

## Transformer Network Motivation

Complexity is increasing

Recurrent Neural Network

Gated Recurrent Unit

Long Short Term Memory

All of the above models are still sequential models, which ingest one word/token of a sentence at a time as if each unit is a bottleneck to the flow of information. For example, in order to compute  $\hat{y}^{(t)}$ , all previous units  $\hat{y}^{(1)}, \dots, \hat{y}^{(t-1)}$  must be computed beforehand.

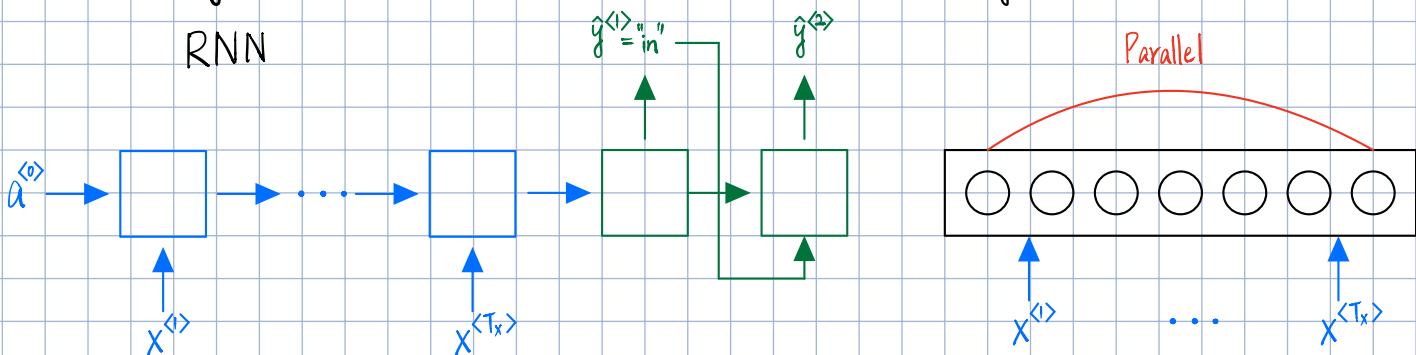
In contrast, Transformer can run an entire sequence in parallel. So, I can ingest an entire sequence all at the same time, rather than processing one word at a time from left to right.

Also, I think Transformer is not a type of RNN, because it doesn't pass its internal activations/states of the current time step  $t$  to the next time step  $t+1$ .

Read 2017 "Attention Is All You Need."

## Transformer Network Intuition

- Combining Attention-based representation and CNN-style processing.



- Self - Attention
- Multi-Head Attention

## Self - Attention

Self - Attention is a way to create attention-based representations for each words in the input sentence.

Use "Jane visite l'Afrique en septembre." as an example.

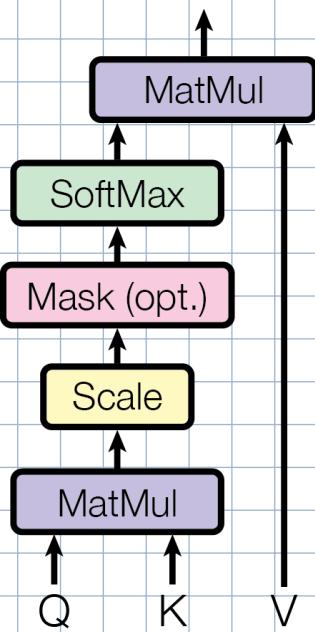
query  
Key  
Value

$A(q, K, V)$ : attention-based vector representation of a word

We want to calculate  $A^{(1)}, A^{(2)}, A^{(3)}, A^{(4)}$  and  $A^{(5)}$  since there are five words in the input sequence.

The word embedding representation, but not one-hot representation, will be used for  $X^{(1)} \text{ to } X^{(5)}$ .

Let's use "l'Afrique" to illustrate self-attention. I could just use the standard word embedding  $x^{(3)}$  to represent the feature of "l'Afrique". But depending on the context, is "l'Afrique" as a site of historical interest, or as a holiday destination, or as the world's second largest continent. Depending on how "l'Afrique" is associated in a sentence (the context), I may choose to represent "l'Afrique" differently, and that's what  $A^{(3)}$  is doing.



