Self-Supervised Learning w/ Attention Mechanism

CSCI 601 471/671
NLP: Self-Supervised Models

https://self-supervised.cs.jhu.edu/sp2023/

JOHNS HOPKINS UNIVERSITY

[Slide credit: Chris Tanner, Jacob Devlin and many others]

Logistics Update

- Q: Will there be any normalization ("curve fitting") for the final grades? **Nope**
- Q: Will we have access to large[r] GPUs? **Yes,** Details after the midterm.
- The midterm:
  - will be on March 7 during class time.
  - it will be on paper
  - It will be based on the ideas you have seen in homework and lectures. If you understand them, you're set!
  - Scope HW 1-5 and lectures until today (Feb 23)

Language Models: A History

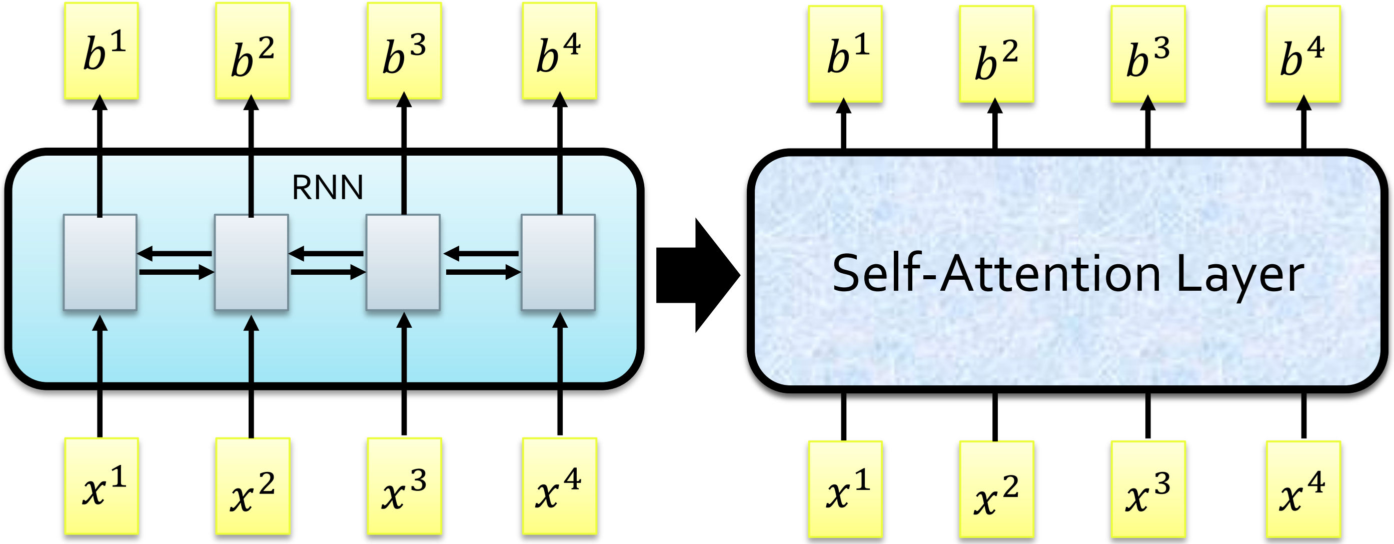
- Probabilistic n-gram models of text generation [Jelinek+ 1980's, …]
  - Applications: Speech Recognition, Machine Translation
- Word representation learning [Brown 1992, …]
  - Brown, LSA, Word2Vec, Glove …
- Statistical or shallow neural LMs (late 90's – mid 00's) [Bengio+ 2001, …]
- Pre-training deep neural language models (2017's onward):
  - Many models based on: Self-Attention

RNNs, Back to the Cons

- While RNNs in theory can represent long sequences, they quickly forget portions of the input.
- Vanishing/exploding gradients
- Difficult to parallelize

Self-Attention

- b^i is obtained based on the whole input sequence.
- can be parallelly computed.



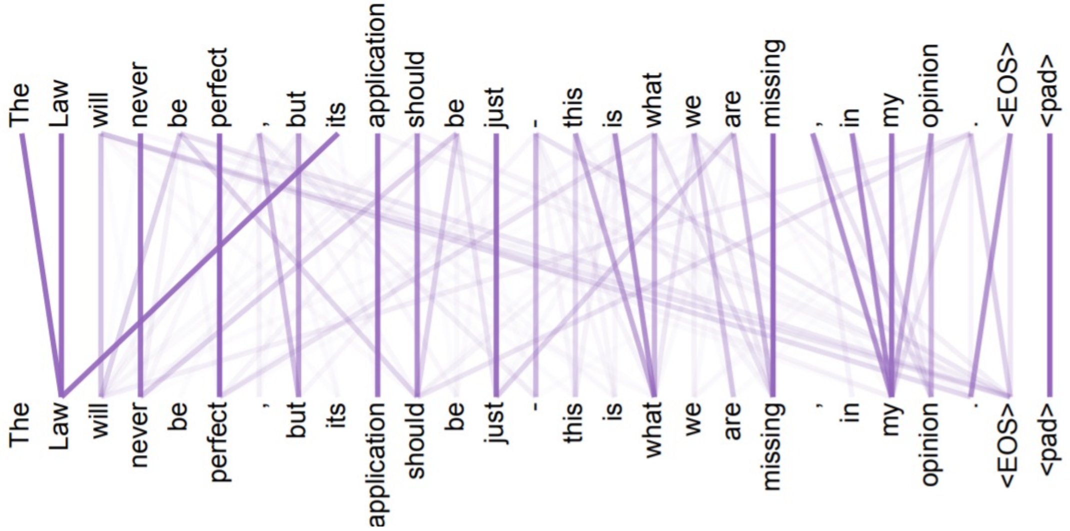
Idea: replace any thing done by RNN with self-attention.

"Neural machine translation by jointly learning to align and translate" Bahdanau etl. 2014;
"Attention is All You Need" Vaswani et al. 2017

[adopted from Hung-yi Lee]

Attention

- <u>Core idea</u>: build a mechanism to focus ("attend") on a particular part of the context.



[Attention Is All You Need, Vaswani et al. 2017]

