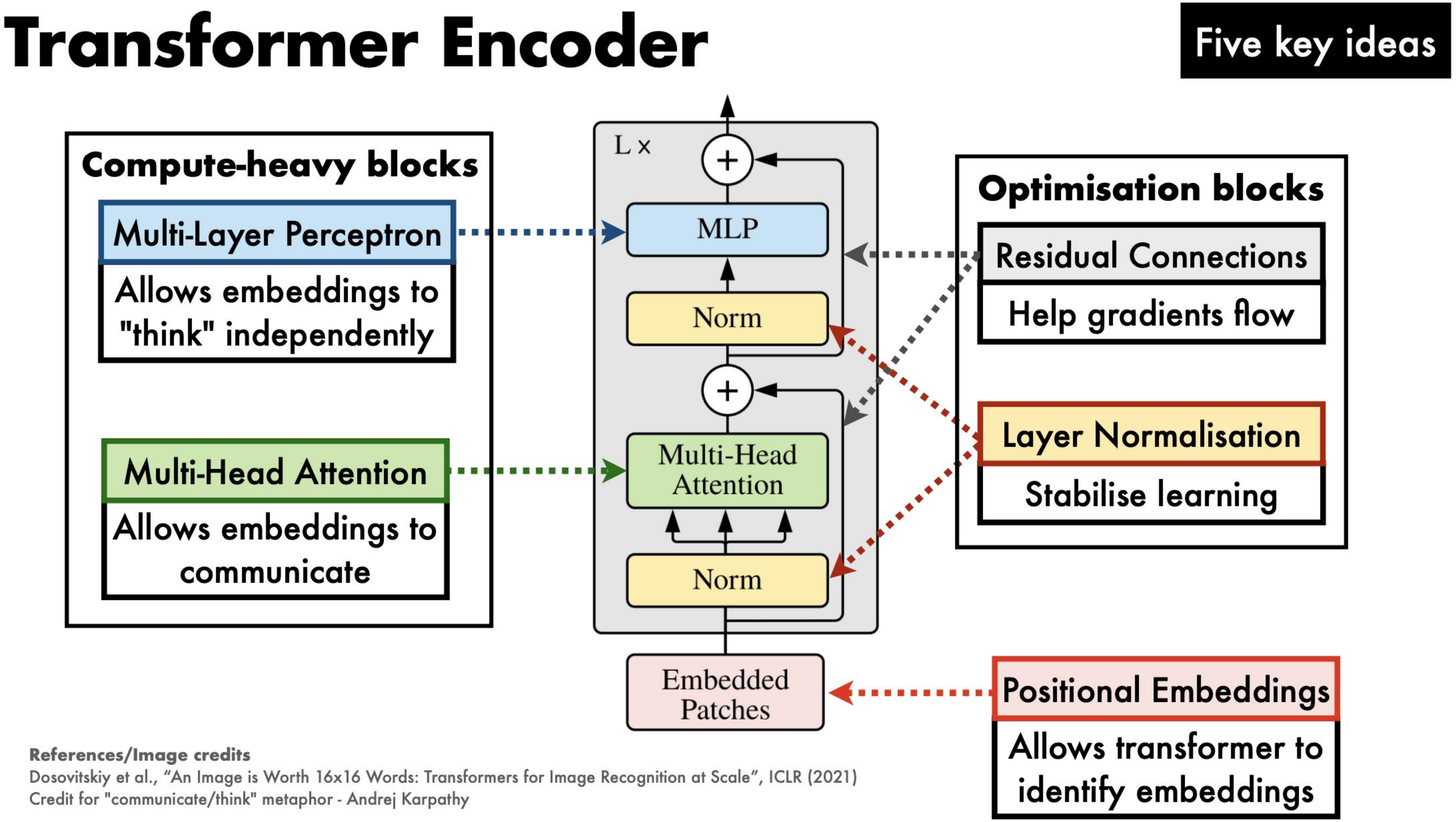
Transformers

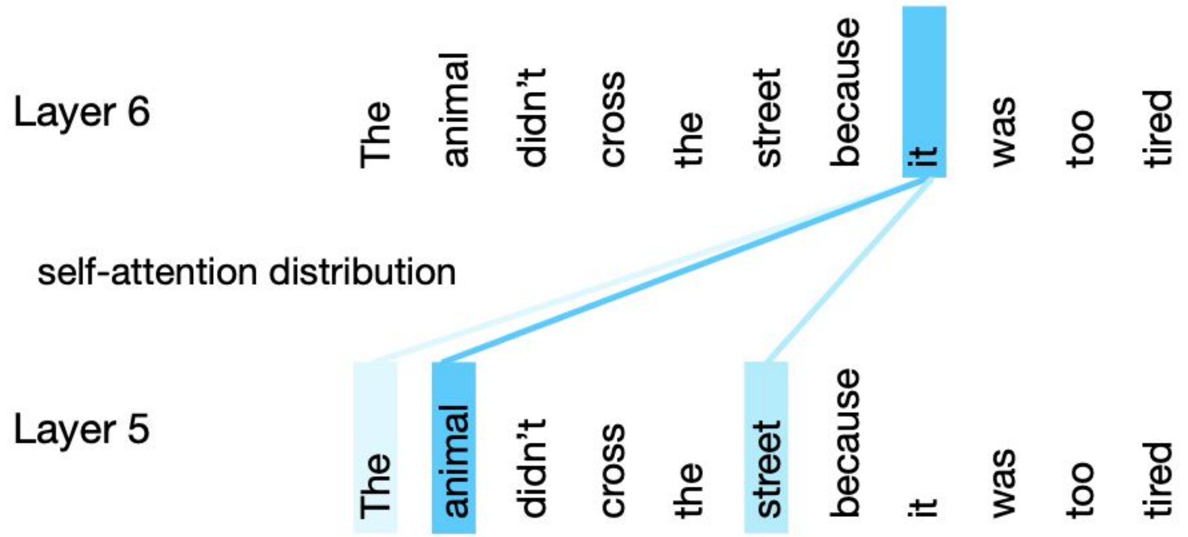
Transformers



References/Image credits
Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR (2021)
Credit for "communicate/think" metaphor - Andrej Karpathy