Query = interesting questions about the words in a sentence.

Key = qualities of words given a Query

Value = specific representations of words given a Query

Now, let's calculate  $A^{(3)}$ . First, Recall RNN Attention  $\alpha^{(t,t')} = \frac{\exp(e^{(t,t')})}{\sum_{t'=1}^{T_x} \exp(e^{(t,t')})}$

The Transformer Self-Attention:

$$A(q, K, V) = \sum_i \frac{\exp(q \cdot k^{(i)})}{\sum_j \exp(q \cdot k^{(j)})} v^{(i)}$$

Based on the paper, the dimensions of each element are:

$$X^{(3)} \in \mathbb{R}^{d_{\text{model}}}, d_{\text{model}} = 512$$

$$\text{query} \in \mathbb{R}^{d_k}, \text{key} \in \mathbb{R}^{d_k}$$

$$\text{value} \in \mathbb{R}^{d_v}, d_k = d_v = \frac{d_{\text{model}}}{h} = 64$$

$X^{(3)}$  is the word embedding for "l'Afrique". And  $q^{(3)}$ ,  $k^{(3)}$ , and  $v^{(3)}$  can be computed as: where  $W^Q$ ,  $W^K$  and  $W^V$  are learned parameters/matrix of the Transformer, and they can pull out  $q^{(3)}$ ,  $k^{(3)}$ , and  $v^{(3)}$  vectors for each word.

$$\begin{aligned} q^{(3)} &= W^Q X^{(3)} \\ k^{(3)} &= W^K X^{(3)} \\ v^{(3)} &= W^V X^{(3)} \end{aligned}$$

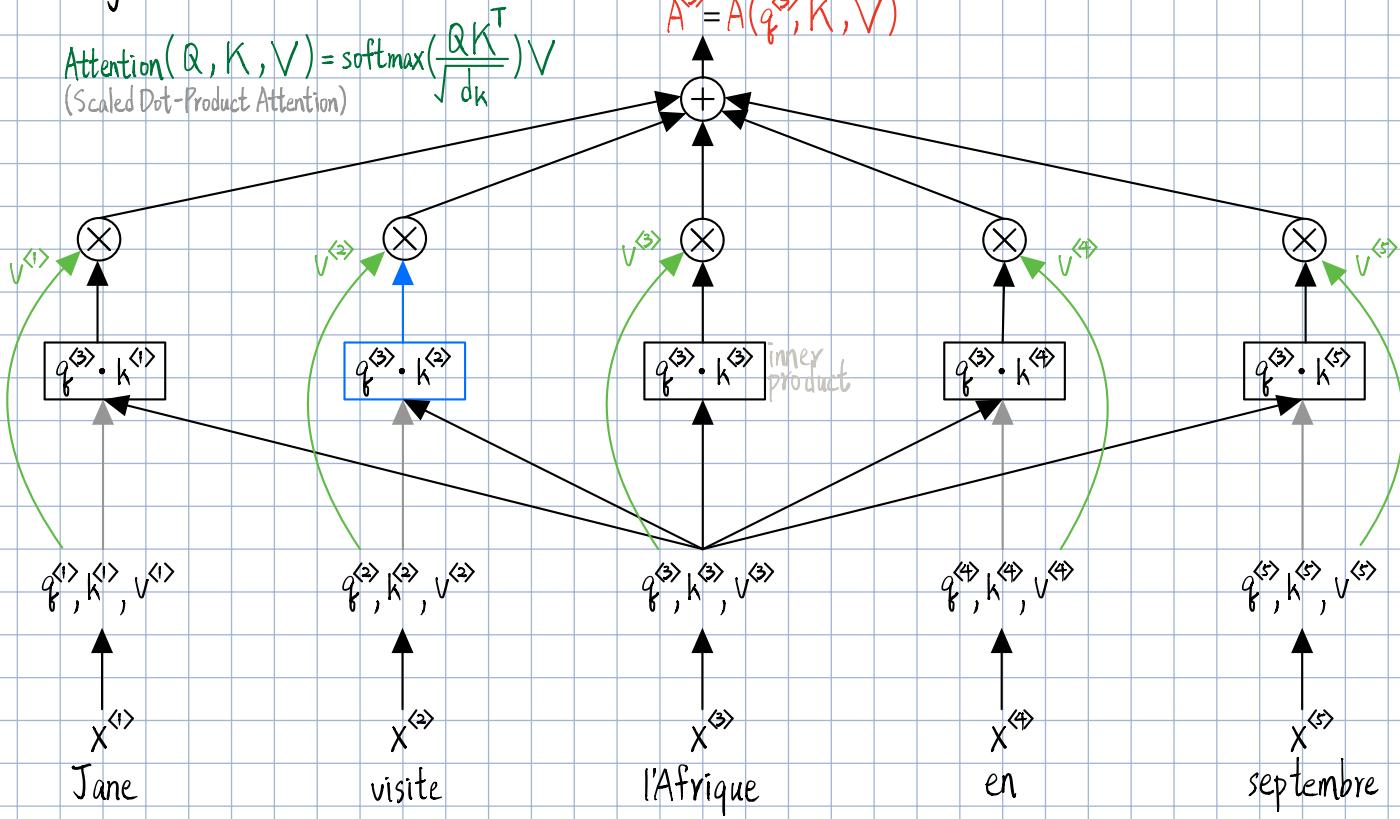
Intuitively,  $q^{(3)}$  is like a question that I get to ask about "l'Afrique". Eg,  $q^{(3)}$  may present a question "what's happening there?" And the inner product between  $q^{(3)}$  and  $k^{(3)}$  can tell how good is "visite" to the question "what's happening in Africa?"