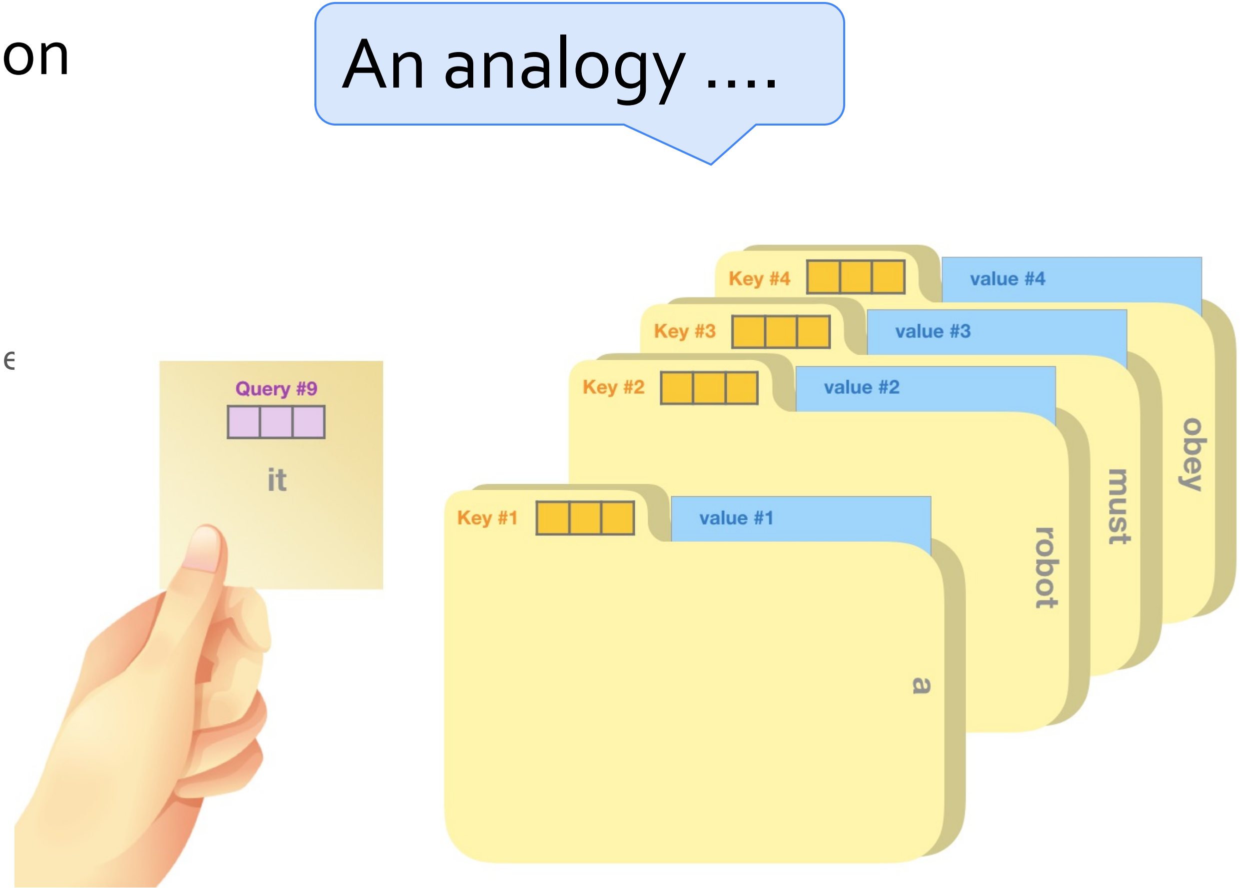
Defining Self-Attention

- Terminology:
  - Query: to match others
  - Key: to be matched
  - Value: information to be extracted

[Vaswani et al. 2017: https://arxiv.org/abs/1706.03762]

Defining Self-Attention

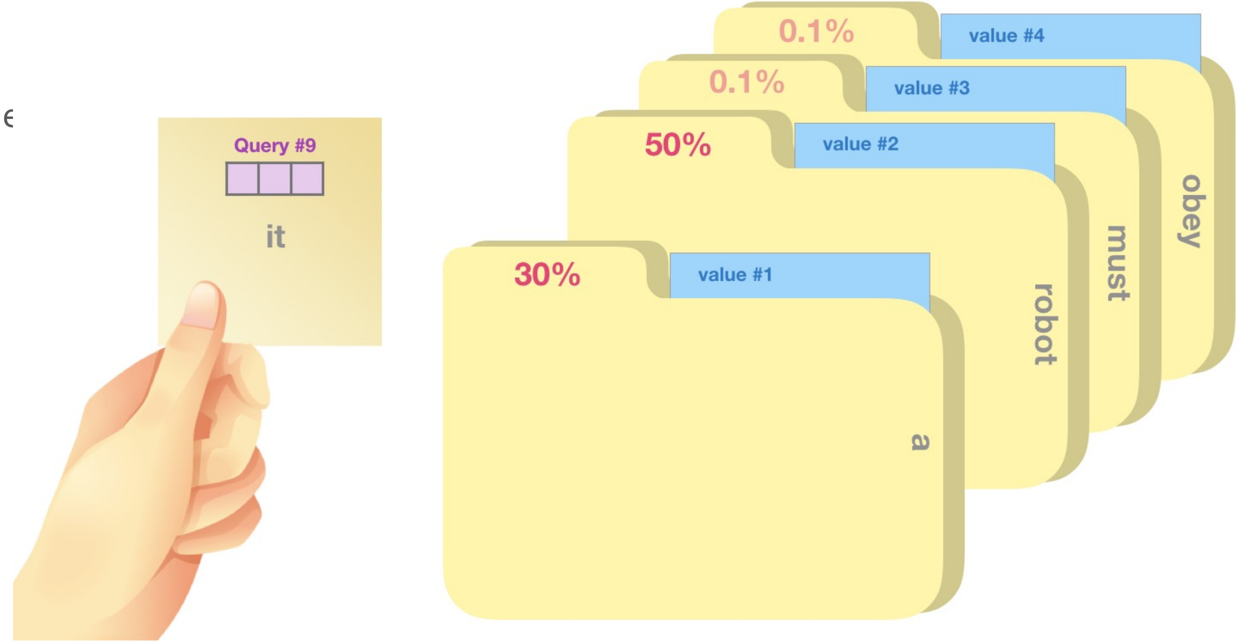
- Terminology:
  - Query: to match others
  - Key: to be matched
  - Value: information to be e



[Vaswani et al. 2017: https://arxiv.org/abs/1706.03762]

Defining Self-Attention

- Terminology:
  - Query: to match others
  - Key: to be matched
  - Value: information to be e



[Vaswani et al. 2017: https://arxiv.org/abs/1706.03762]

q: query (to match others)

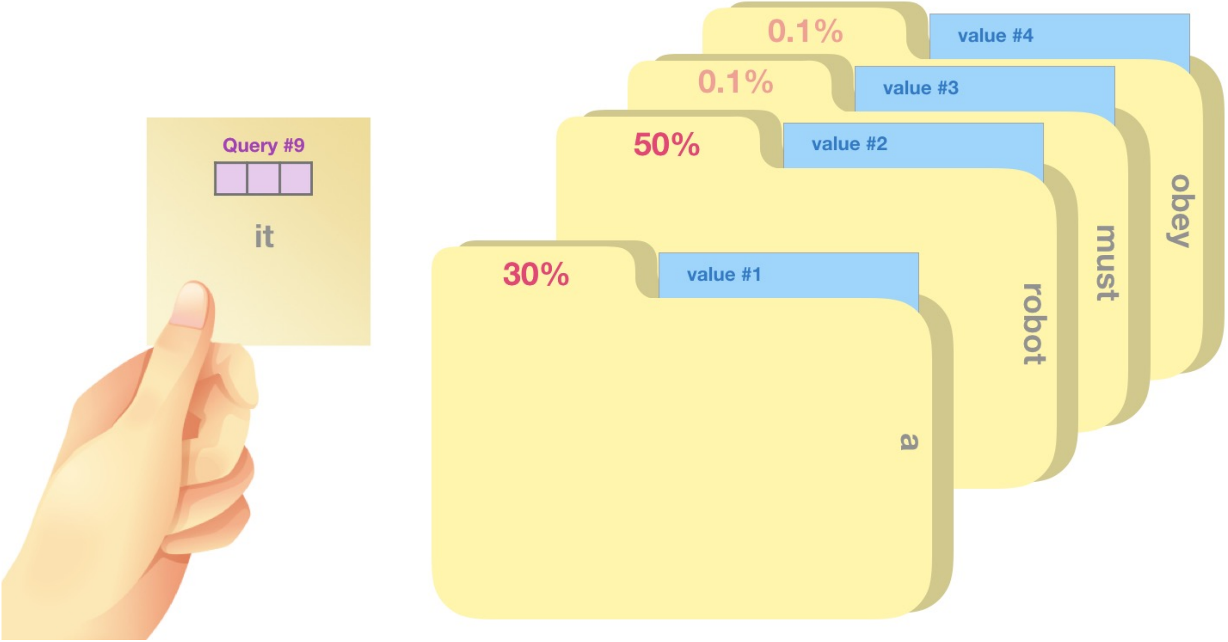
$$q_i = W^q x_i$$

k: key (to be matched)

$$k_i = W^k x_i$$

v: value (information to be extracted)

$$v_i = W^v x_i$$



q: query (to match others)

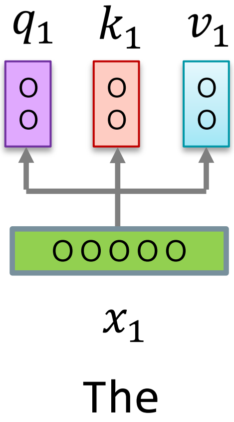
$$q_i = W^q x_i$$

k: key (to be matched)

$$k_i = W^k x_i$$

v: value (information to be extracted)

$$v_i = W^v x_i$$



q: query (to match others)