Blank Slide

Self Attention

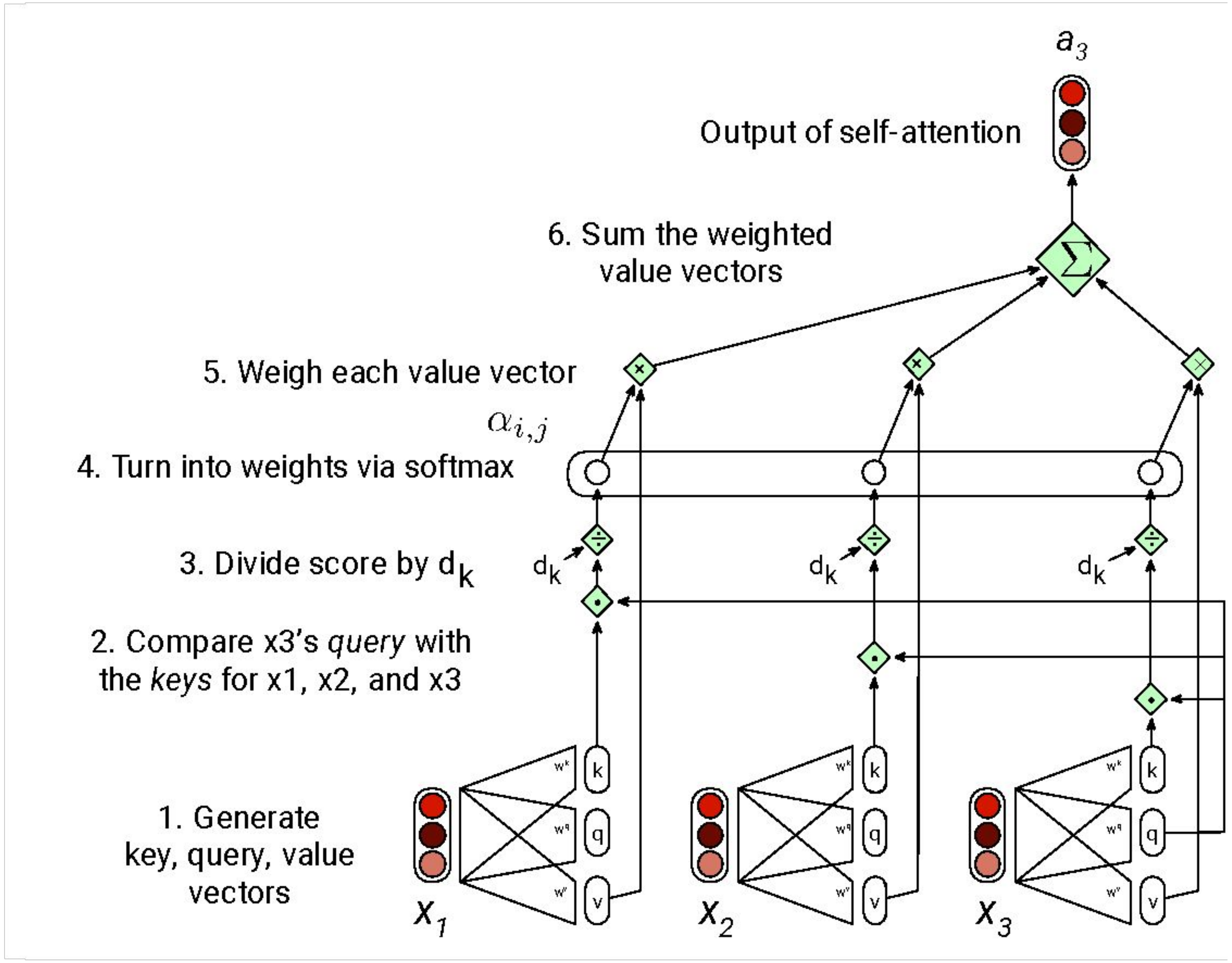


Query, Key and Value

- As *the current focus of attention* when being compared to all of the other preceding inputs. We'll refer to this role as a **query**.
- In its role as *a preceding input* being compared to the current focus of attention. We'll refer to this role as a **key**.
- And finally, as a **value** used to compute the output for the current focus of attention.

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

Example Computation



Computation

$$q_i = x_i W^Q;\ k_i = x_i W^K;\ v_i = x_i W^V$$

$$\mathrm{score}(x_i, x_j) = \frac{q_i \cdot k_j}{\sqrt{d_k}}$$

$$a_{ij} = \mathrm{softmax}(\mathrm{score}(x_i, x_j)) \quad \forall j \leq i$$

$$a_i = \sum_{j \leq i} a_{ij} v_j$$

Blank Slide

Parallelization of Computation

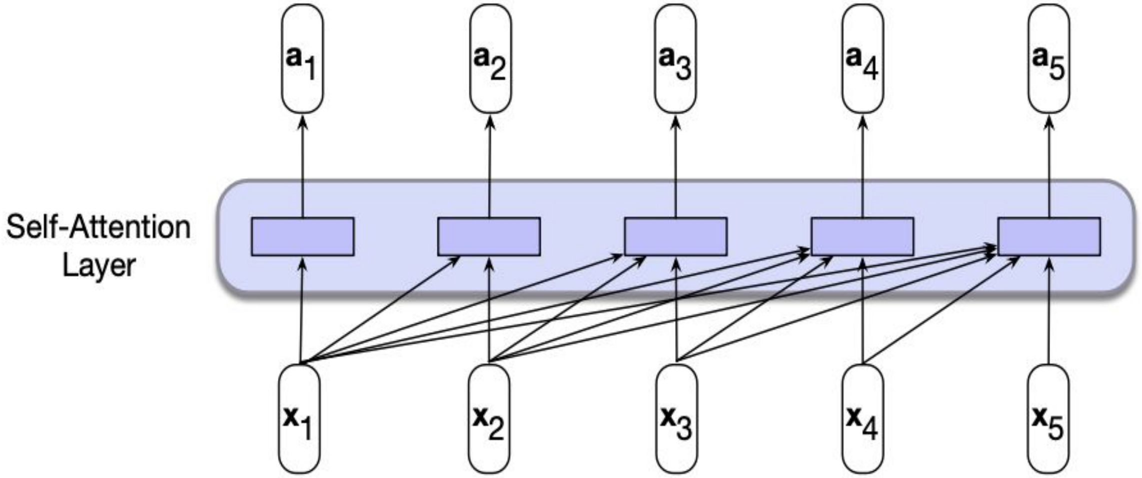
$$\mathbf{Q} = \mathbf{X}\mathbf{W}^{\mathbf{Q}}; \;\; \mathbf{K} = \mathbf{X}\mathbf{W}^{\mathbf{K}}; \;\; \mathbf{V} = \mathbf{X}\mathbf{W}^{\mathbf{V}}$$

$$\mathbf{A} = \mathrm{SelfAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \;=\; \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}}\right)\mathbf{V}$$

Blank Slide

Self Attention: Casual/Masked

Masking out the future



QK^T

	q1•k1	−∞	−∞	−∞	−∞
	q2•k1	q2•k2	−∞	−∞	−∞
N	q3•k1	q3•k2	q3•k3	−∞	−∞
	q4•k1	q4•k2	q4•k3	q4•k4	−∞
	q5•k1	q5•k2	q5•k3	q5•k4	q5•k5
			N		

Unlike RNNs, computation at each step are independent of all other time steps and can be done parallelly.

What is the typical value of N?
4096 tokens

Masked Attention

Don’t let vectors “look ahead” in the sequence

Inputs:

Input vectors: X (Shape: $N_X \times D_Q$)

Key matrix: $W_K(Shape\colon D_X \times D_Q)$

Value matrix: $W_V(Shape\colon\ D_X{\times}D_V)$

Query matrix: $\mathbf{W}_Q$ (Shape: $\mathrm{D}_Q \times D_Q$)

Computation:

Query Vectors $Q = XW_!$

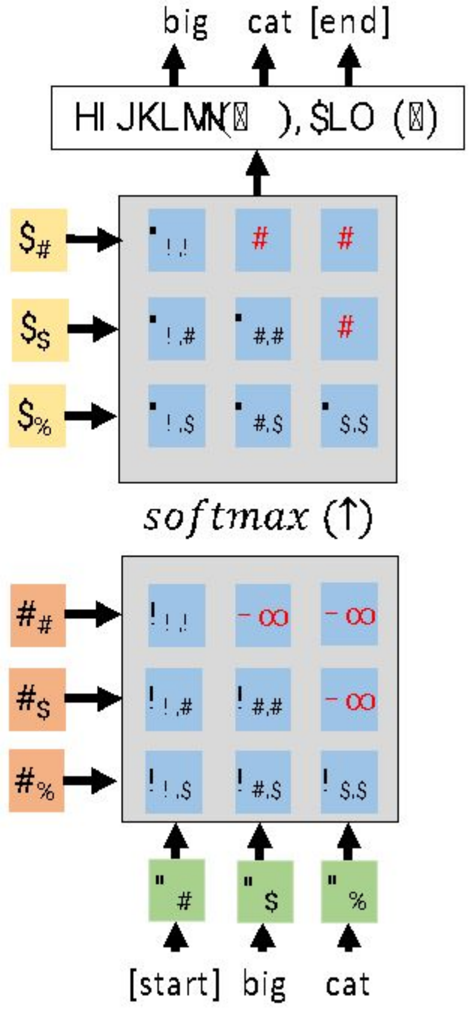
Key vectors: $K\ =\ XW_)\ \ (Shape\colon\ N_" \ \times D_!\ )$

Value Vectors: $V\ =\ XW_*\ (Shape\colon\ N_"\ \times D_*)$

Similarities: $E\ = \frac{!\ "\ !}{\sqrt{\$}_"}\ (Shape\colon\ N_" \ \times N_"\ ) E_{%'}\ =\ (Q_{%}\cdot K\ )/\sqrt{D_(}$

