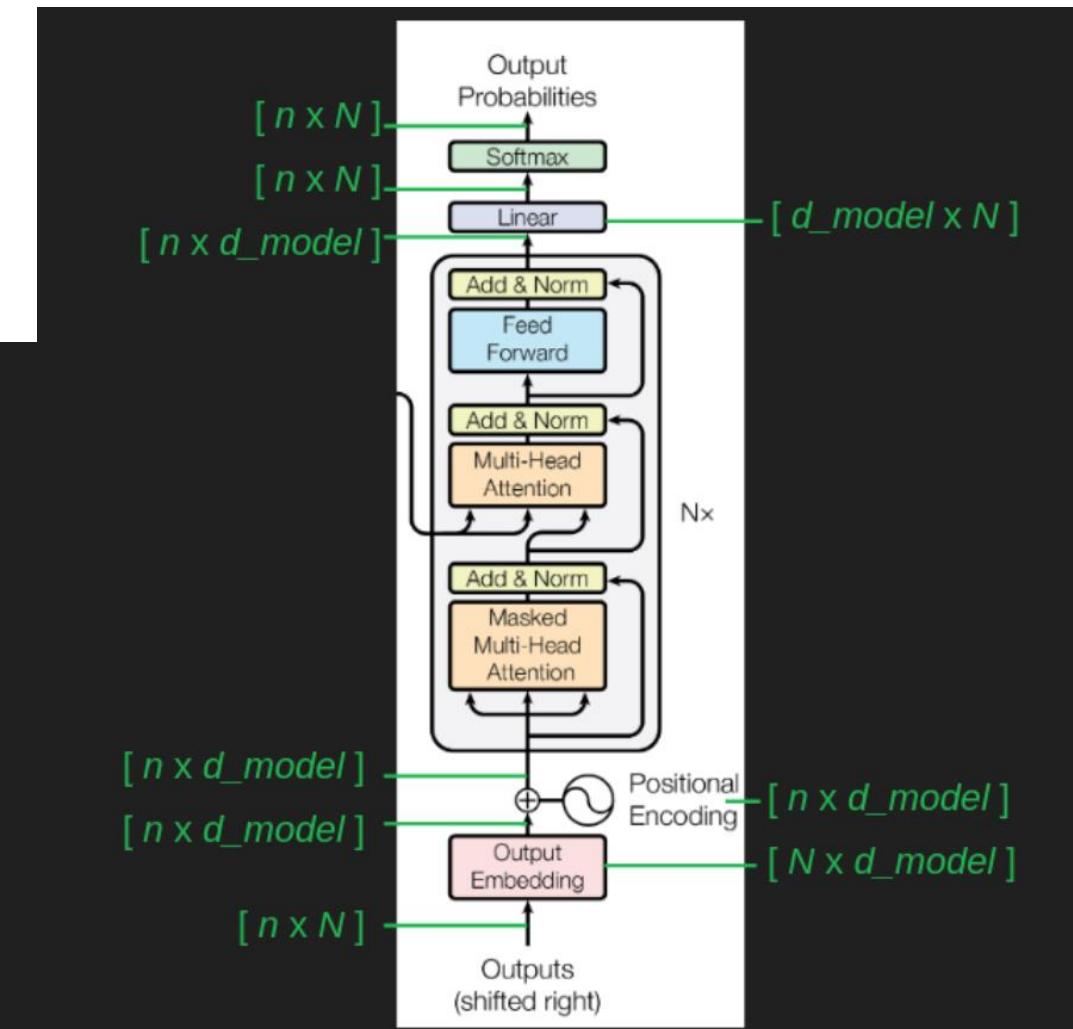
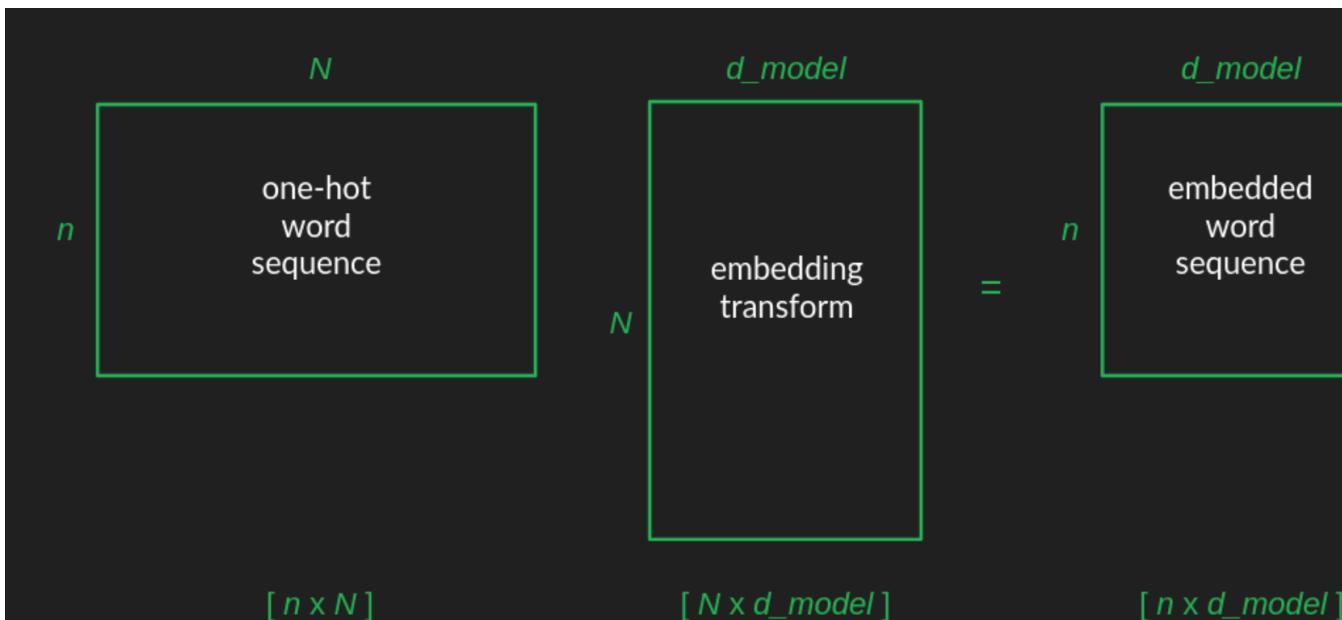


- Why do we use softmax instead of argmax?