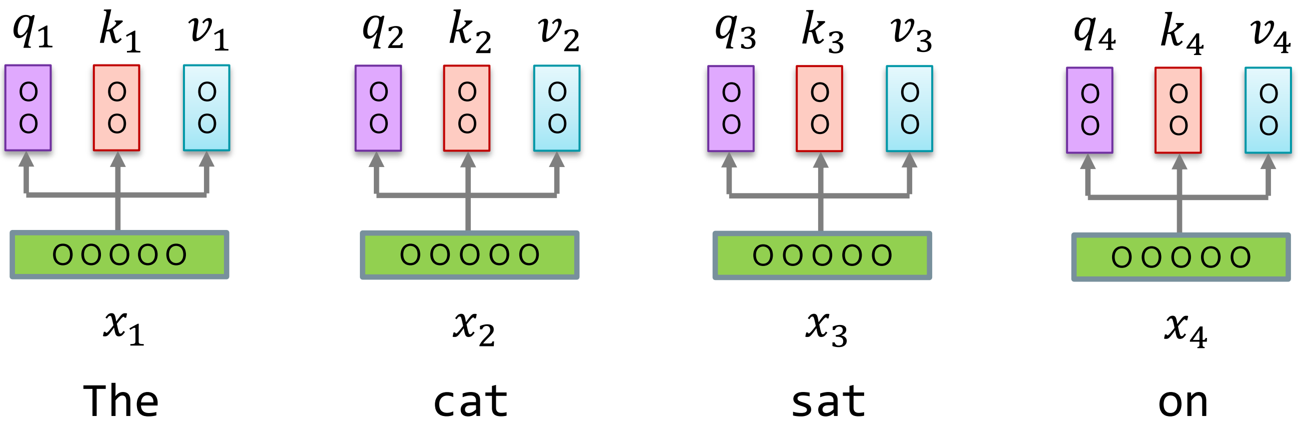
$$q_i = W^q x_i$$

k: key (to be matched)

$$k_i = W^k x_i$$

v: value (information to be extracted)

$$v_i = W^v x_i$$



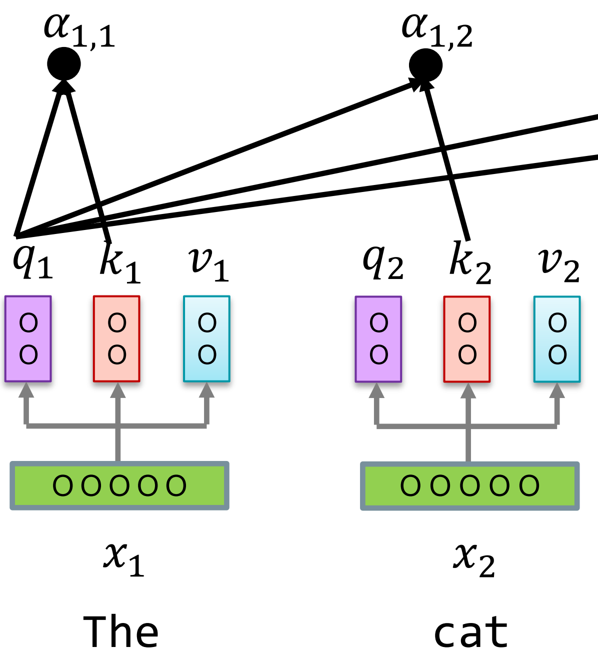
q: query (to match others)

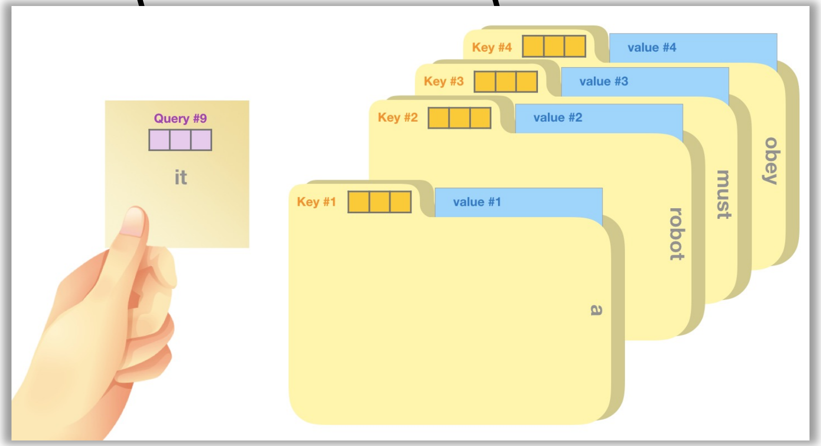
k: key (to be matched)

v: value (information to be extracted)

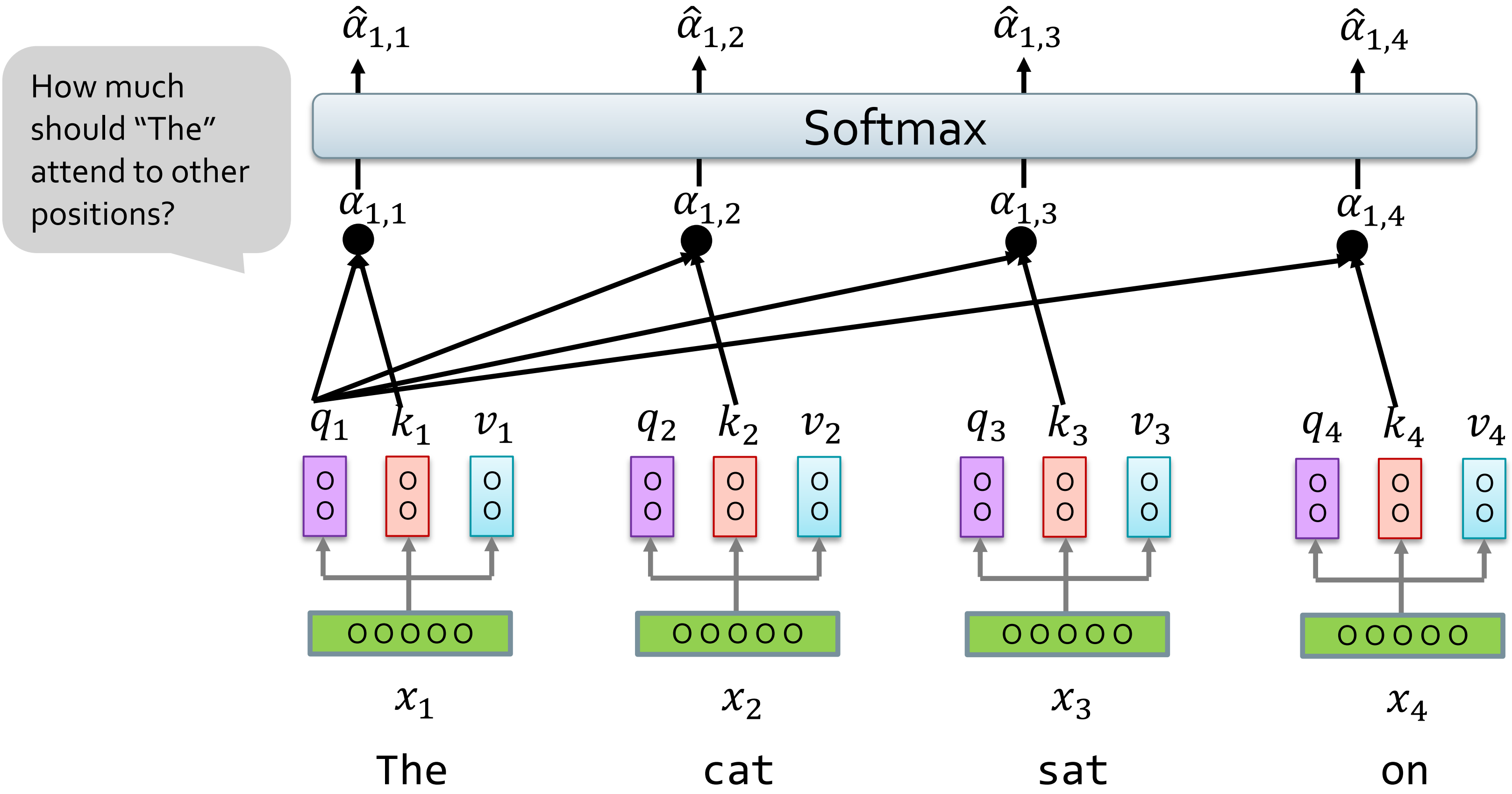
$$\alpha_{1,i} = \underbrace{q^1 \cdot k^i / \alpha}_{\text{Scaled dot product}}$$

How much should "The" attend to other positions?



