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 $\hat{y}^{\langle t \rangle}$, all previous units $\hat{y}^{\langle 1 \rangle}, \ldots, \hat{y}^{\langle t-1 \rangle}$ must be computed beforehand.

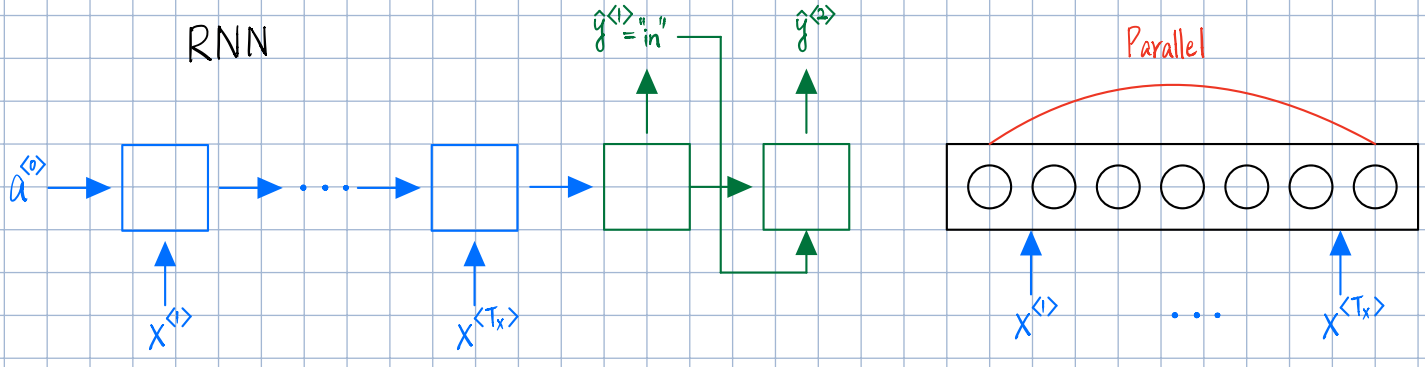
In contrast, Transformer can run an entire sequence in parallel. So, I can ingest an entire sequence all at the same time, rather than processing one word at a time from left to right.

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.

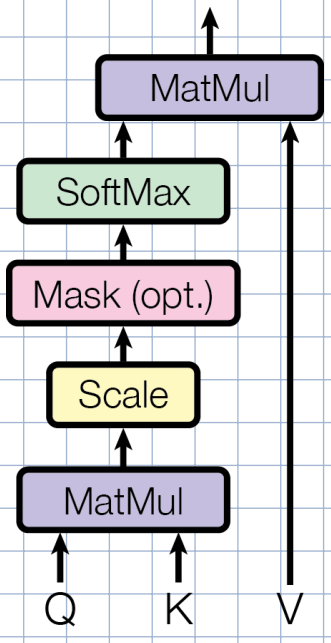
query
Key
Value

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

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

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

Let's use "l'Afrique" to illustrate self-attention. I could just use the standard word embedding $X^{\langle 3 \rangle}$ 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 $A^{\langle 3 \rangle}$ 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 $A^{\langle 3 \rangle}$. First, Recall RNN Attention $\alpha^{\langle t,t' \rangle} = \frac{\exp(e^{\langle t,t' \rangle})}{\sum_{t'=1}^{T_x} \exp(e^{\langle t,t' \rangle})}$

$q \cdot k^{\langle i \rangle}$ is akin to $e^{\langle t,t' \rangle}$

The Transformer Self-Attention:

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

Query (Q) Key (K) Value (V)

$q^{\langle 1 \rangle}$ $k^{\langle 1 \rangle}$ → person $v^{\langle 1 \rangle}$

$q^{\langle 2 \rangle}$ $k^{\langle 2 \rangle}$ → action $v^{\langle 2 \rangle}$

what's happening there ← $q^{\langle 3 \rangle}$ $k^{\langle 3 \rangle}$ $v^{\langle 3 \rangle}$