Attention weights: $A\ =\ softmax(E, \dim = 1)\ (Shape\colon\ N_" \ \times N_"\ )$

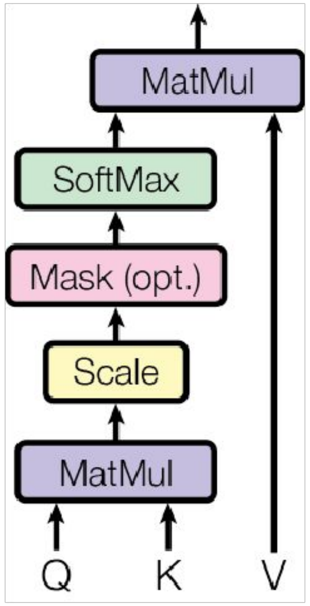
Output vectors: $Y\ =\ AV\ (Shape\colon\ N_"\ \times D_*) Y_{%} = \ \Sigma\ A_{%'}\ V$



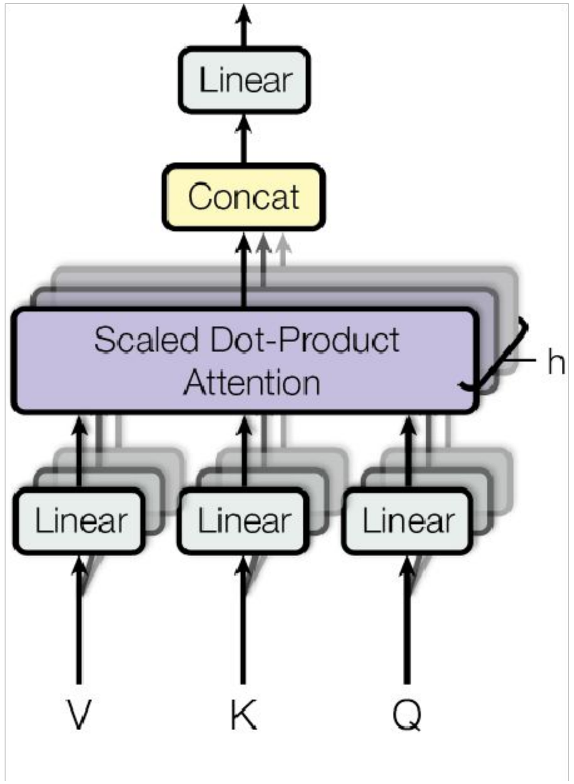
Multi Head Attention: Why?

- Why are we limited to one query, key and value?
- Multiple Feature Maps (like Convolution Layers)

Scaled Dot-Product Attention



Multi-Head Attention



Masked Multi-Head Attention

- Masked multi-head attention: multi-head where some values are masked (i.e., probabilities of masked values are nullified to prevent them from being selected).
- When decoding, an output value should only depend on previous outputs (not future outputs). Hence we mask future outputs.