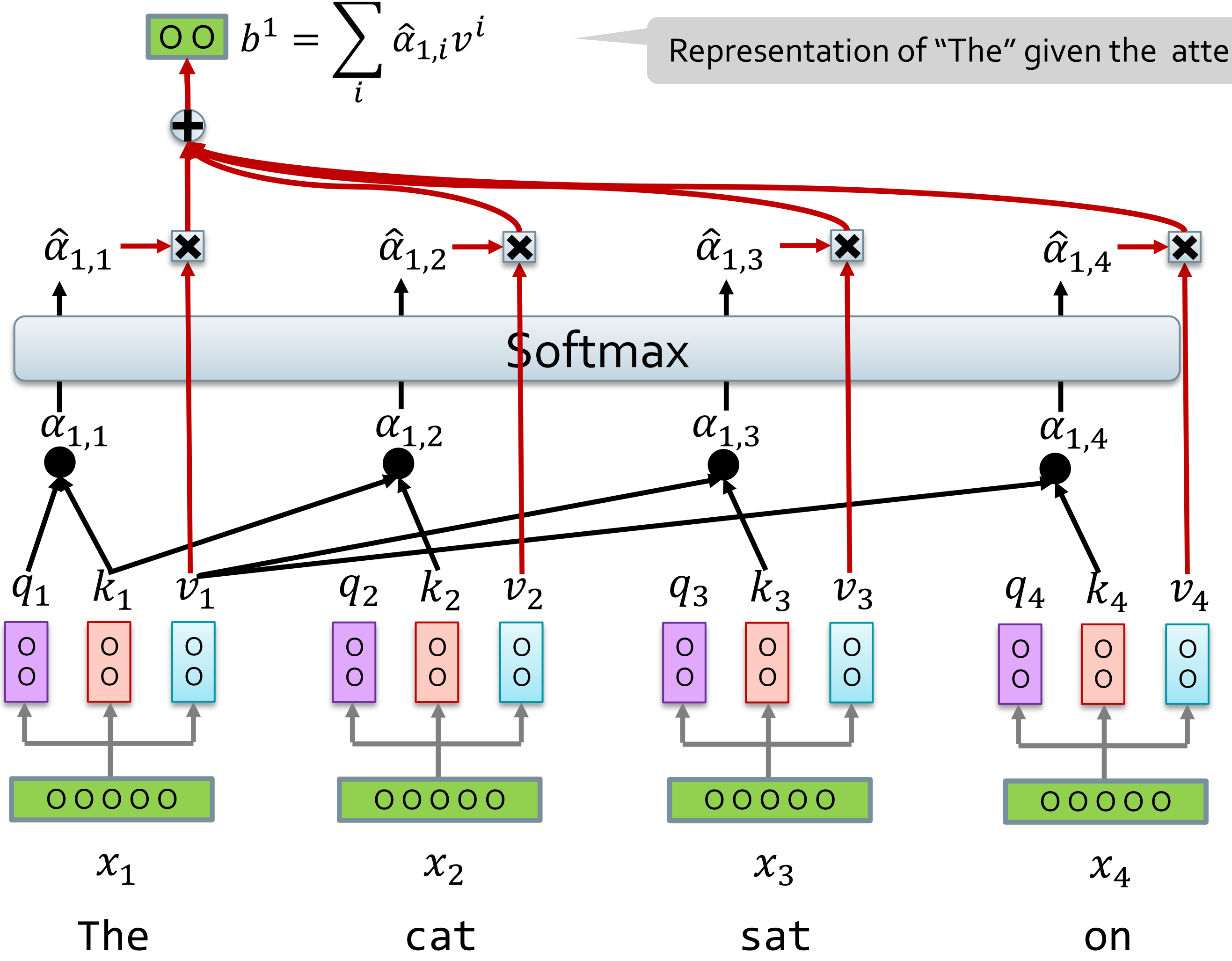


$$\sigma(z)_i = \frac{exp(z_i)}{\sum_j exp(z_j)}$$



$$b^1 = \sum_i \hat{\alpha}_{1,i} v^i$$

Representation of "The" given the attention weights



Self-Attention

- Can write it in matrix form:
- Given input $\mathbf{x}$:

$$Q = \mathbf{W}^q \mathbf{x}$$
$$K = \mathbf{W}^k \mathbf{x}$$
$$V = \mathbf{W}^v \mathbf{x}$$

$$\text{Attention}(\mathbf{x}) = \text{softmax}\left(\frac{QK^{\mathrm{T}}}{\alpha}\right)V$$

hardmaru
@hardmaru

The most important formula in deep learning after 2018

Self-Attention

What is self-attention? Self-attention calculates a weighted average of feature representations with the weight proportional to a similarity score between pairs of representations. Formally, an input sequence of n tokens of dimensions d, $X \in \mathbf{R}^{n\times d}$, is projected using three matrices $W_Q \in \mathbf{R}^{d\times d_q}$, $W_K \in \mathbf{R}^{d\times d_k}$, and $W_V \in \mathbf{R}^{d\times d_v}$ to extract feature representations Q, K, and V, referred to as query, key, and value respectively with $d_k = d_q$. The outputs Q, K, V are computed as