Some notation about dimensionality

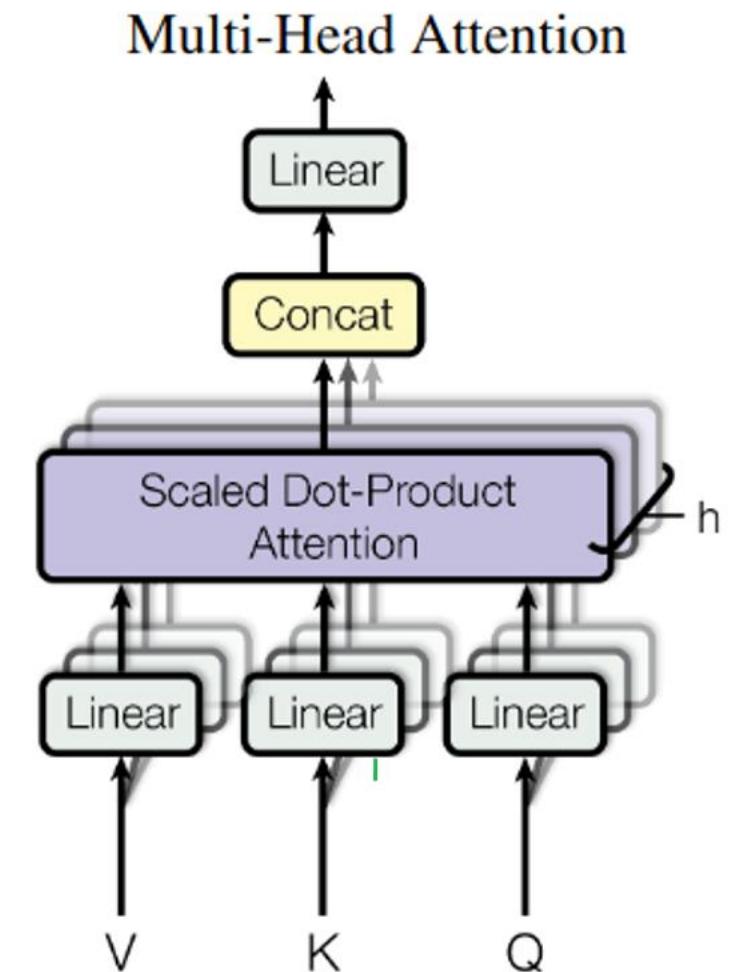
- N : vocabulary size (in the thousands)
- n : max sequence length (ex. 2000 tokens)
- d_{model} : embedding dimensions (ex. 512)





Multiple Attention Heads?

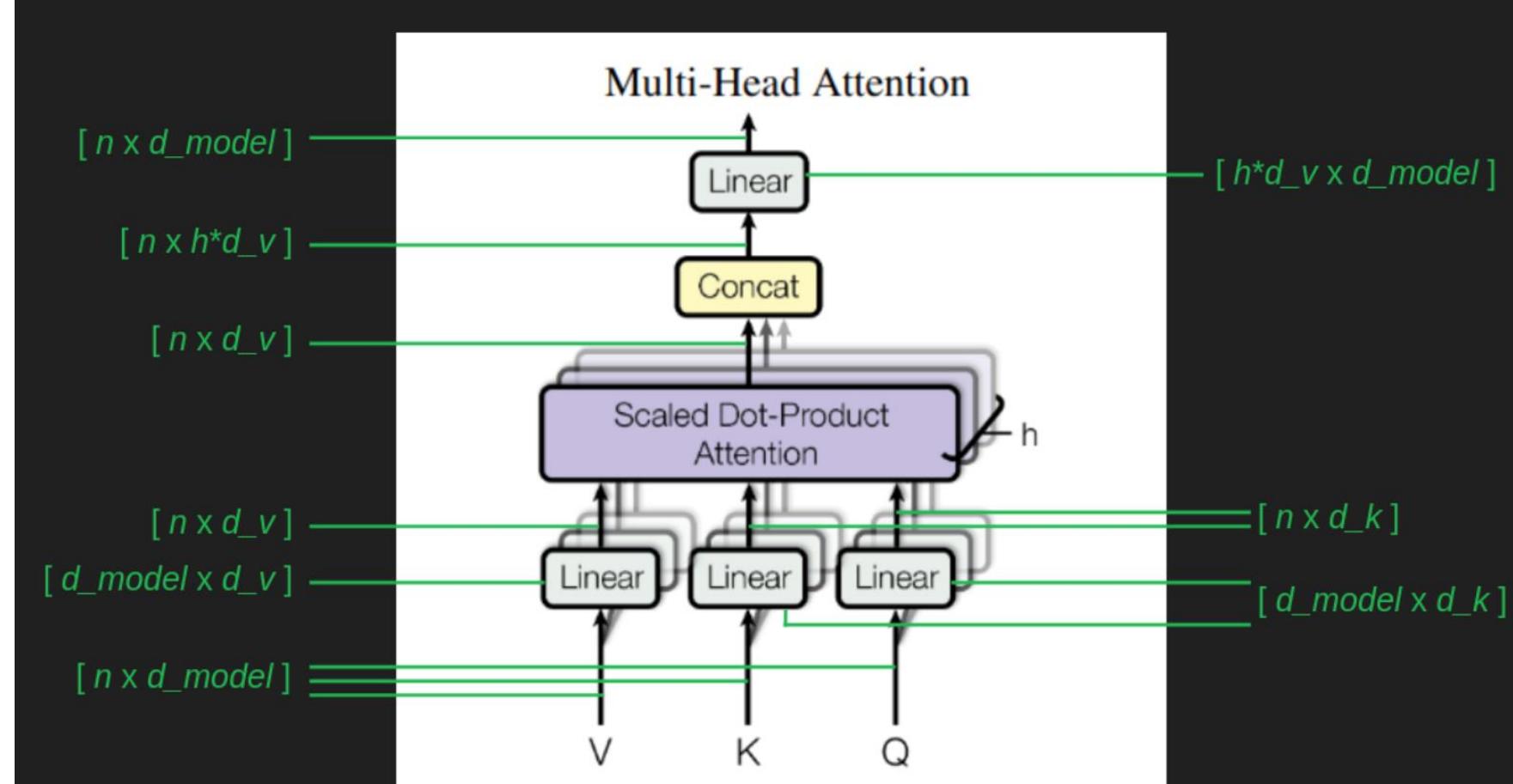
- Softmax tends to produce only one value close to 1
 - but we may need to focus on more than one previous tokens
- h : number of attention heads
- How can we reduce the computational overhead?