$$attention(Q,K,V) = softmax\left(\frac{Q^T K}{\sqrt{d_k}}\right)V$$

$$maskedAttention(Q,K,V) = softmax\left(\frac{Q^T K + M}{\sqrt{d_k}}\right)V$$

where M is a mask matrix of 0's and $-\infty$'s

Multi-Head Attention

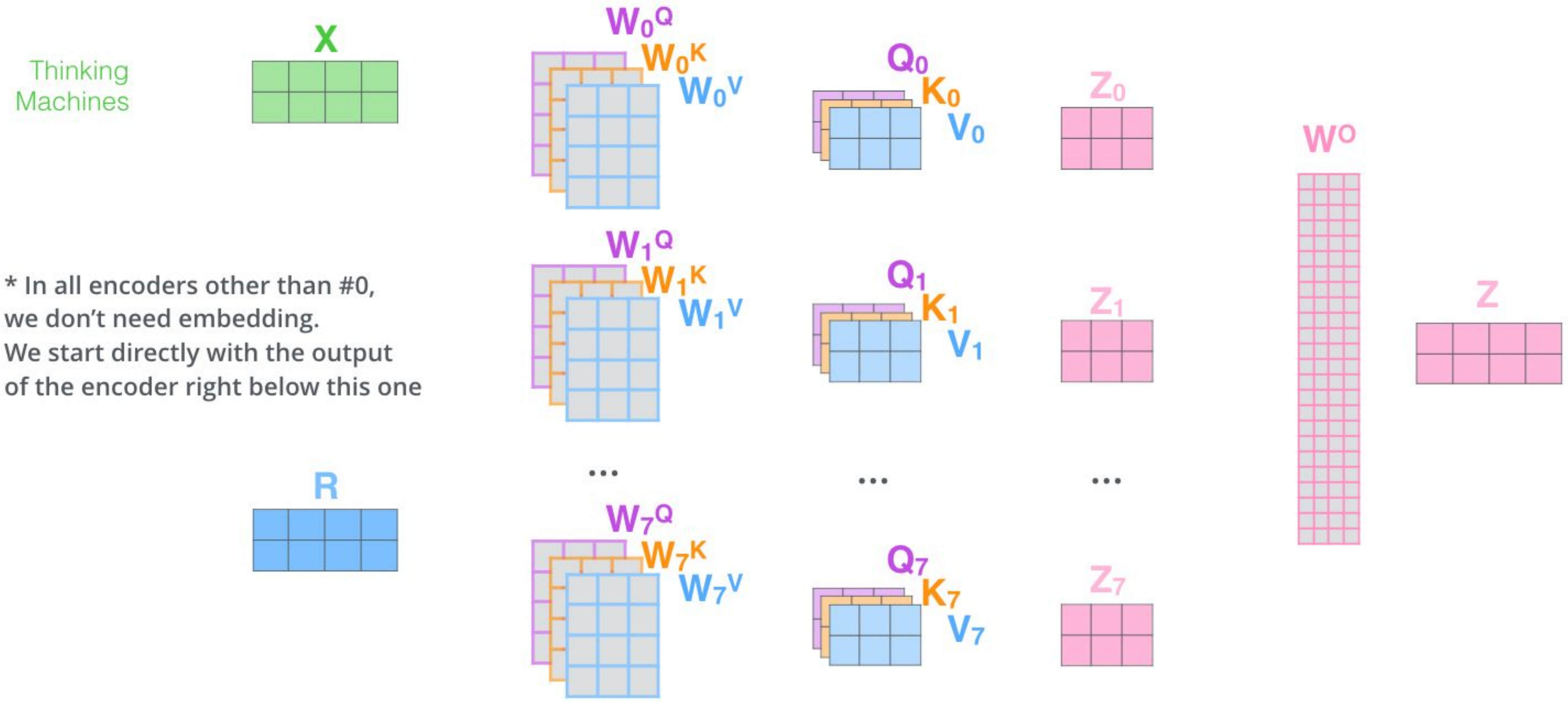
1) This is our input sentence*

2) We embed each word*

3) Split into 8 heads. We multiply X or R with weight matrices

4) Calculate attention using the resulting Q/K/V matrices

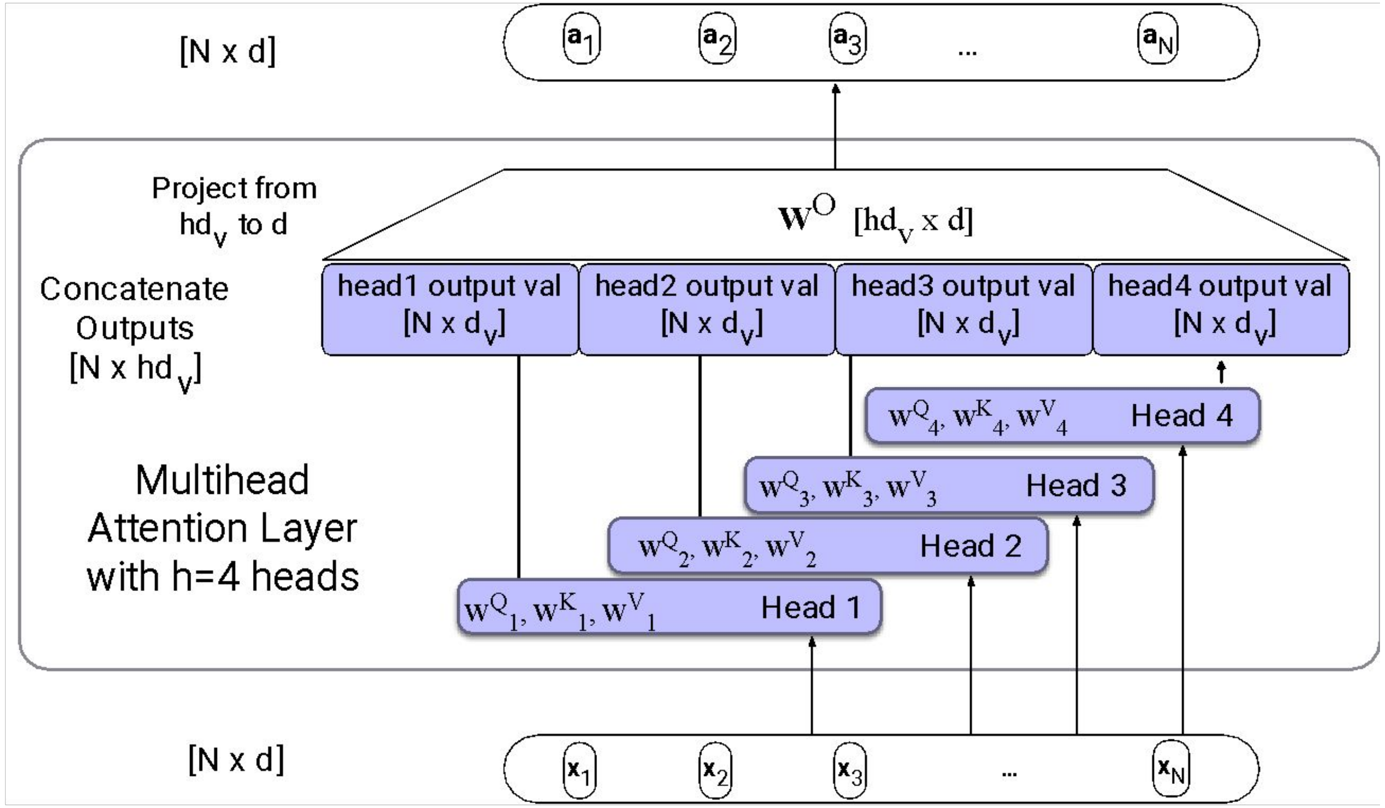
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer



* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one

Blank Slide

Multi Head Attention

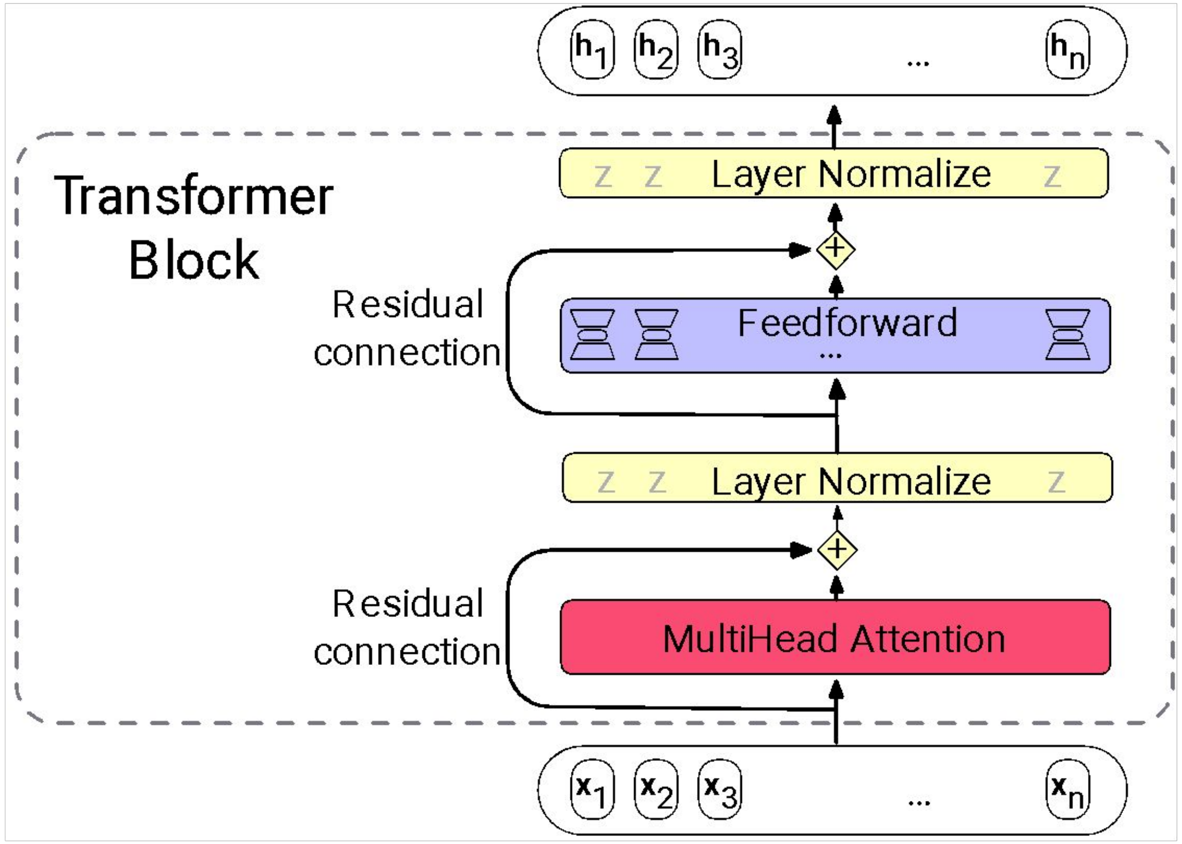


$$\mathbf{Q} = \mathbf{X}\mathbf{W}_i^Q \; ; \; \mathbf{K} = \mathbf{X}\mathbf{W}_i^K \; ; \; \mathbf{V} = \mathbf{X}\mathbf{W}_i^V$$

$$\mathbf{head}_i = \mathrm{SelfAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$$

$$\mathbf{A} = \mathrm{MultiHeadAttention}(\mathbf{X}) = (\mathbf{head}_1 \oplus \mathbf{head}_2 \ldots \oplus \mathbf{head}_h)\mathbf{W}^O$$