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V. \tag{1}$$

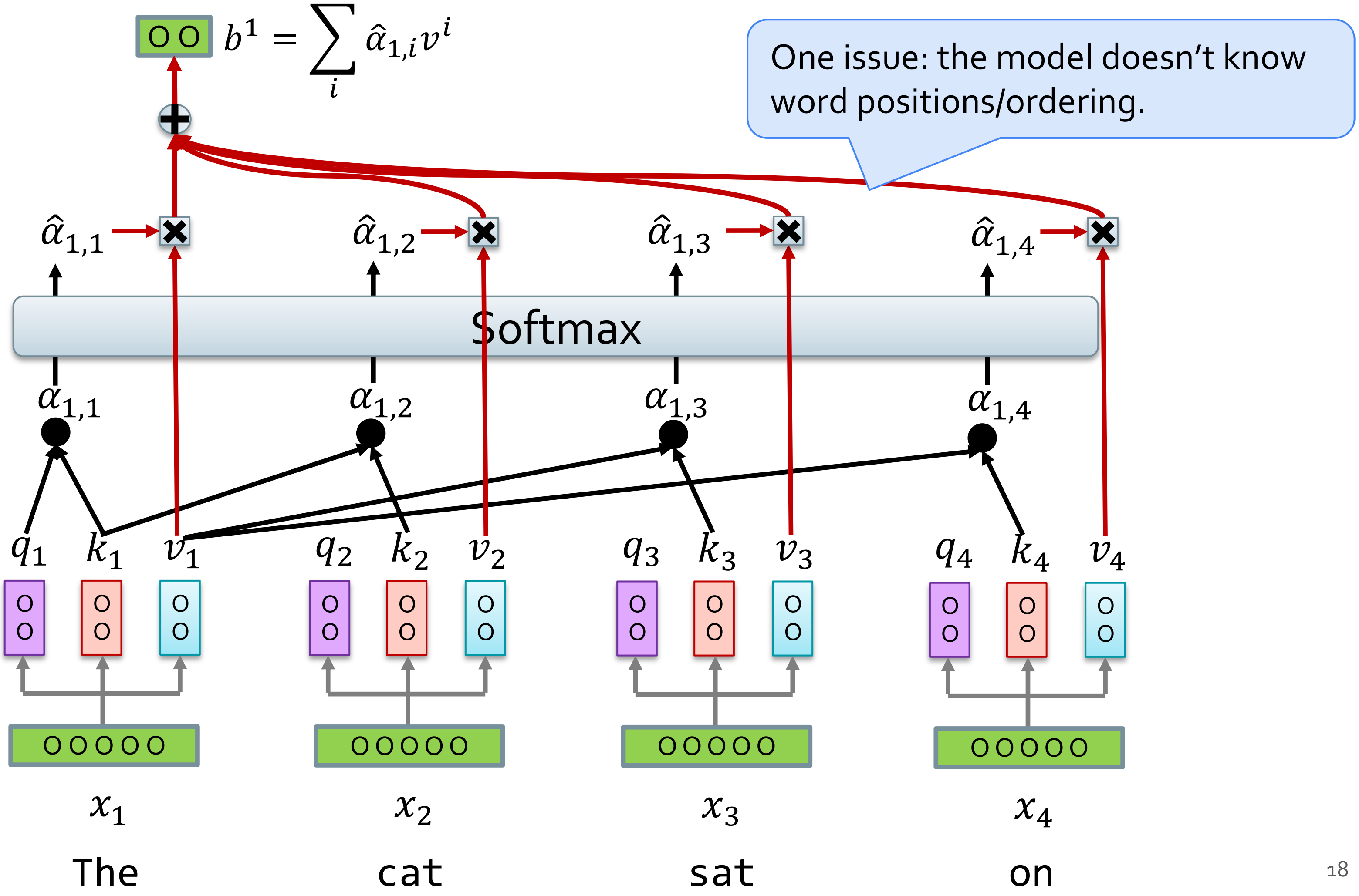
So, self-attention can be written as,

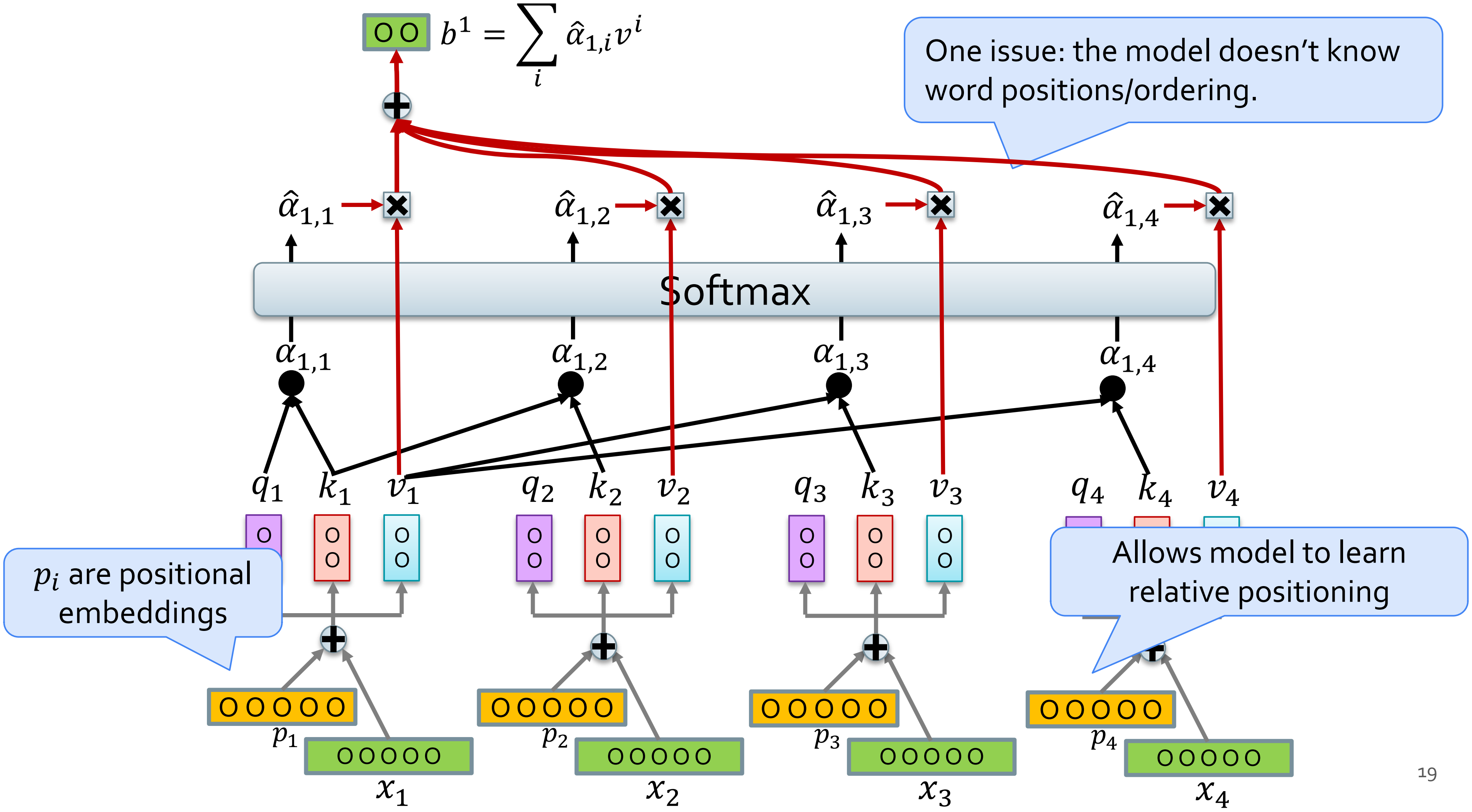
$$S = D(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_q}}\right)V, \tag{2}$$

where softmax denotes a *row-wise* softmax normalization function. Thus, each element in S depends on all other elements in the same row.

9:08 PM · Feb 9, 2021 · Twitter Web App

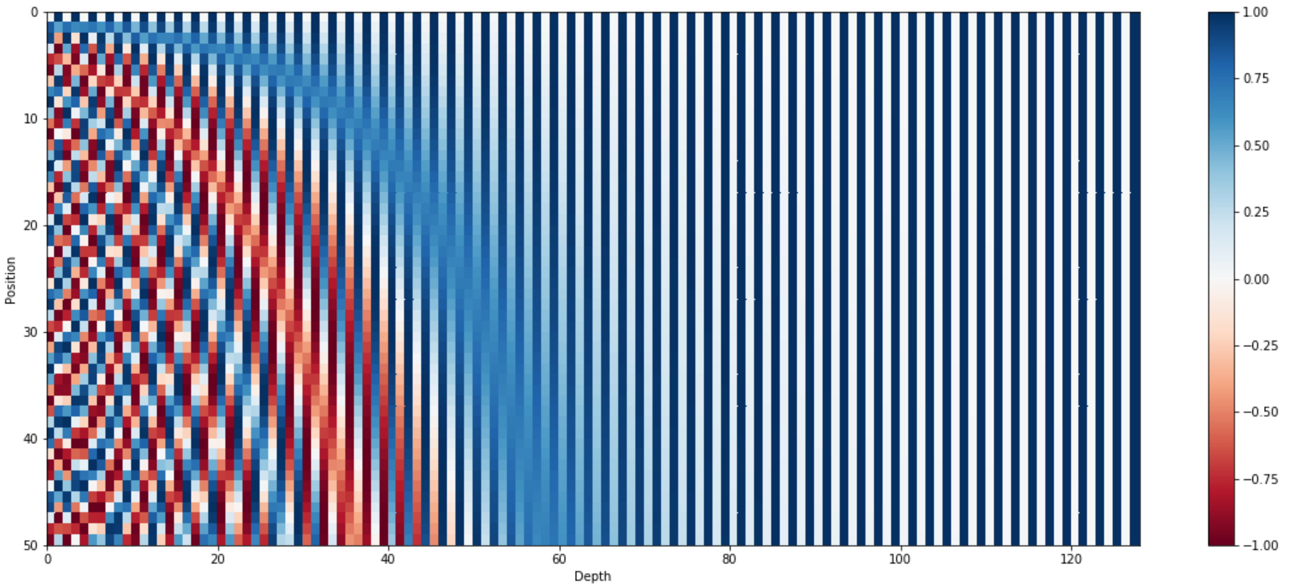
553 Retweets **42** Quote Tweets **3,338** Likes