Multiple Attention Heads?

- h : number of attention heads
- d_v : dimensions for representing Values
- d_k : dimensions for representing Keys & Queries



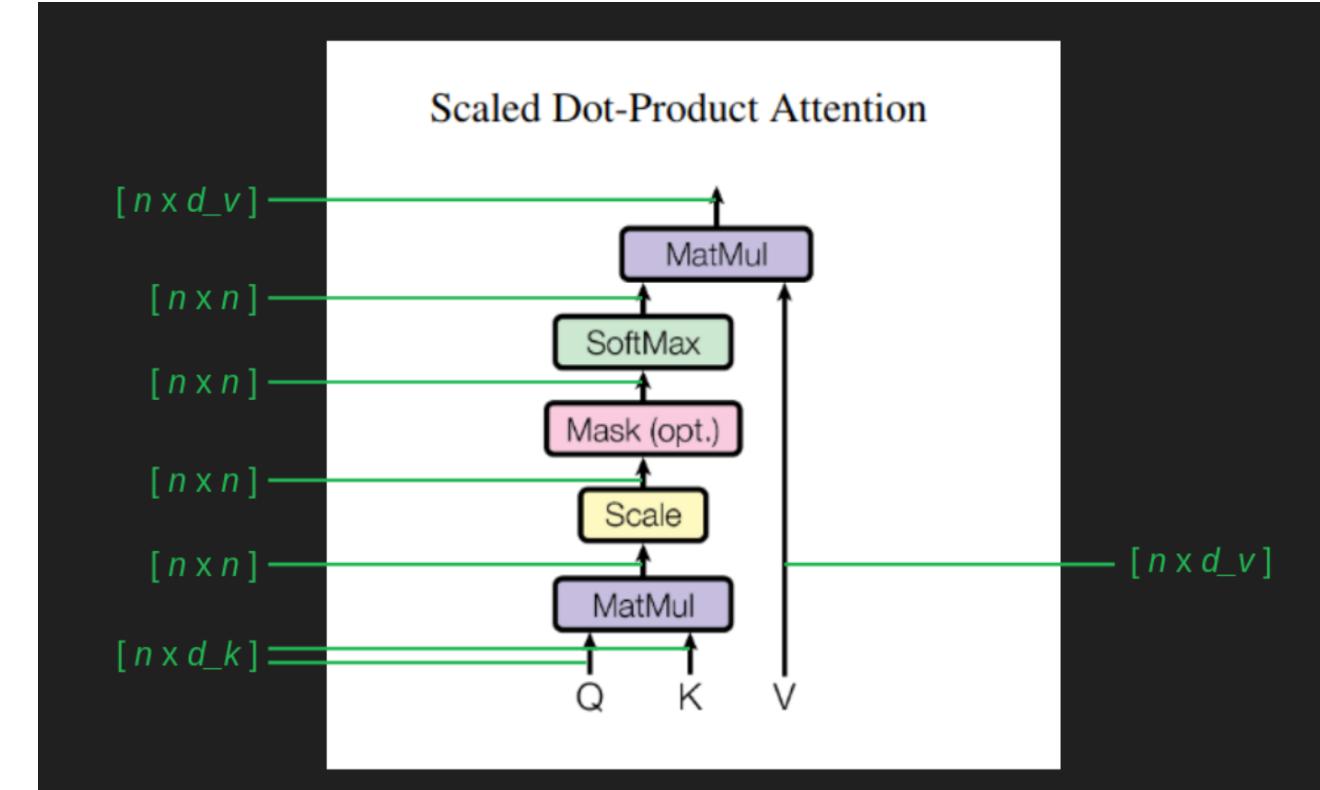
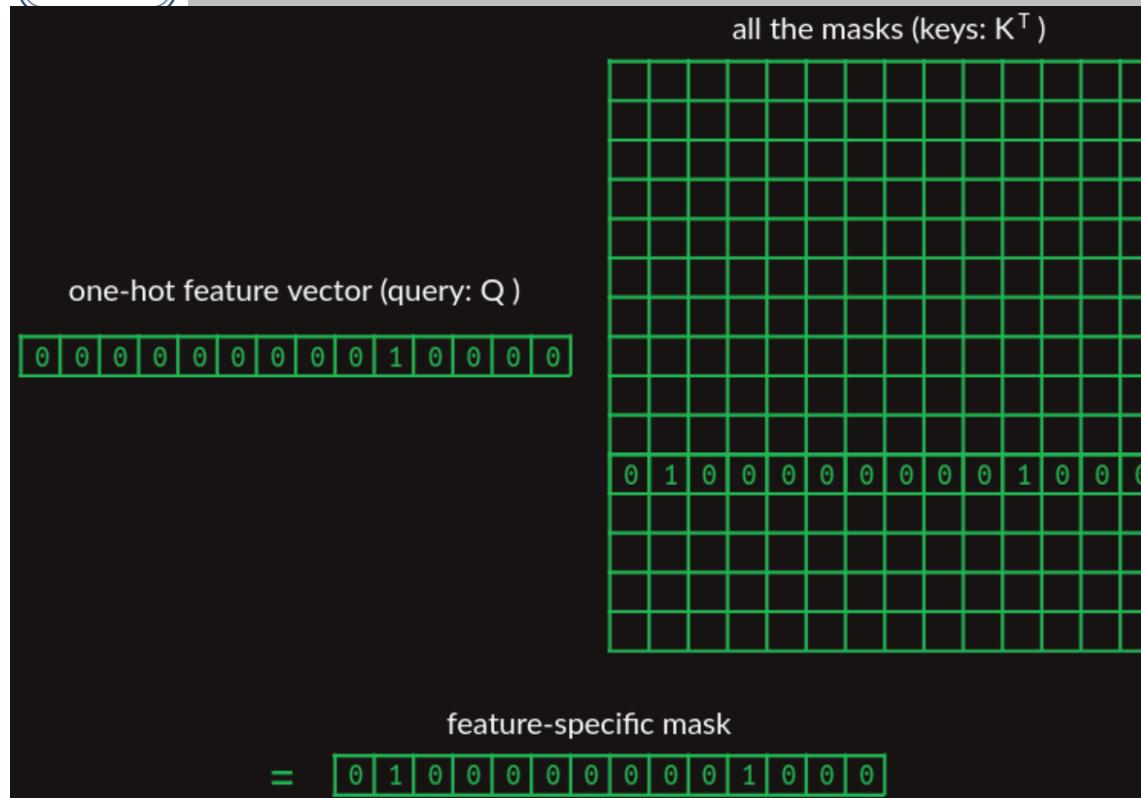
$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$

Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$.



Single attention head -- revisited

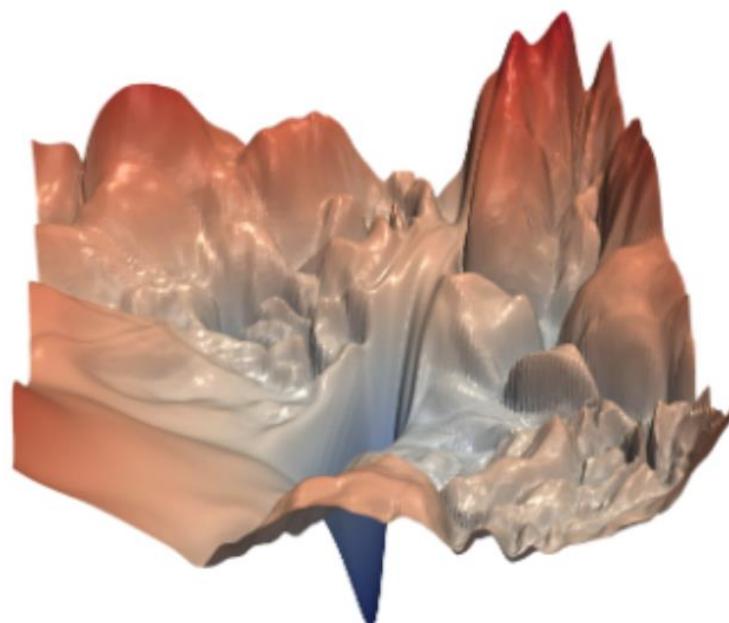


- Attention computations are performed in embedding subspaces (i.e., not just 0s and 1s)
- Query does not correspond to single word - but group of semantically related words
- Attention: position-to-position relationships (not word-to-word relationships)
- Masking (for sequence completion tasks): attention does not consider positions that have not been generated yet

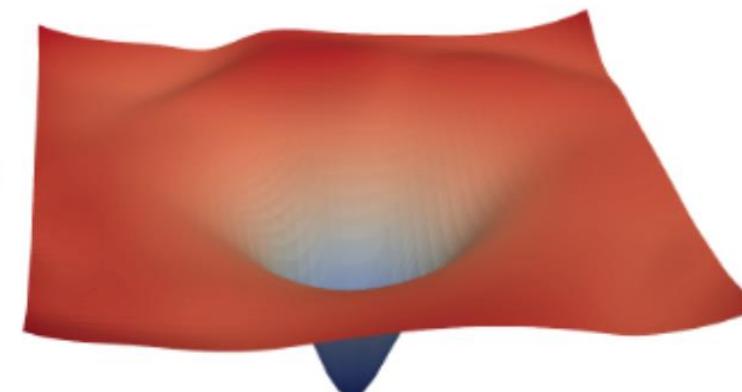


What about those skip connections?

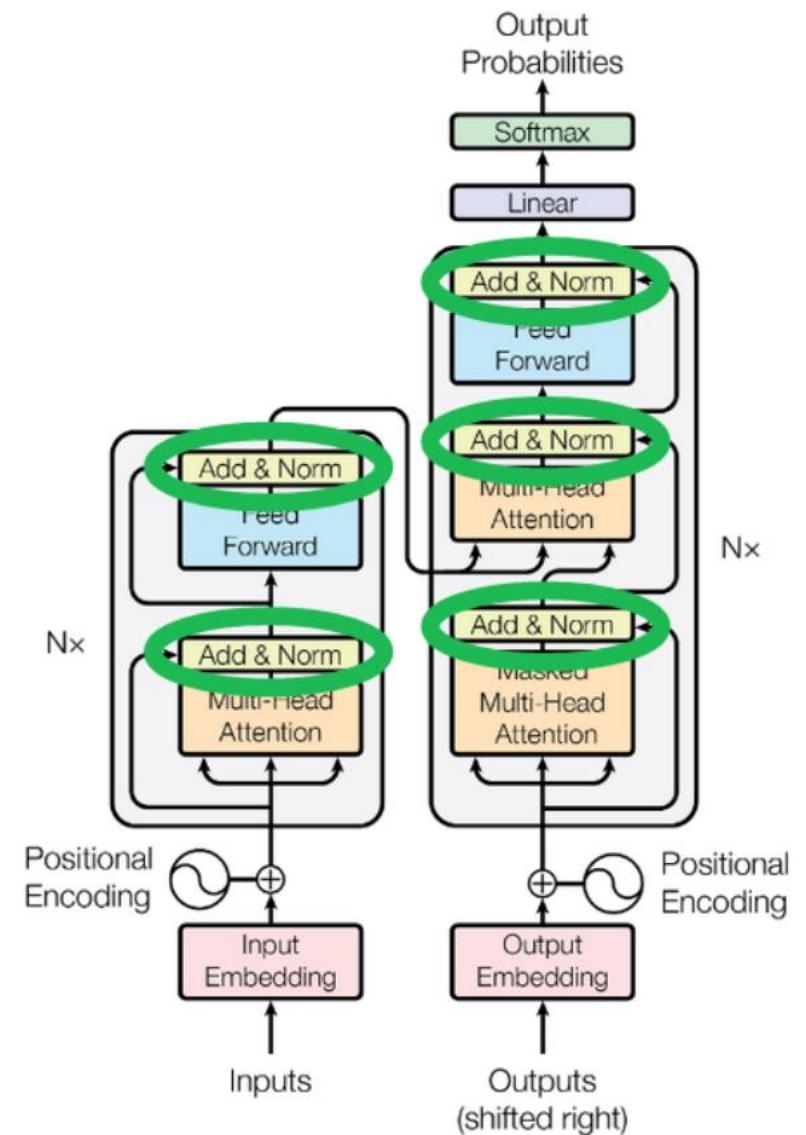
- It is important to attend to the current word as well - not only earlier words
- Skip connections help smoothen the optimization landscape
 - Attention module may filter out everything - zero output?