$q^{\langle 4 \rangle}$ $k^{\langle 4 \rangle}$ $v^{\langle 4 \rangle}$

$q^{\langle 5 \rangle}$ $k^{\langle 5 \rangle}$ $v^{\langle 5 \rangle}$

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

$x^{\langle 3 \rangle} \in \mathbb{R}^{d_{model}}$, $d_{model} = 512$

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

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

$x^{\langle 3 \rangle}$ is the word embedding for "l'Afrique". And $q^{\langle 3 \rangle}$, $k^{\langle 3 \rangle}$, and $v^{\langle 3 \rangle}$ can be computed as: $q^{\langle 3 \rangle} = W^Q x^{\langle 3 \rangle}$, $k^{\langle 3 \rangle} = W^K x^{\langle 3 \rangle}$, $v^{\langle 3 \rangle} = W^V x^{\langle 3 \rangle}$

, where W^Q, W^K and W^V are learned parameters/matrix of the Transformer and they can pull out $q^{\langle 3 \rangle}$, $k^{\langle 3 \rangle}$, and $v^{\langle 3 \rangle}$ vectors for each word.

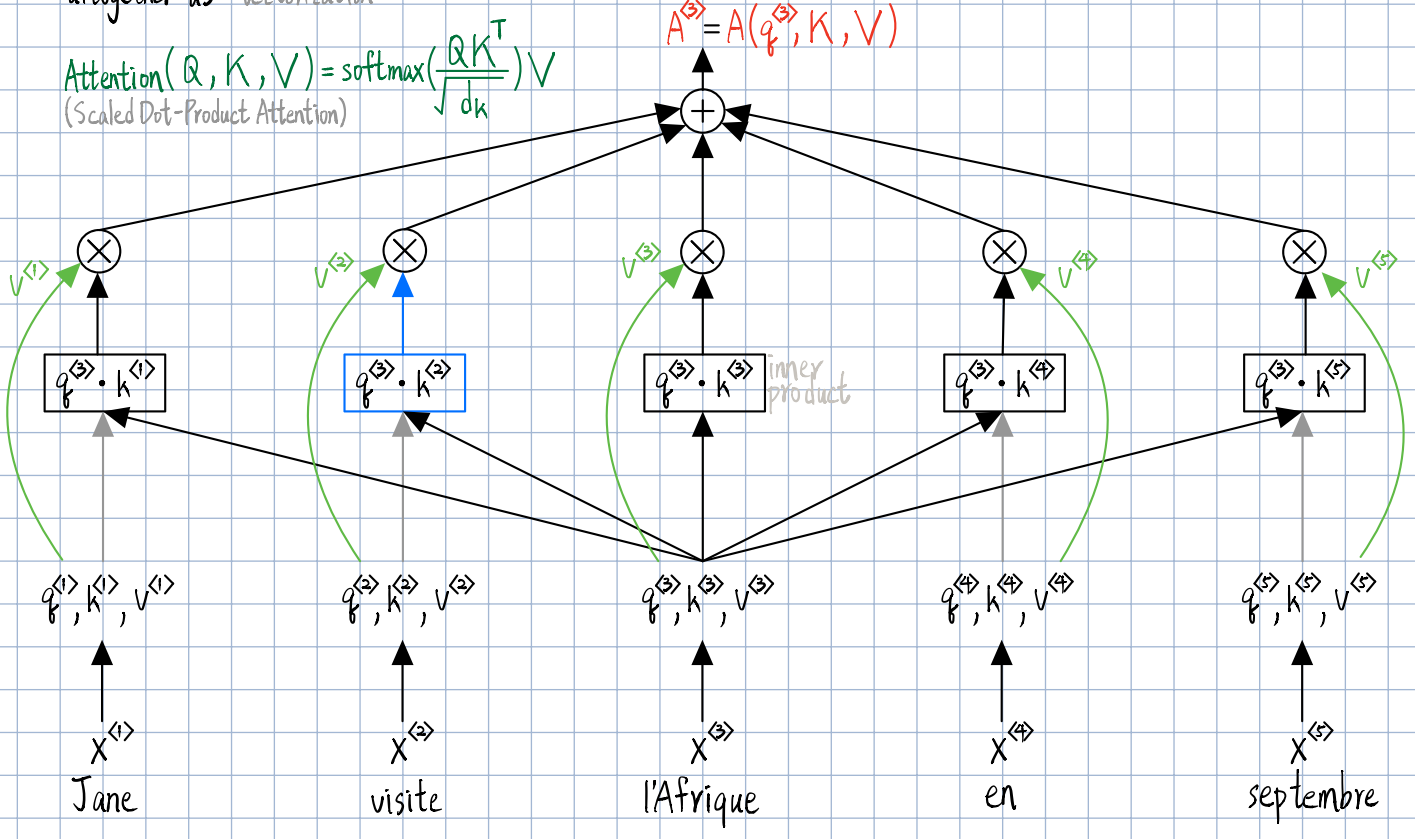
Intuitively, $q^{\langle 3 \rangle}$ is like a question that I get to ask about "l'Afrique". Eg, $q^{\langle 3 \rangle}$ may present a question "what's happening there?" And the inner product between $q^{\langle 3 \rangle}$ and $k^{\langle 2 \rangle}$ can tell how good is "visite" to the question "what's 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 $A^{\langle 1 \rangle}$ to $A^{\langle 5 \rangle}$ can be written altogether as: vectorization