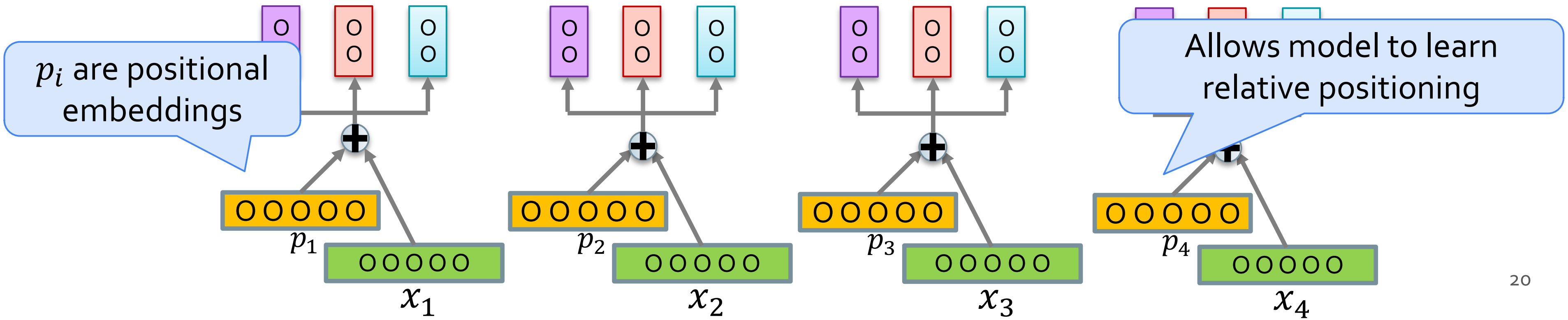

One issue: the model doesn't know word positions/ordering.
$b^1 = \sum_i \hat{\alpha}_{1,i} v^i$
Softmax
$\hat{\alpha}_{1,1}$ $\hat{\alpha}_{1,2}$ $\hat{\alpha}_{1,3}$ $\hat{\alpha}_{1,4}$
$\alpha_{1,1}$ $\alpha_{1,2}$ $\alpha_{1,3}$ $\alpha_{1,4}$
q_1 k_1 v_1 q_2 k_2 v_2 q_3 k_3 v_3 q_4 k_4 v_4
x_1 x_2 x_3 x_4
The cat sat on



$b^1 = \sum_i \hat{\alpha}_{1,i} v^i$
One issue: the model doesn't know word positions/ordering.
$\hat{\alpha}_{1,1}$ $\hat{\alpha}_{1,2}$ $\hat{\alpha}_{1,3}$ $\hat{\alpha}_{1,4}$
Softmax
$\alpha_{1,1}$ $\alpha_{1,2}$ $\alpha_{1,3}$ $\alpha_{1,4}$
q_1 k_1 v_1 q_2 k_2 v_2 q_3 k_3 v_3 q_4 k_4 v_4
p_i are positional embeddings
Allows model to learn relative positioning
p_1 p_2 p_3 p_4
x_1 x_2 x_3 x_4

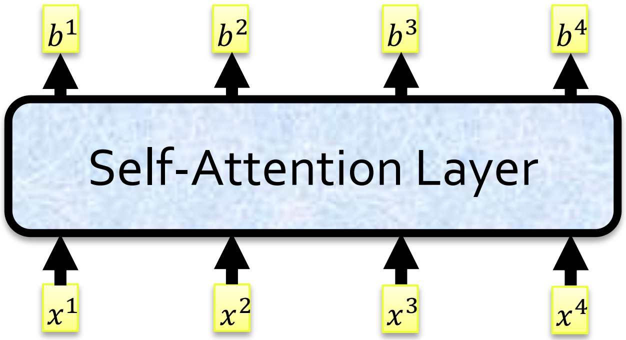
An approach:
Sine/Cosine encoding





Self-Attention: Back to Big Picture

- Attention is a powerful mechanism to create context-aware representations
- A way to focus on select parts of the input



- Better at maintaining long-distance dependencies in the context.

[Attention Is All You Need, Vaswani et al. 2017]

Properties of Self-Attention

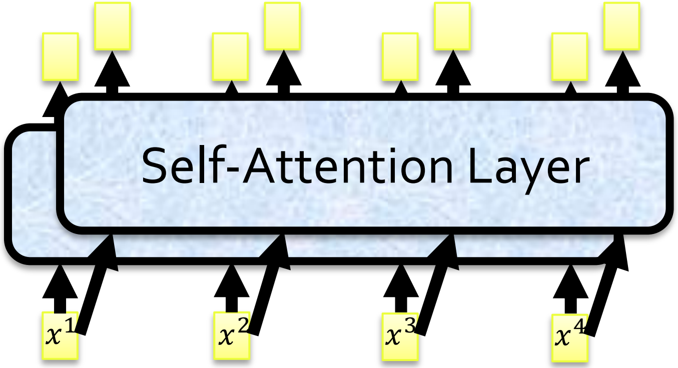
Layer Type	Complexity per Layer	Sequential Operations
Self-Attention	$O(n^2 \cdot d)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$

- n = sequence length, d = hidden dimension
- Quadratic complexity, but:
  - O(1) sequential operations (not linear like in RNN)
- Efficient implementations

[Attention Is All You Need, Vaswani et al. 2017]

Multi-Headed Self-Attention

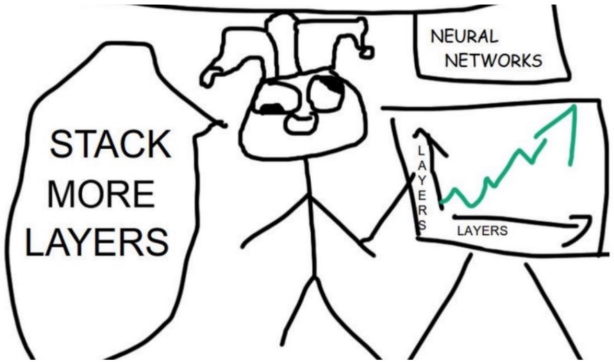
- Multiple parallel attention layers is quite common.
    - Each attention layer has its own parameters.
    - Concatenate the results and run them through a linear projection.

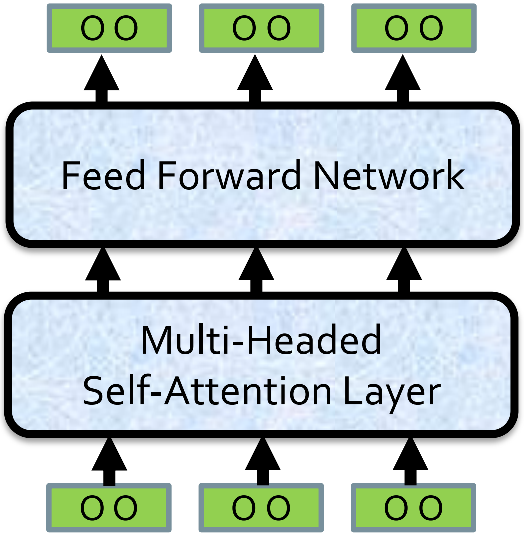


[Attention Is All You Need, Vaswani et al. 2017]

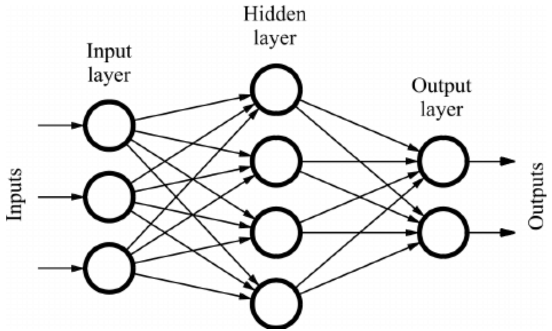
How Do We Make it Deep?

- Add a feed-forward network on top it to add more capacity/expressivity.
- Repeat!





Feedforward Net: Refresher



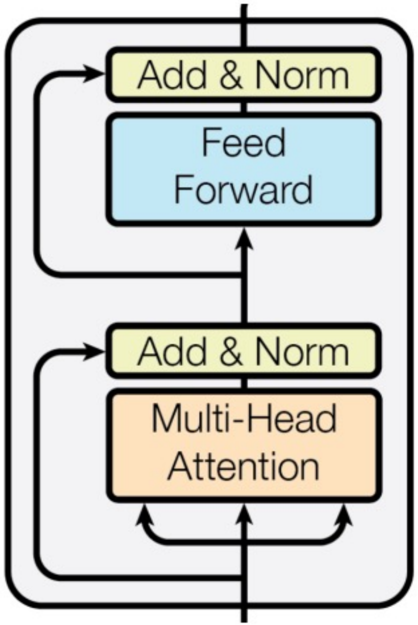
A fully-connected network of nodes and weights.

[Attention Is All You Need, Vaswani et al. 2017]

How Do We Prevent Vanishing Gradients?