(a) without skip connections

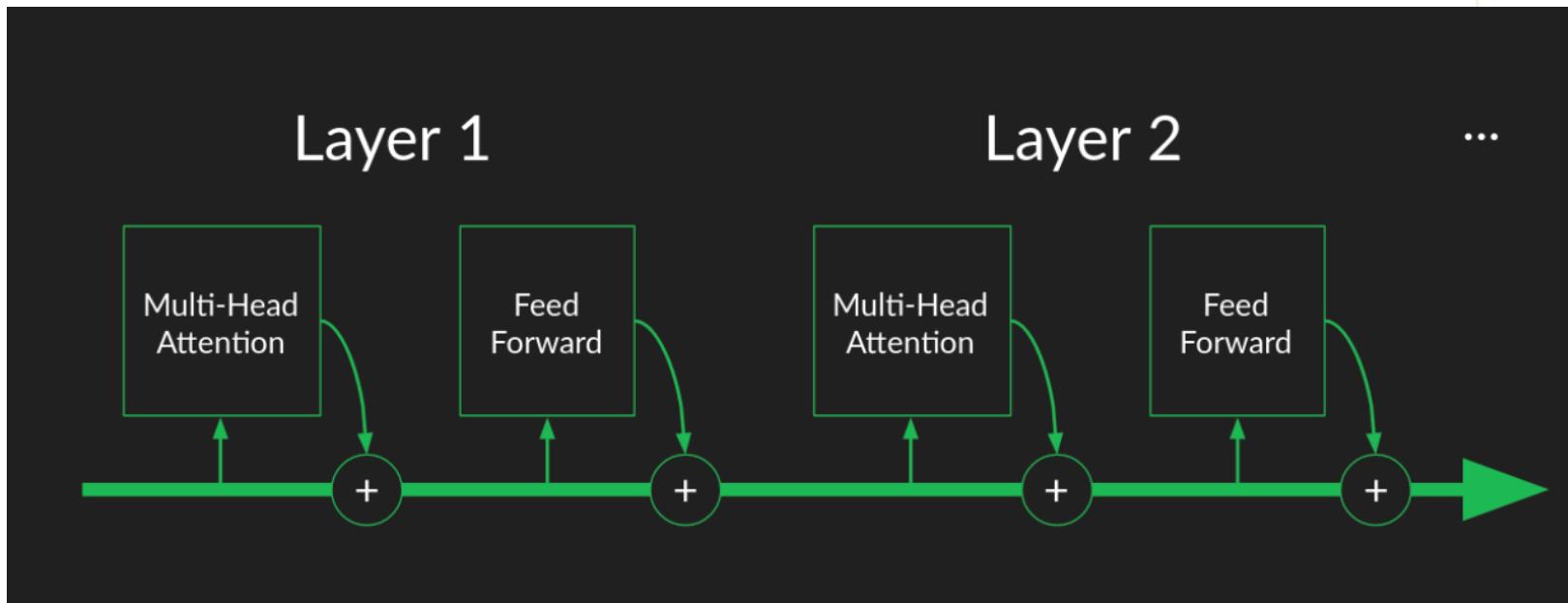


(b) with skip connections

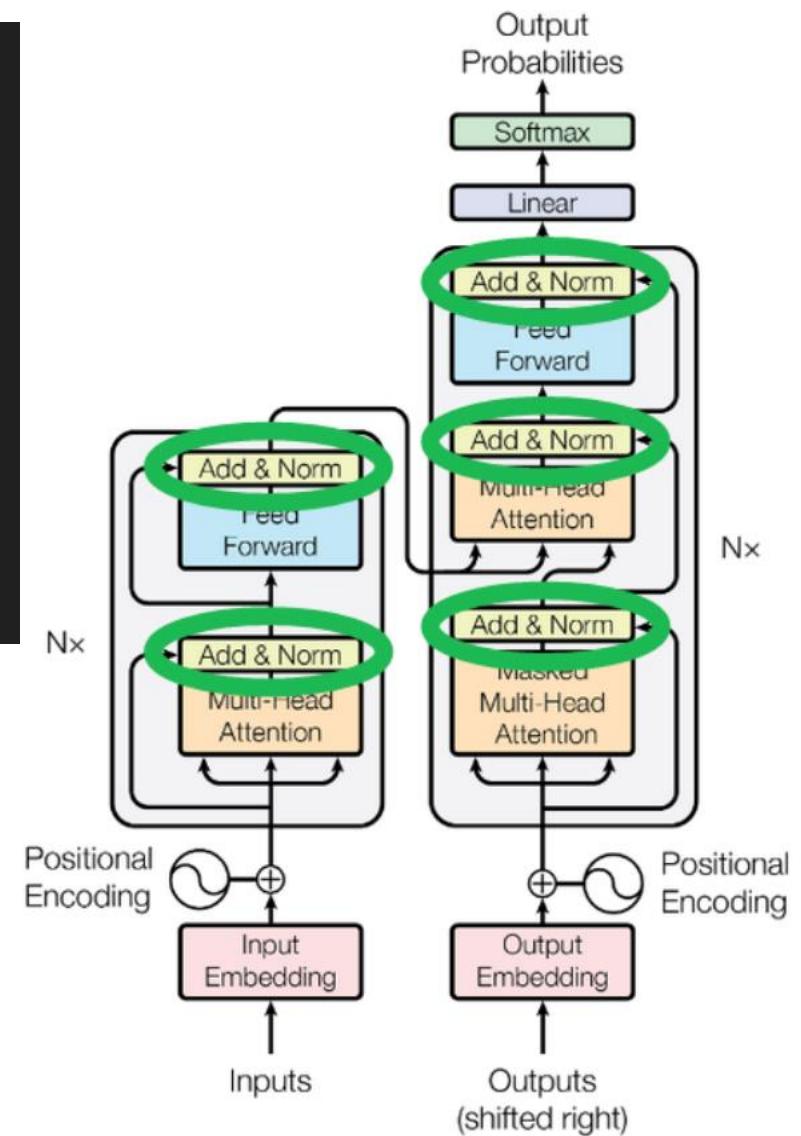




What about multiple layers?



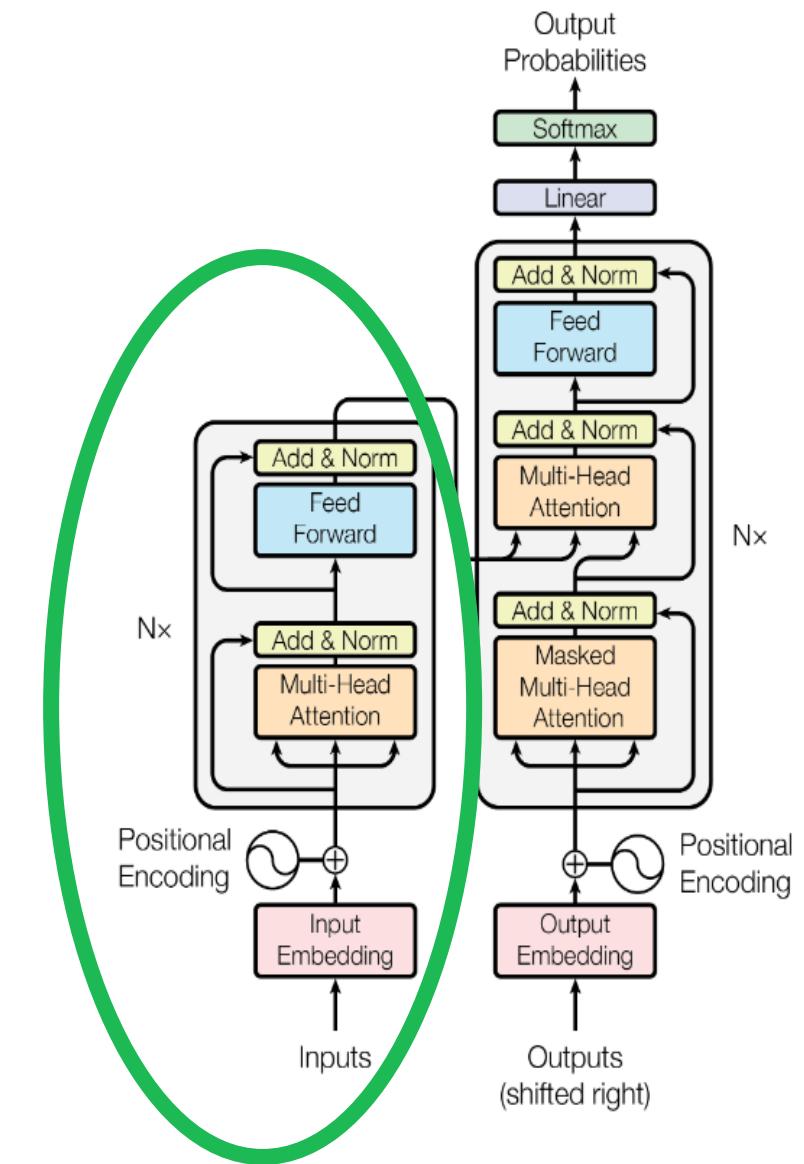
- **But note:** the use of several layers here is different than in deep learning -- hierarchical feature extractors
 - Here, each layer is one more "worker at a conveyor belt"