The advantage of self-attention is that the word "l'Afrique" isn't a fixed word embedding anymore. Instead, the self-attention mechanism realizes that "l'Afrique" is a destination of "visite".

All representations  $A^{(1)}$  to  $A^{(5)}$  can be written altogether as: vectorization

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (\text{Scaled Dot-Product Attention})$$

$$A^{(3)} = A(q^{(3)}, K, V)$$



## Multi-Head Attention

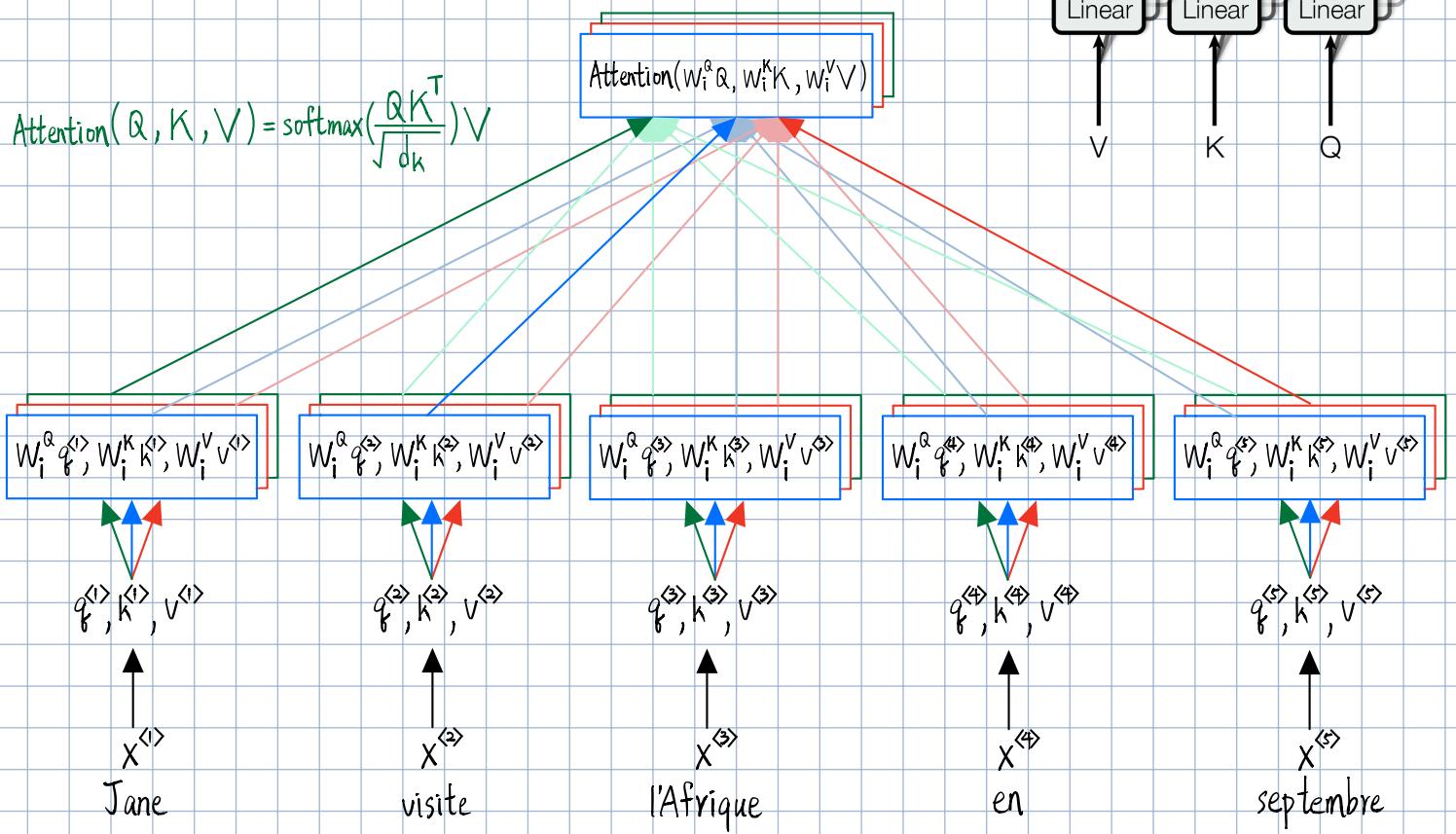
Each time when a self-attention for a sequence is calculated, it is called a **head**.