Transformer Block



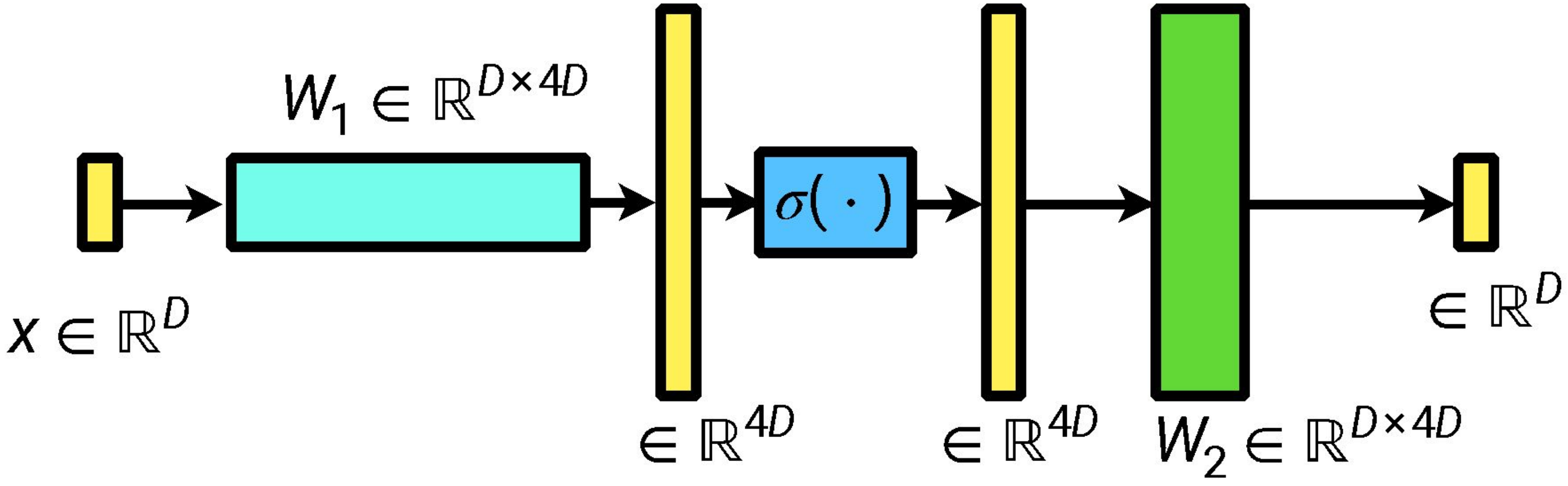
$$
\begin{aligned}
\mathbf{T}^1 &= \mathrm{SelfAttention}(\mathbf{X}) \\
\mathbf{T}^2 &= \mathbf{X} + \mathbf{T}^1 \\
\mathbf{T}_3 &= \mathrm{LayerNorm}(\mathbf{T}^2) \\
\mathbf{T}^4 &= \mathrm{FFN}(\mathbf{T}^3) \\
\mathbf{T}^5 &= \mathbf{T}^4 + \mathbf{T}^3 \\
\mathbf{H} &= \mathrm{LayerNorm}(\mathbf{T}^5)
\end{aligned}
$$

Both X and H are of N X d

Blank Slide

Blank Slide

MLP/Feed Forward Layer



$$\mathrm{MLP}(x) = W_2\, \sigma(W_1 x + b_1) + b_2$$

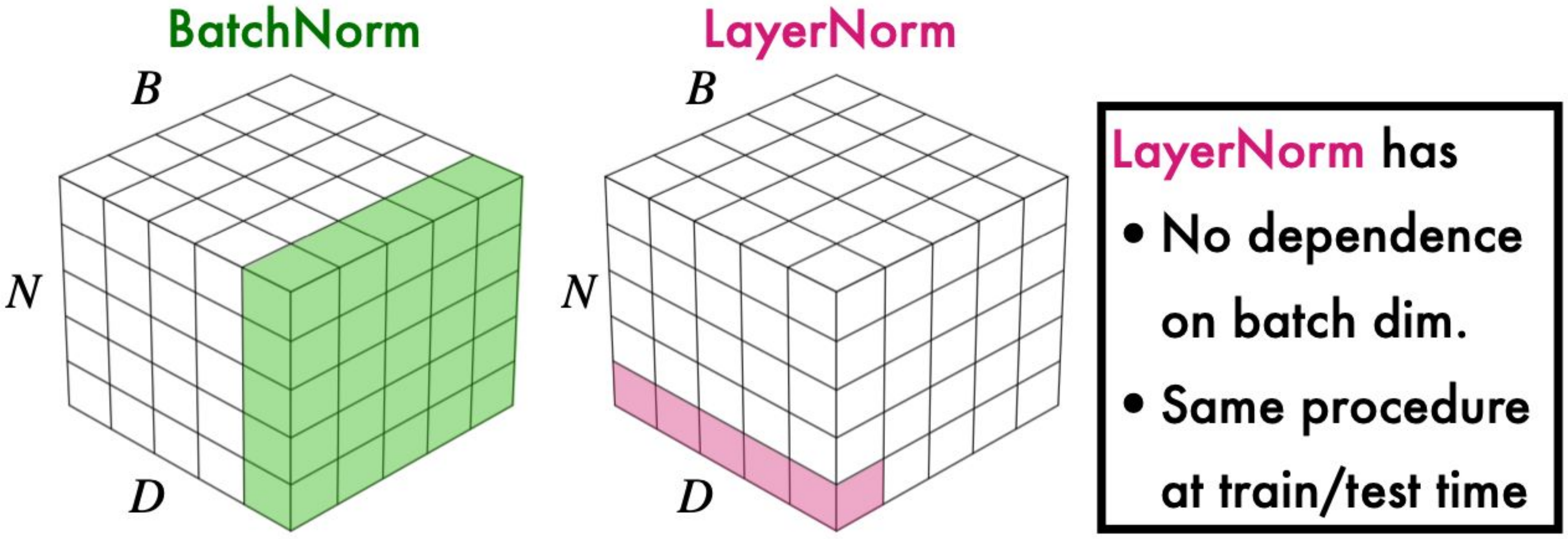
Layer Norm Enhances the Stability

$$\mu = \frac{1}{d_h}\sum_{i=1}^{d_h} x_i$$

$$\sigma = \sqrt{\frac{1}{d_h}\sum_{i=1}^{d_h}(x_i - \mu)^2}$$

$$\hat{\mathbf{x}} = \frac{(\mathbf{x} - \mu)}{\sigma}$$

$$\text{LayerNorm} = \gamma\,\hat{\mathbf{x}} + \beta$$



Positional Encoding

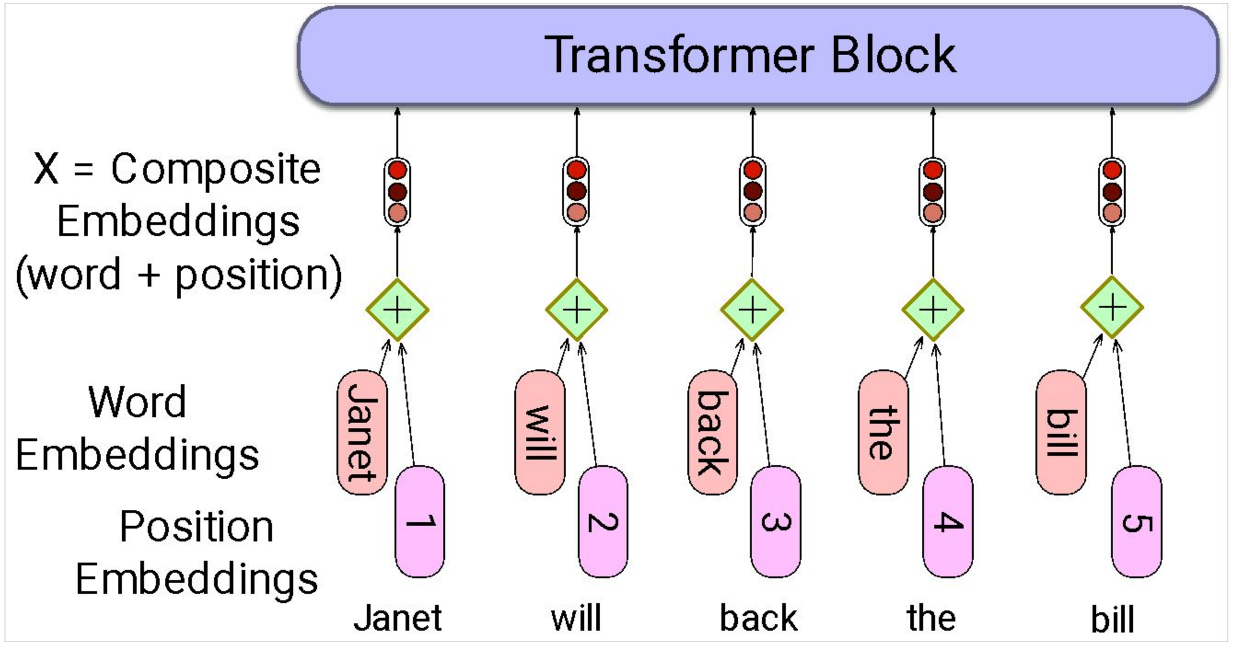
- Position embedding soon after input enbedding
- “Bag of words” to “ordered words”
- Integer (position) to a vector of say 100s