What does the encoder do?

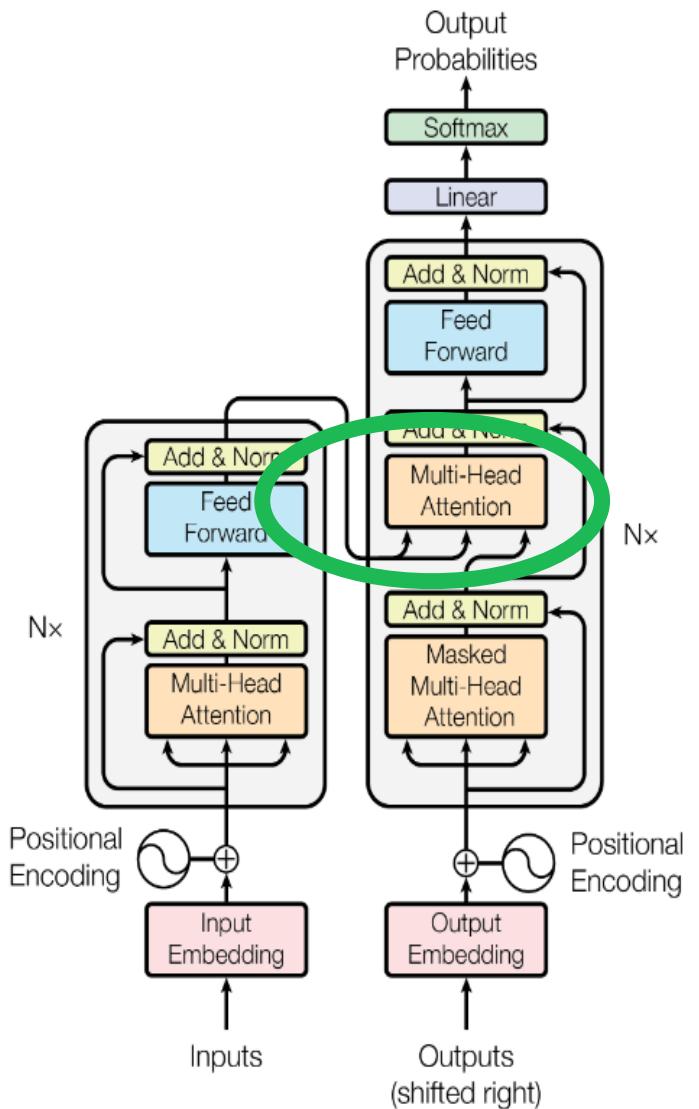
- In a sequence transformation task, where the entire input sequence is given, it provides the input to each decoder layer
 - No masking in that case
 - E.g., machine translation
- In sequence completion, it processes the prompt to generate a semantic representation of the desired output
 - See next slide