$h$ : # heads, number of heads

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^0$$

where  $\text{head}_i = \text{Attention}(W_i^Q Q, W_i^K K, W_i^V V)$  the equation here is slightly different from the paper

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{dk}}\right)V$$

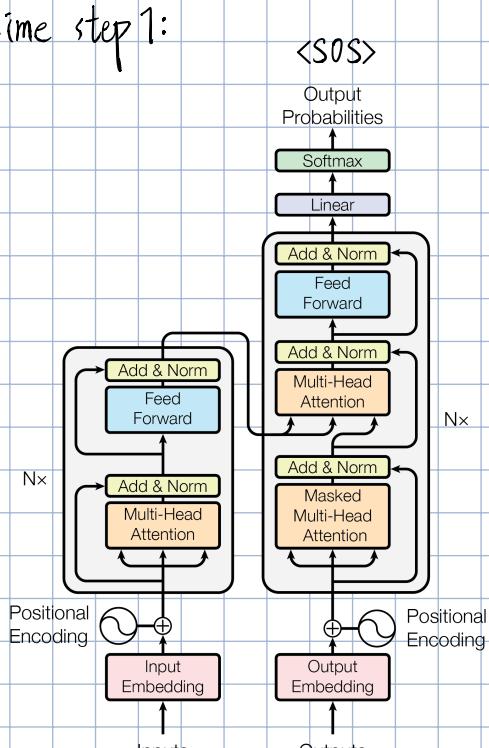


An intuitive explanation could be that each set of  $W_i^Q, W_i^K, W_i^V$  query a different question.  
 For instance,  $W_1^Q, W_1^K, W_1^V$ : What's happening?  $W_2^Q, W_2^K, W_2^V$ : When?  $W_3^Q, W_3^K, W_3^V$ : Who?