Positional embedding

– Embedding to distinguish each position

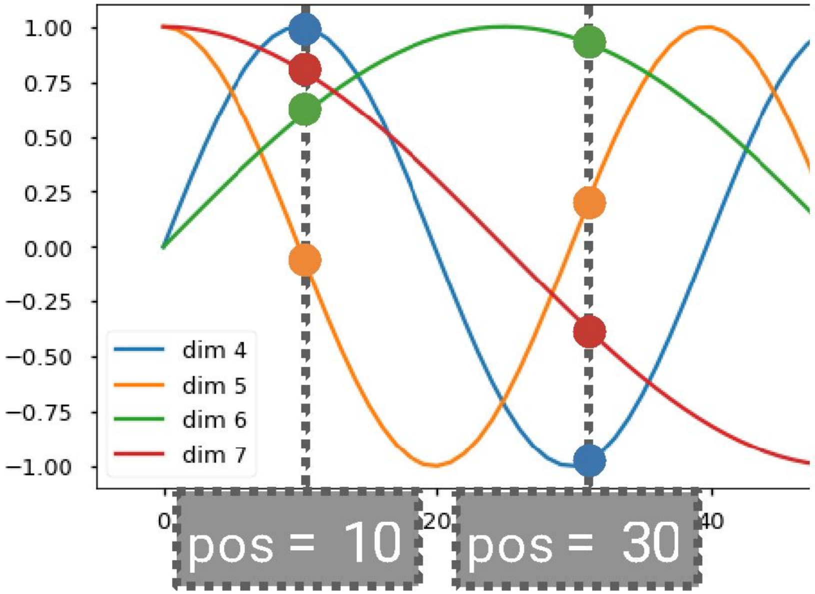
$$PE_{position,2i} = \sin(position/10000^{2i/d})$$
$$PE_{position,2i+1} = \cos(position/10000^{2i/d})$$

Position Embedding



$$PE(pos, 2i) = \sin\left(\frac{pos}{10{,}000^{2i/D}}\right)$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10{,}000^{2i/D}}\right)$$



Absolute Positions

Choose static functions (say sinusoids) that map integer inputs to real valued vectors that captures inherent relationship between vectors

Blank Slide

With Positional Encoding

One query per input vector

puts:

put vectors: X (Shape: $N_X \times D_Q$)

ey matrix: $W_K(Shape: D_X \times D_Q)$

alue matrix: $W_V(Shape: D_X \times D_V)$

uery matrix: $\mathbf{W_Q}$ (Shape: $D_Q \times D_Q$)

omputation:

uery Vectors $Q = XW$

y vectors: $K = XW$ (Shape: $N \times D$)

lue Vectors: $V = XW$ (Shape: $N \times D$)

milarities: $E = \frac{!\,\cdot\,!}{\sqrt{\$}}$ (Shape: $N \times N$) $E = (Q \cdot K)/\sqrt{D}$

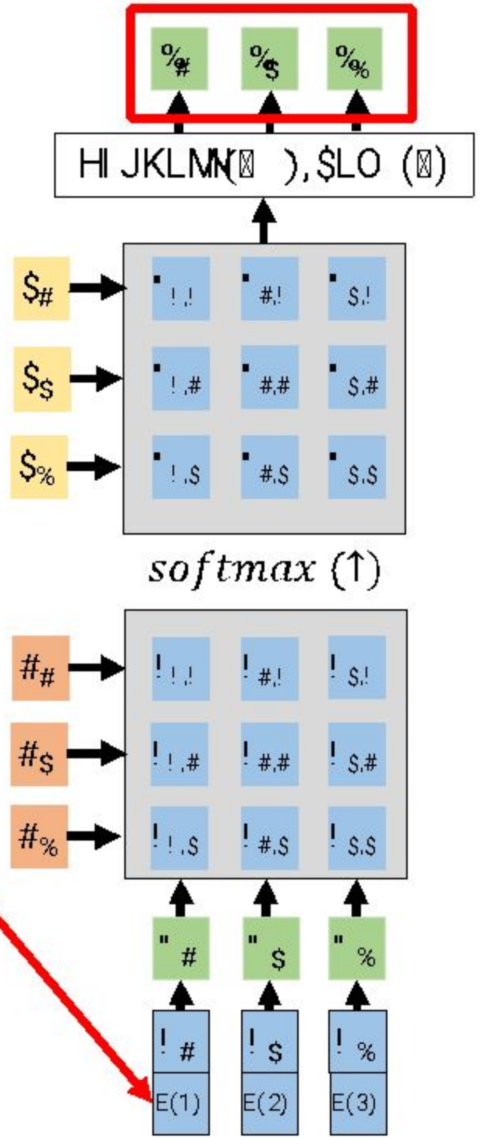
tention weights: $A = softmax(E, \dim = 1)$ (Shape: $N \times N$)

tput vectors: $Y = AV$ (Shape: $N \times D$) $Y = \Sigma A V$

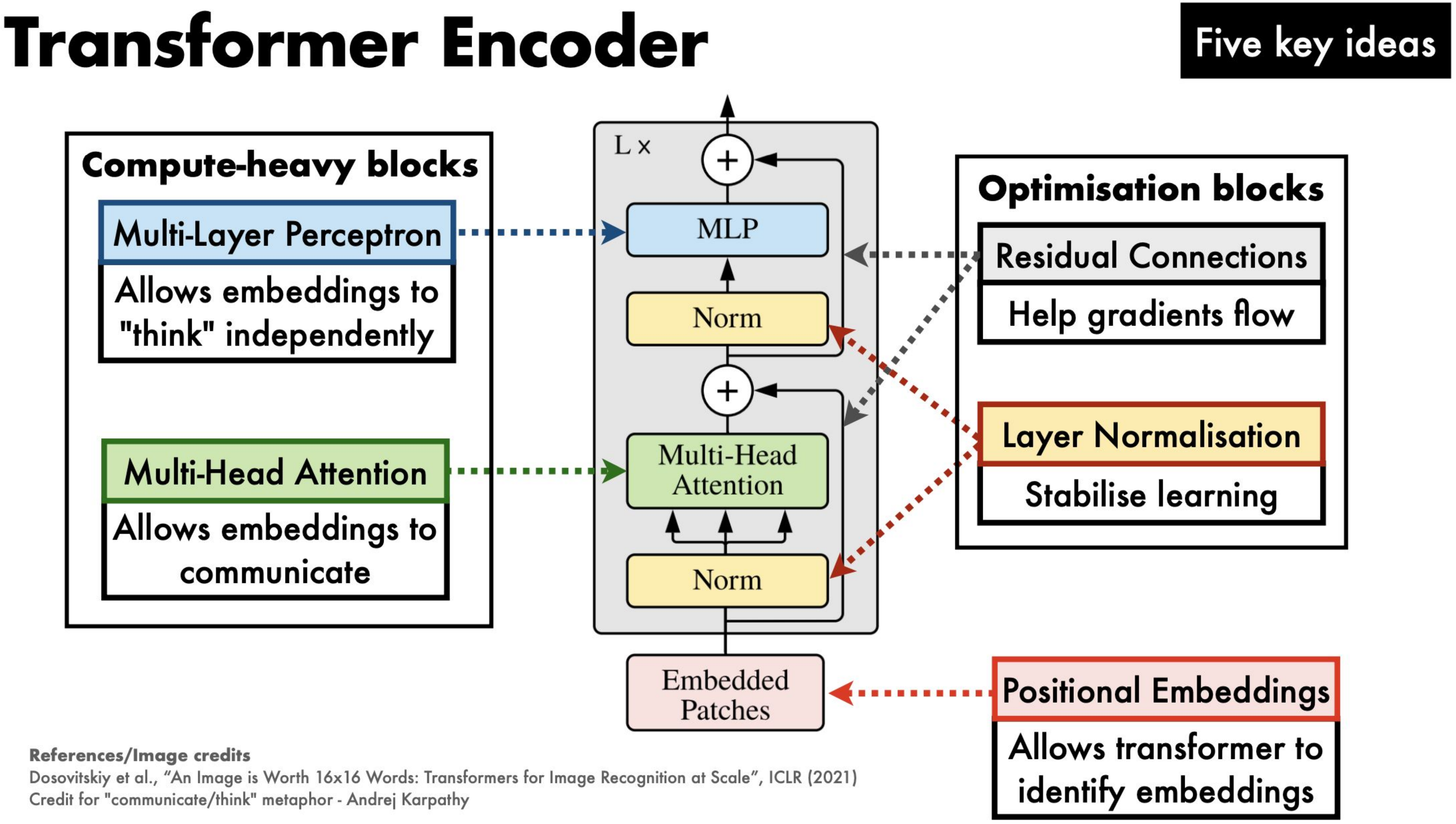
Self attention doesn't "know" the order of the vectors it is processing!