Cross Attention: encoder-decoder connections

- Self-attention versus cross-attention
- The next token should depend not only to the previously generated output - but also on the given input
 - In cross-attention, Keys and Values matrices depend on output of encoder's final layer
 - Same embedded source sequence given to every decoder layer





Part-B: what we do NOT understand yet

Ok, we understand how transformers and attention work, right?

But how is it possible that ChatGPT appears to UNDERSTAND language?



Emerging GPT-4 capability: understanding of basic physics?

Prompt: Here we have a book, 9 eggs, a laptop, a bottle and a nail. Please tell me how to stack them onto each other in a stable manner.

GPT-4:

One possible way to stack the objects onto each other in a stable manner is:

- Place the book flat on a level surface, such as a table or a floor. The book will serve as the base of the stack and provide a large and sturdy support.
- Arrange the 9 eggs in a 3 by 3 square on top of the book, leaving some space between them. The eggs will form a second layer and distribute the weight evenly. Make sure the eggs are not cracked or broken, and handle them gently to avoid dropping them.
- Place the laptop on top of the eggs, with the screen facing down and the keyboard facing up. The laptop will fit snugly within the boundaries of the book and the eggs, and its flat and rigid surface will provide a stable platform for the next layer. Make sure the laptop is closed and turned off, and do not press any keys or buttons.
- Place the bottle on top of the laptop, with the cap facing up and the bottom facing down. The bottle will add some height and balance to the stack, and its cylindrical shape will reduce the contact area with the laptop. Make sure the bottle is empty and closed, and do not shake or tilt it.
- Place the nail on top of the bottle cap, with the pointy end facing up and the flat end facing down. The nail will be the final and smallest object in the stack, and its sharp tip will prevent it from rolling or sliding off the bottle. Make sure the nail is clean and straight, and do not touch or move it.



GPT-4 capabilities: "theory of mind" and emotional intelligence?

Scenario:

Tom just lost his ZURFIN, but he didn't tell Adam.

Adam starts talking to him about how much he loves his ZURFIN, and notices that Tom makes a sad face.

--

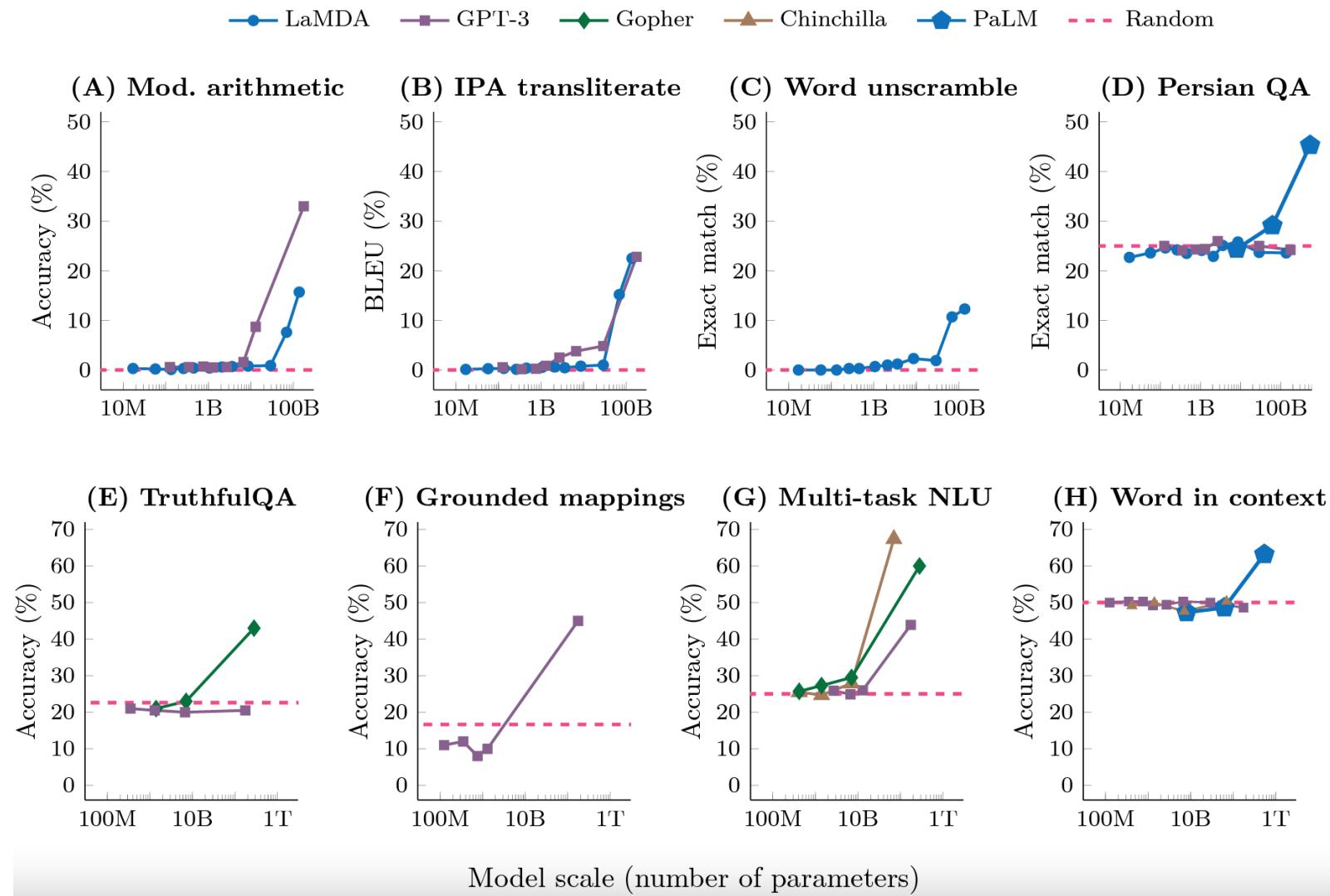
Q: Why is Tom making a sad face?<|endofprompt|>