- Residual connections let the model “skip” layers
  - These connections are particularly useful for training deep networks

- Use layer normalization to stabilize the network and allow for proper gradient flow



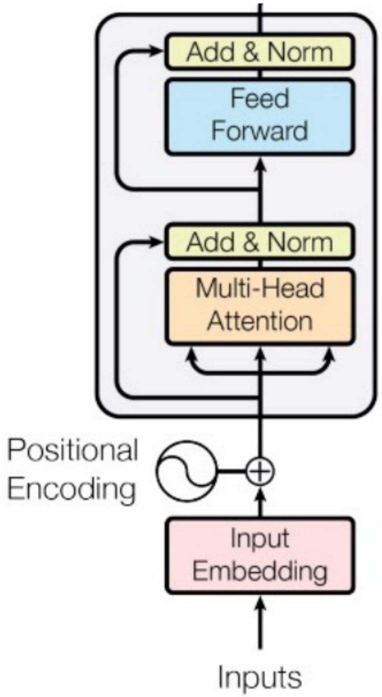
[Attention Is All You Need, Vaswani et al. 2017]

Putting It Together

Given input $\mathbf{x}$:

$$Q = \mathbf{W}^q \mathbf{x}$$
$$K = \mathbf{W}^k \mathbf{x}$$
$$V = \mathbf{W}^v \mathbf{x}$$

$$\text{Attention}(\mathbf{x}) = \text{softmax}\left(\frac{QK^{\mathrm{T}}}{\alpha}\right)V$$



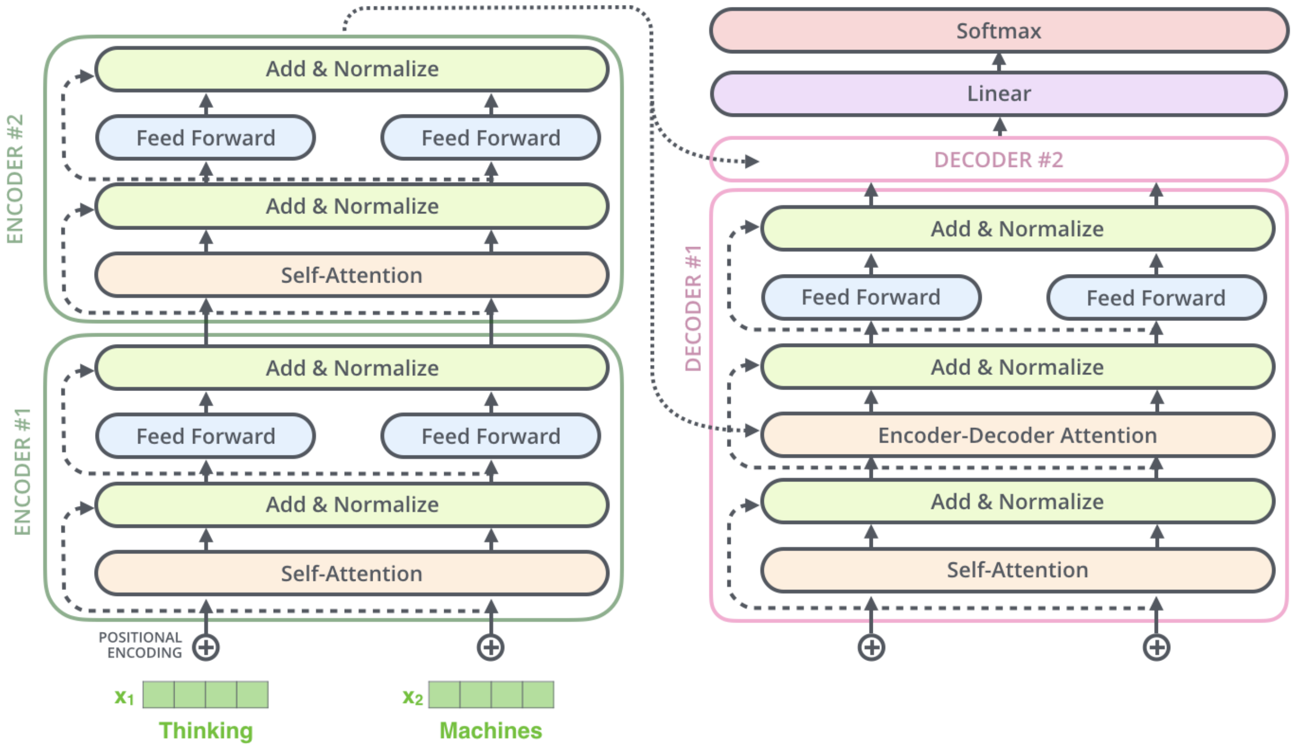
[Attention Is All You Need, Vaswani et al. 2017]

Transformer

[Attention Is All You Need, Vaswani et al. 2017]

Transformer [Vaswani et al. 2017]

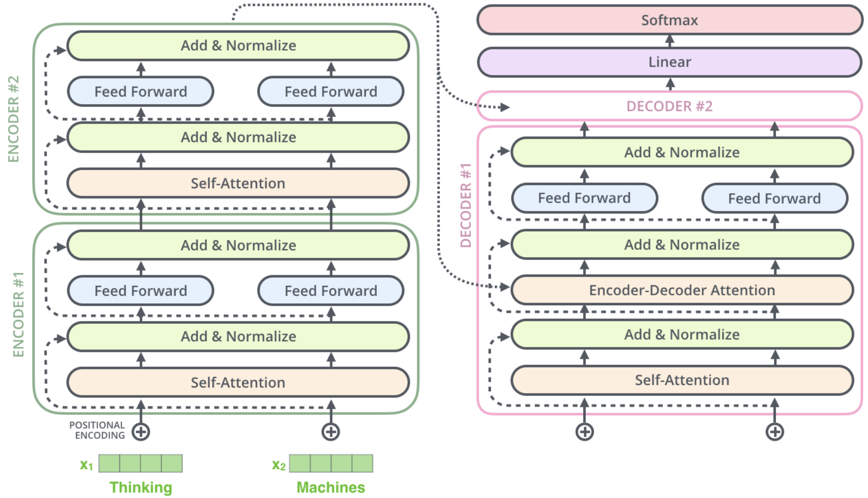
- An encoder-decoder architecture built with attention modules.



Transformer [Vaswani et al. 2017]

- Computation of **encoder** attends to both sides.

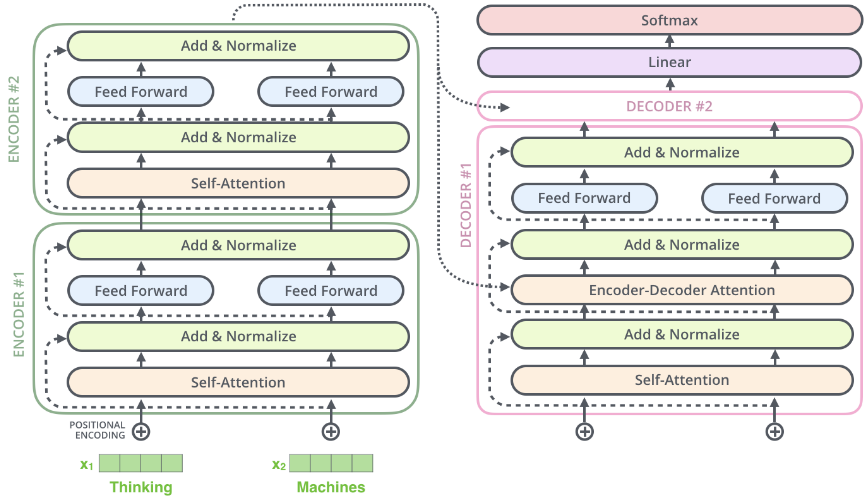
Encoder Self-Attention



[Attention Is All You Need, Vaswani et al. 2017]

Transformer [Vaswani et al. 2017]

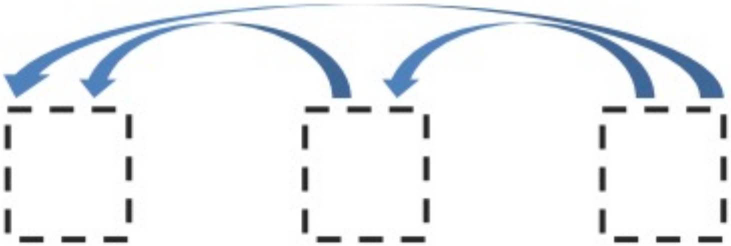
- At any step of **decoder**, it attends to previous computation of **encoder**