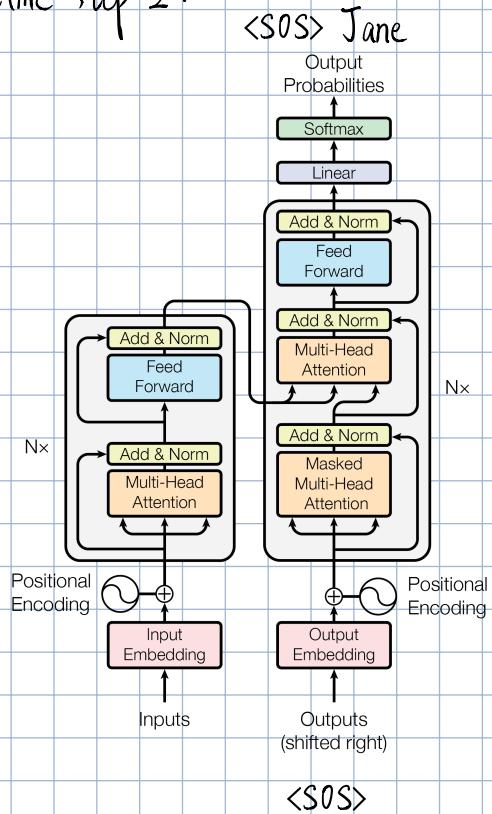
# Transformer

Now, we can put all the pieces together and form a Transformer. The input sentence is "Jane visite l'Afrique en septembre." input: <SOS> X<sup>(1)</sup> X<sup>(2)</sup> X<sup>(3)</sup> X<sup>(4)</sup> X<sup>(5)</sup> <EOS>

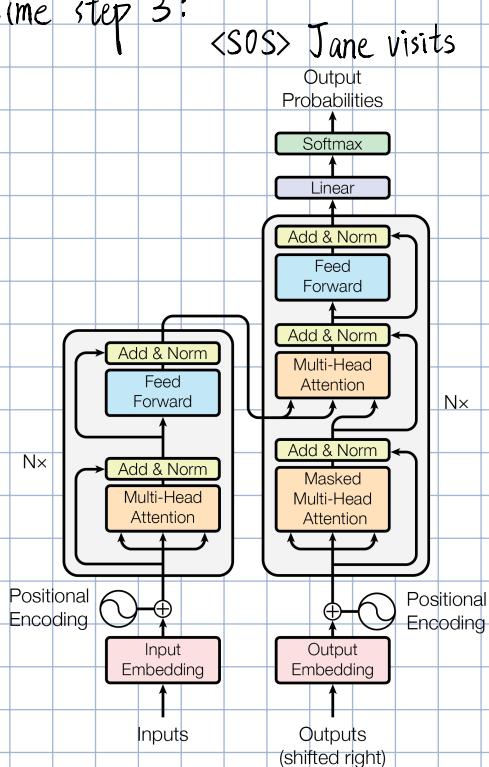
time step 1:



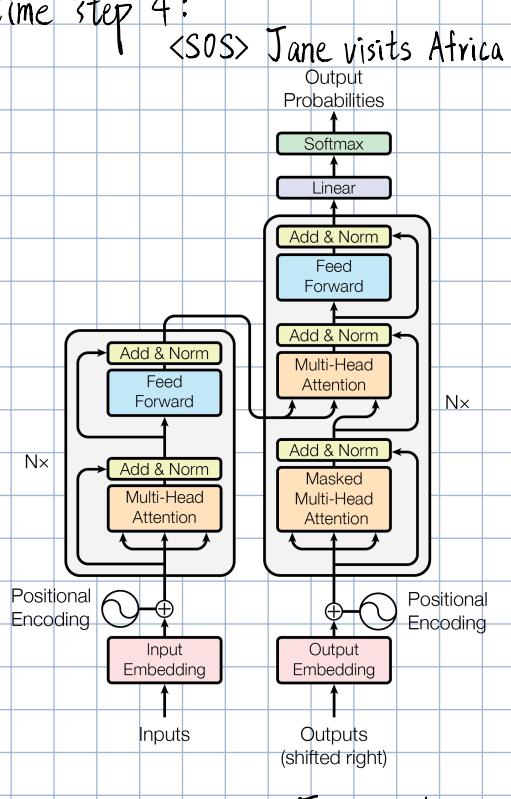
time step 2:



time step 3:



time step 4:



<SOS> Jane

<SOS> Jane visits

## Positional Encoding

$$PE_{(pos, 2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

$$PE_{(pos, 2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

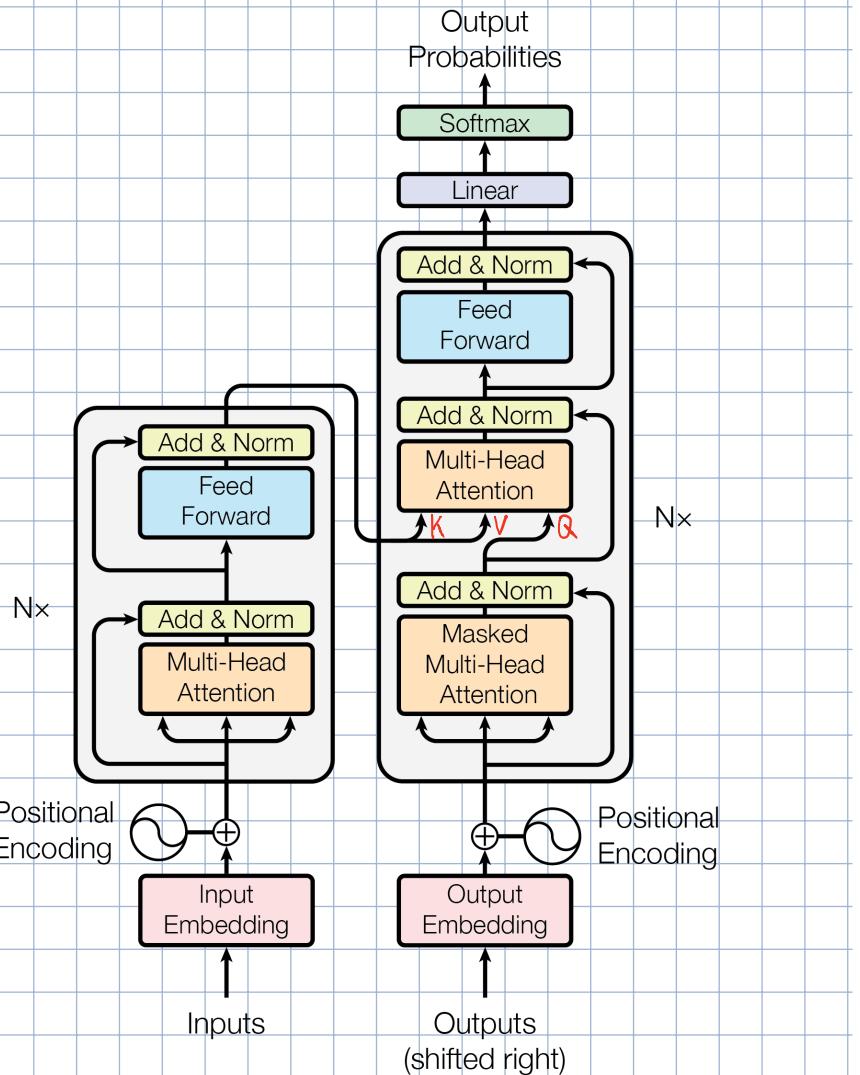
"Jane visite l'Afrique en septembre."

pos : the numerical position of a word

i : the i-th dimension in the encoding vector

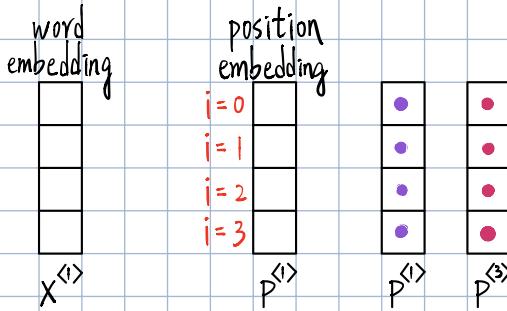
For "Jane", pos is 1.

For "l'Afrique", pos is 3.



Recall the self-attention equation, there is nothing that indicates the position of a word. But the position of a word within a sentence can be extremely important to machine translation. Now, let's use sine and cosine equations to encode the position of a word in a sentence.

Eg: assume the dimension of the word embedding  $d_{model} = 4$



$X^{(1)} + P^{(1)}$  now has the semantic and positional information.