A: Tom is making a sad face because he just lost his ZURFIN, and hearing Adam talk about his ZURFIN reminds him of what he is missing. Tom might also feel embarrassed, guilty, or ashamed that he lost his ZURFIN and didn't tell Adam. Tom might also worry that Adam will be angry or disappointed if he finds out.

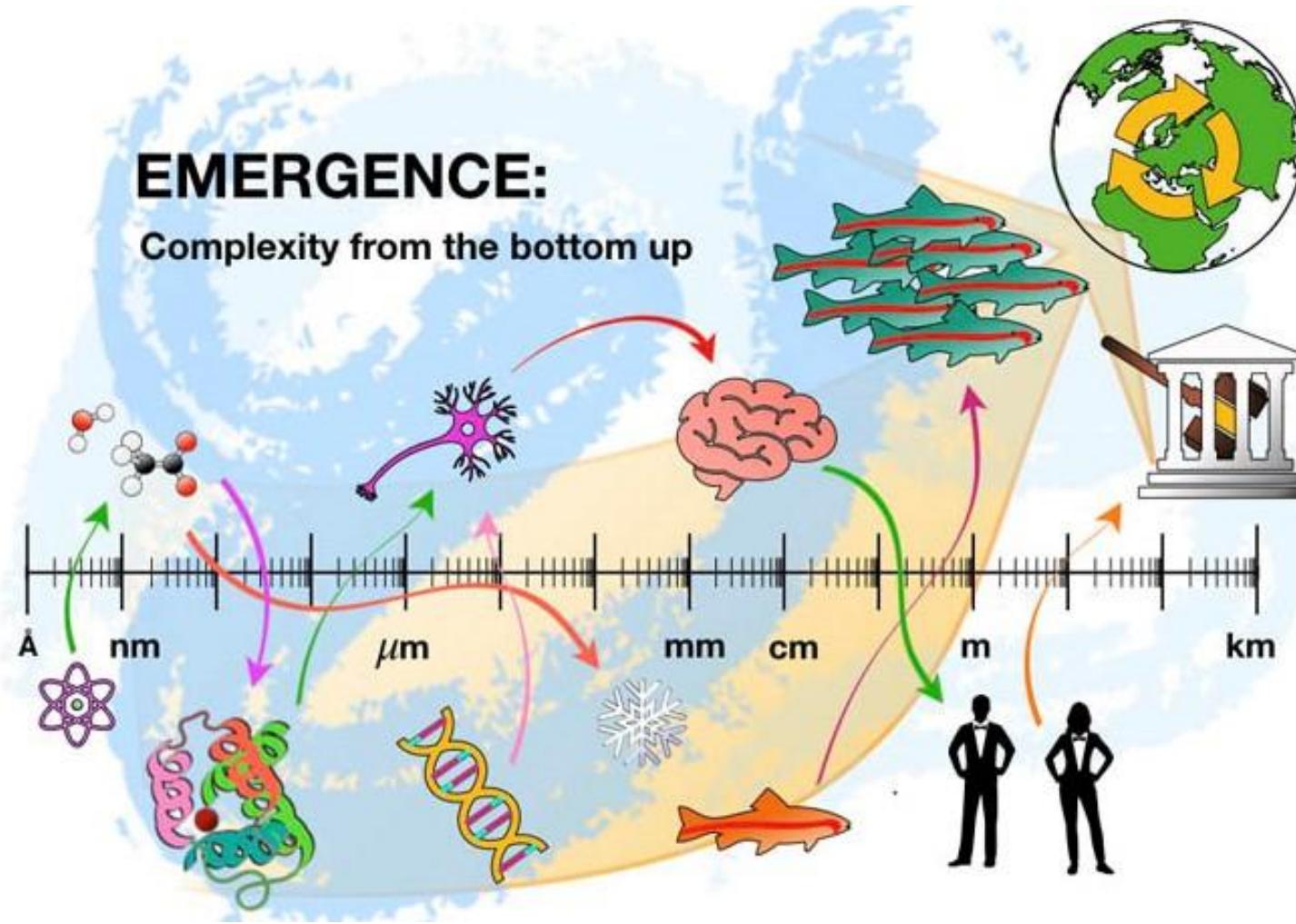
Q: What does Adam think causes Tom's sad face?

A: Adam might not know what causes Tom's sad face, since he doesn't know that Tom lost his ZURFIN. He might think that Tom is bored, or tired, or sick, or unhappy about something else. He might also wonder if Tom doesn't like his ZURFIN, or if he has a problem with Adam.

An ability is emergent if it is not present in smaller models but is present in larger models.



- *"Emergence is when quantitative changes in a system result in qualitative changes in behavior." (P. Anderson, 1972, "More is different")*



- In my opinion,
Artificial General Intelligence (AGI)
is already here



Source: "Ex-Machina" film, 2014

- And this will be the greatest invention of our generation
 - Everything will change - have no doubt about it!
 - Many innovation and commercialization opportunities
 - CyI has the potential for major impact in Cyprus and beyond
- An "once in a lifetime" scientific challenge is open for all of us:
 - Understand how AGI emerges from LLMs
 - Make it better, safer, aligned with human values
 - But also, understand what distinguishes the human mind from AGI