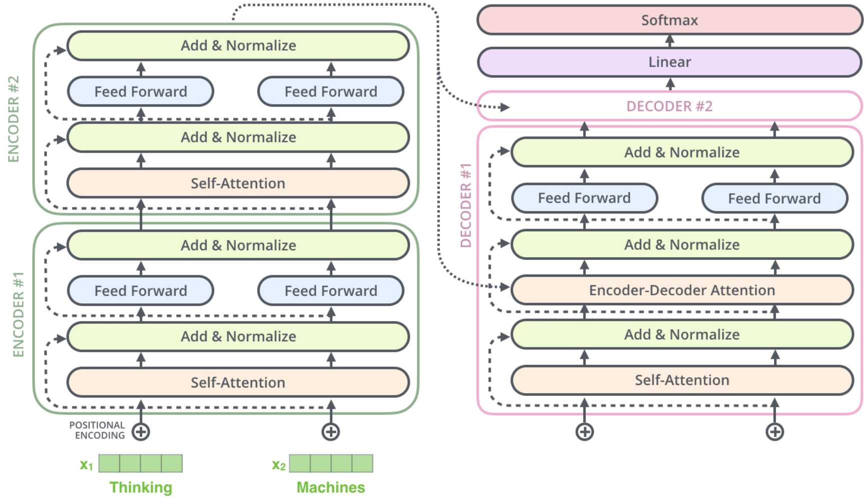


[Attention Is All You Need, Vaswani et al. 2017]

Transformer [Vaswani et al. 2017]

- At any step of **decoder**, it attends to previous computation of **encoder** as well as **decoder's** own generations



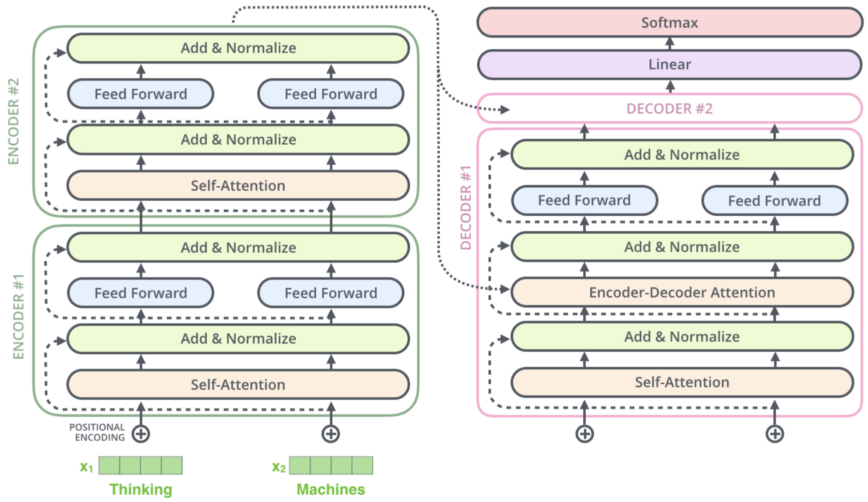


[Attention Is All You Need, Vaswani et al. 2017]

Transformer [Vaswani et al. 2017]

- At any step of **decoder**, it attends to previous computation of **encoder** as well as **decoder's** own generations

- At any step of decoder, **re-use** previous computation of encoder**.**
- Computation of decoder is **linear**, instead of quadratic.

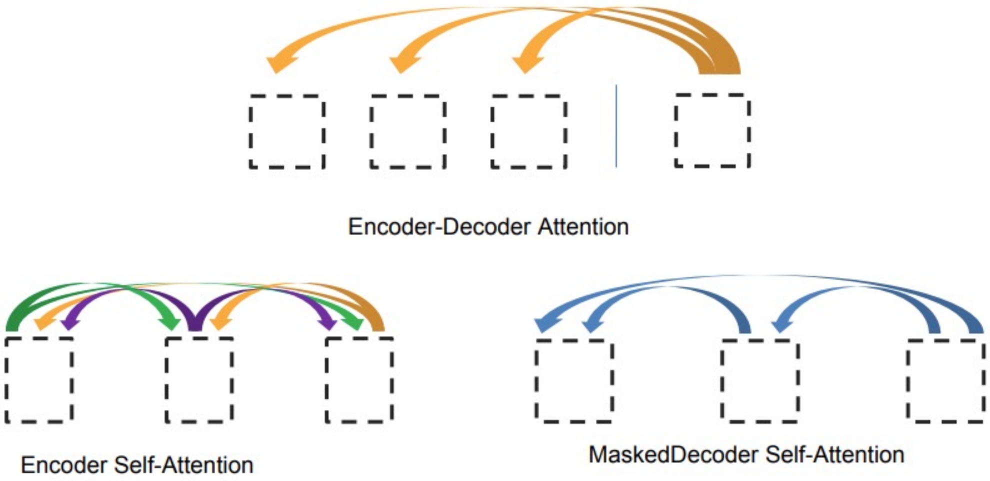


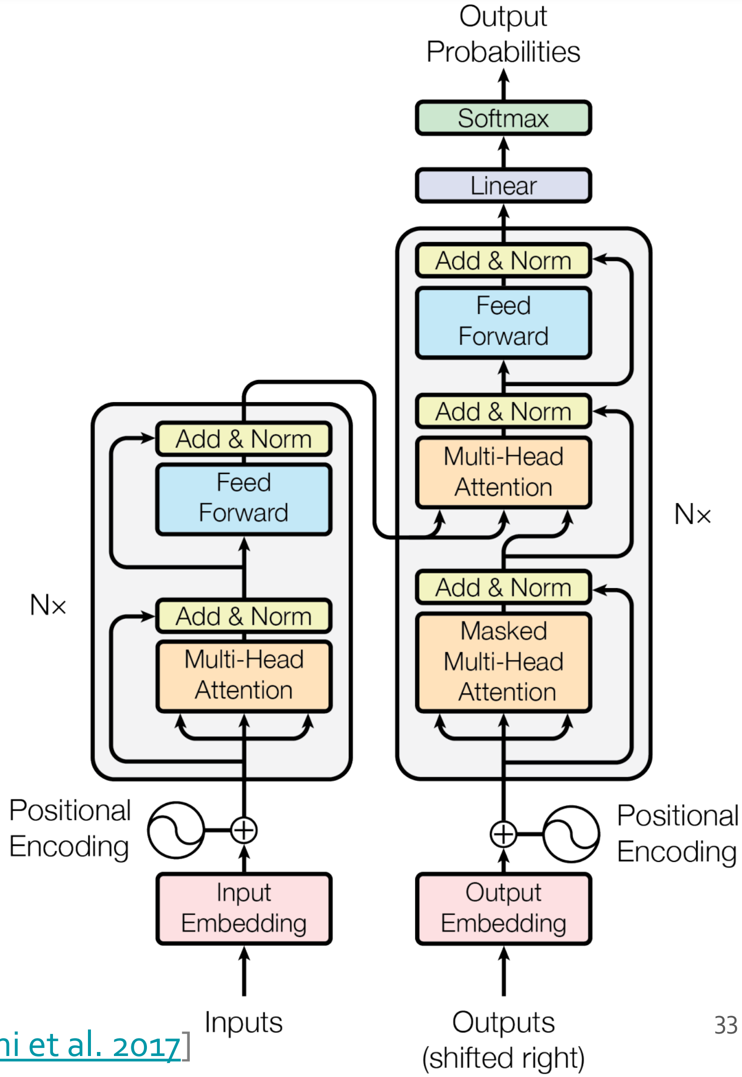
[Attention Is All You Need, Vaswani et al. 2017]

Transformer [Vaswani et al. 2017]

- An encoder-decoder architecture

- 3 forms of attention





[Attention Is All You Need, Vaswani et al. 2017]

Impact of Transformers

- Let to better predictive models of language!

Model	Layers	Heads	Perplexity
LSTMs (Grave et al., 2016)	-	-	40.8
QRNNs (Merity et al., 2018)	-	-	33.0
Transformer	16	16	19.8

[“Efficient Content-Based Sparse Attention with Routing Transformers” Roy et al. 2020]

Wrapping it up

- Yaaay we know Transformers now! 🥳
- Midterm will be up to here!
- Next: extensions on Transformers.