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

(Scaled Dot-Product Attention)



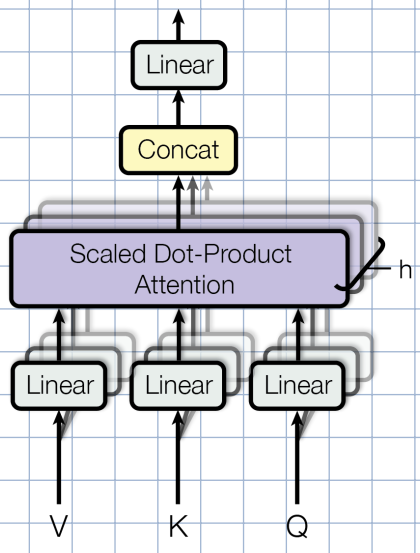
Multi-Head Attention

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

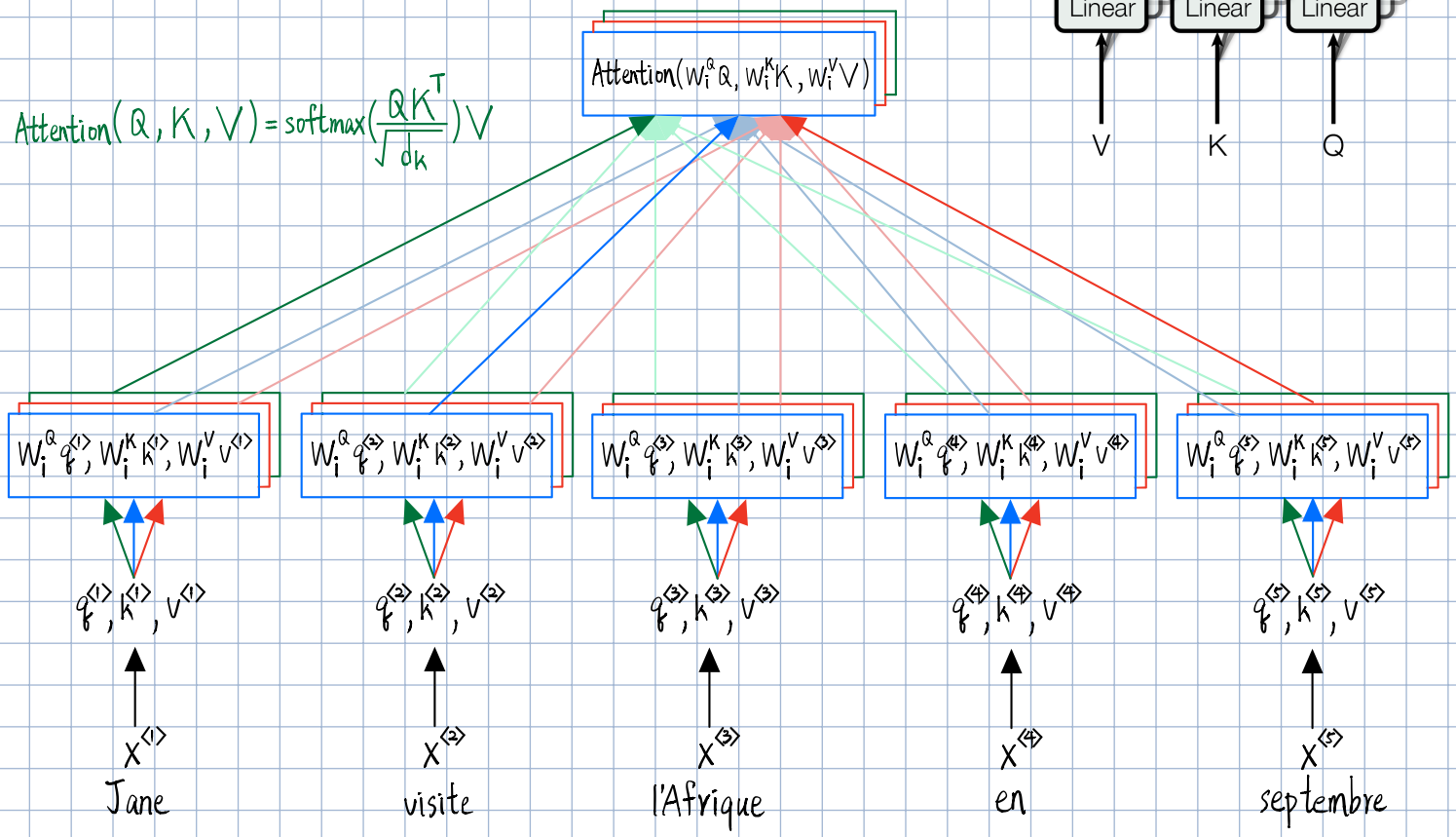
h: # heads, number of heads

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