In order to make processing position aware, concatenate or add positional encoding to the input

; can be learned lookup table, or fixed function



Transformers



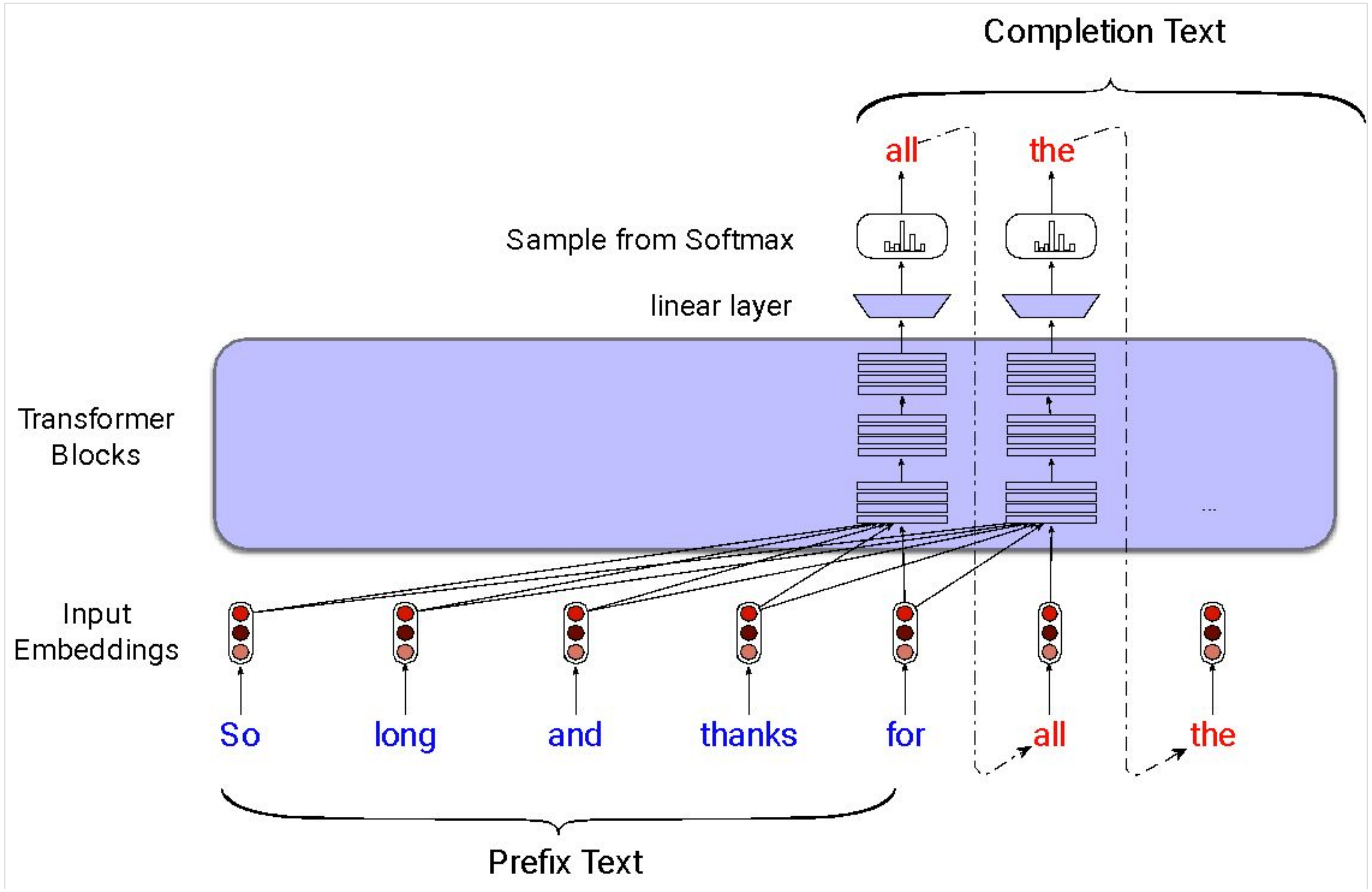
References/Image credits

Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR (2021)

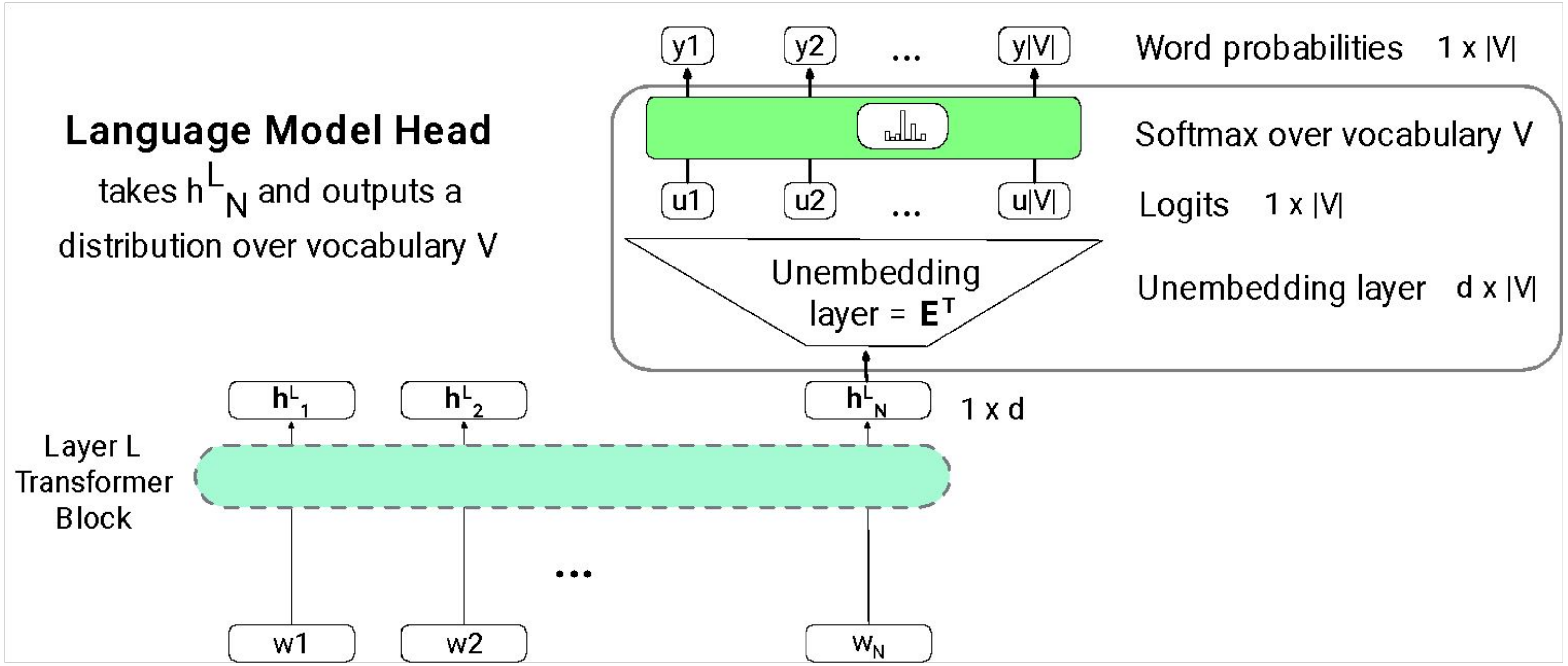
Credit for "communicate/think" metaphor - Andrej Karpathy