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



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

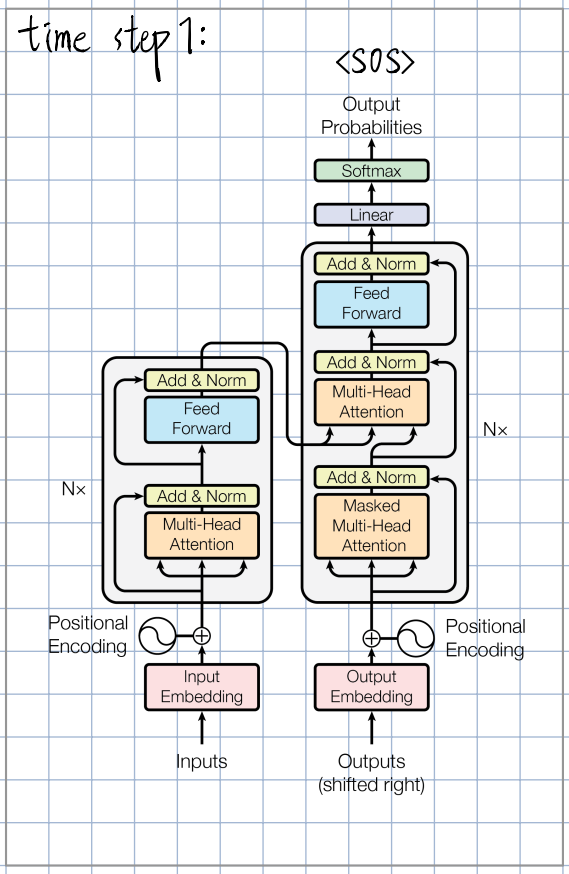


An intuitive explanation could be that each set of W_i^Q, W_i^K, W_i^V query a different question.
For instance, W_1^Q, W_1^K, W_1^V: What's happening? W_2^Q, W_2^K, W_2^V: When? W_3^Q, W_3^K, W_3^V: Who?