Blank Slide

Sentence Completion



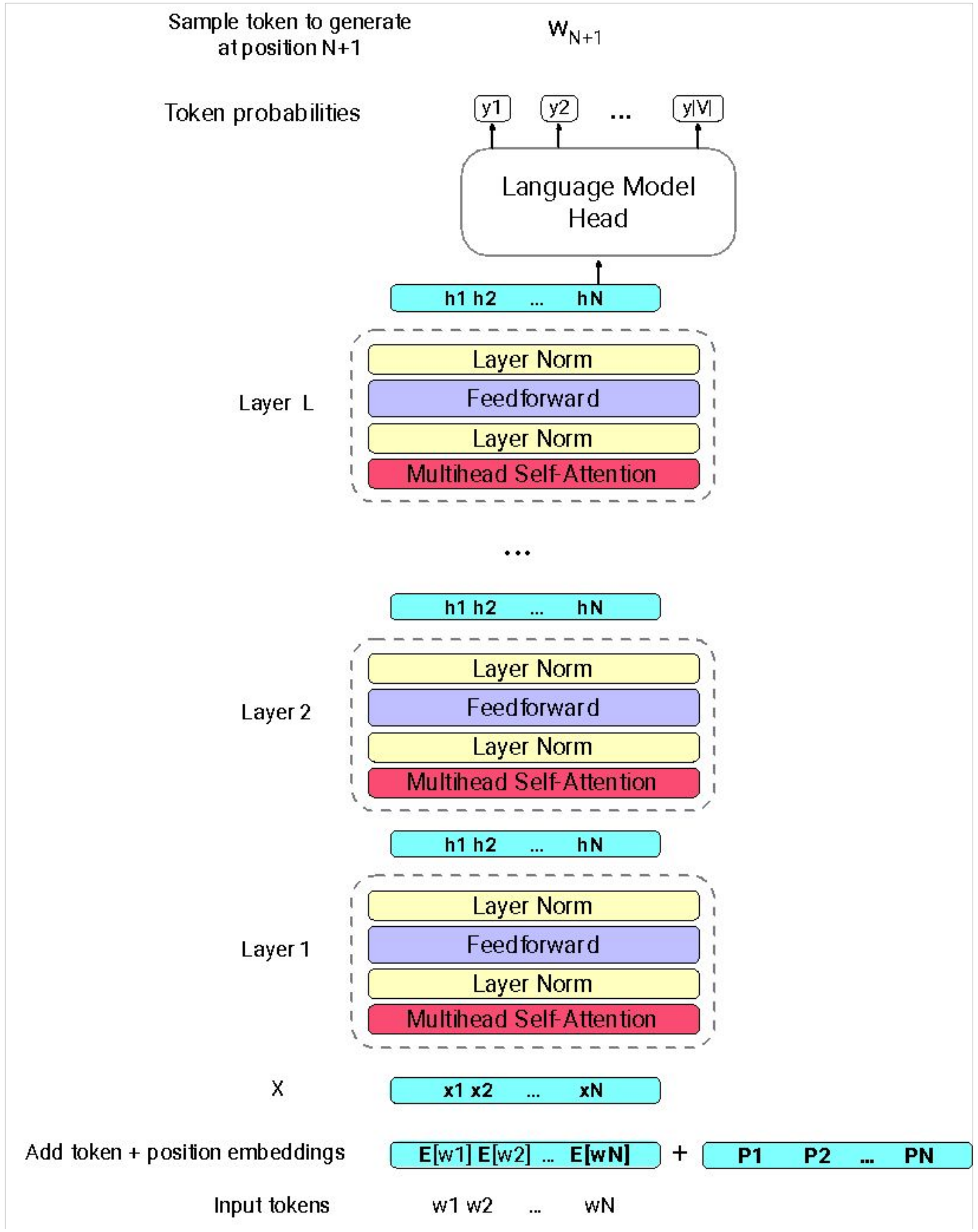
Language Modelling Head



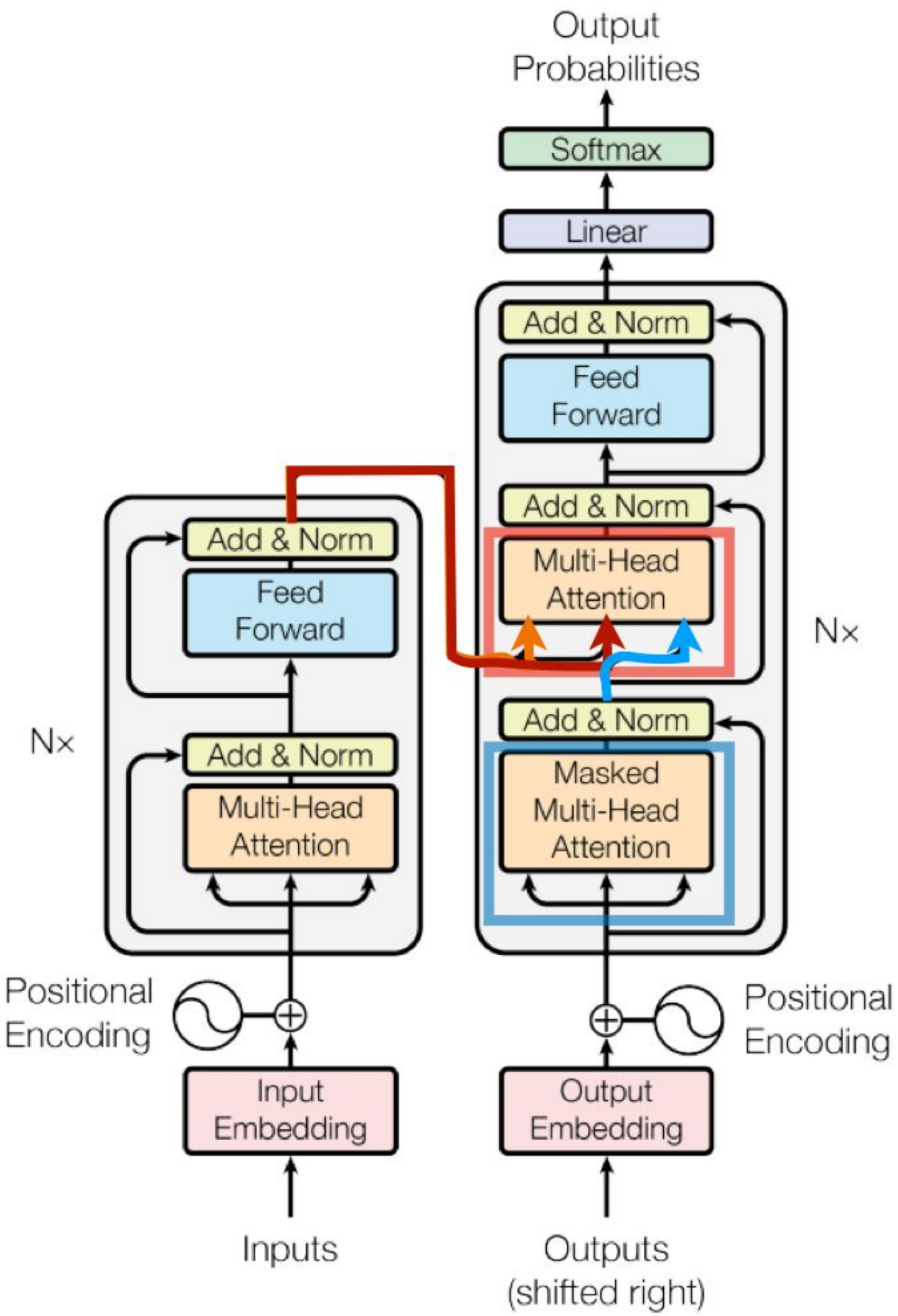
$$u = h^L_N E^T$$

$$y = \text{softmax}(u)$$

Blank Slide



Eg. A Complete Decoder Model



Cross Attention

Query from one sequence and key + value from other (in translation)

Thanks!!

Questions?