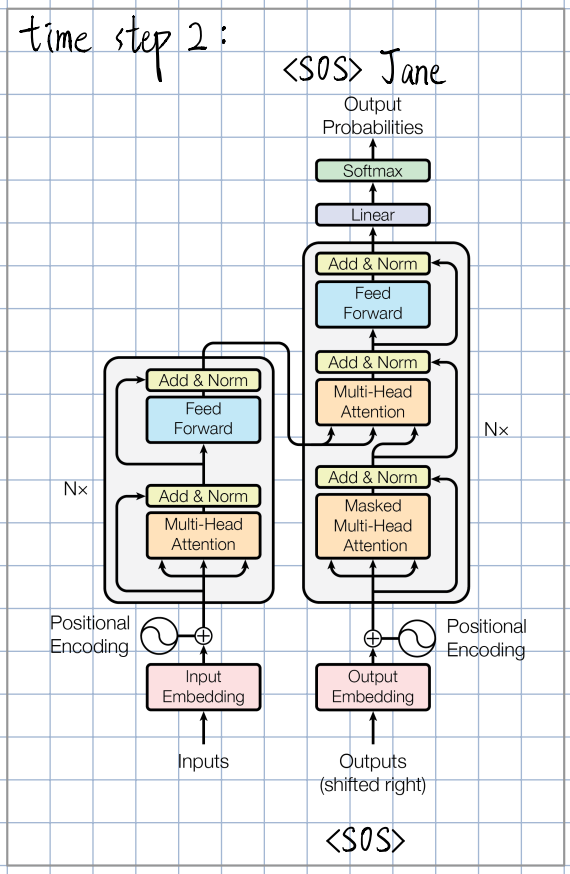
Transformer

Now, we can put all the pieces together and form a Transformer. The input sentence is "Jane visite l'Afrique en septembre." input : <SOS> $X^{\langle 1 \rangle}$ $X^{\langle 2 \rangle}$ $X^{\langle 3 \rangle}$ $X^{\langle 4 \rangle}$ $X^{\langle 5 \rangle}$ <EOS>

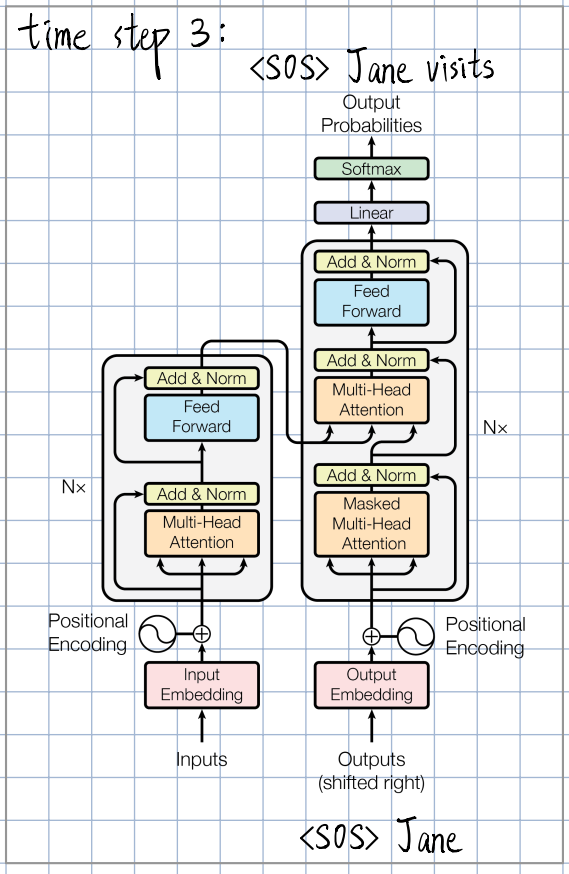
time step 1:



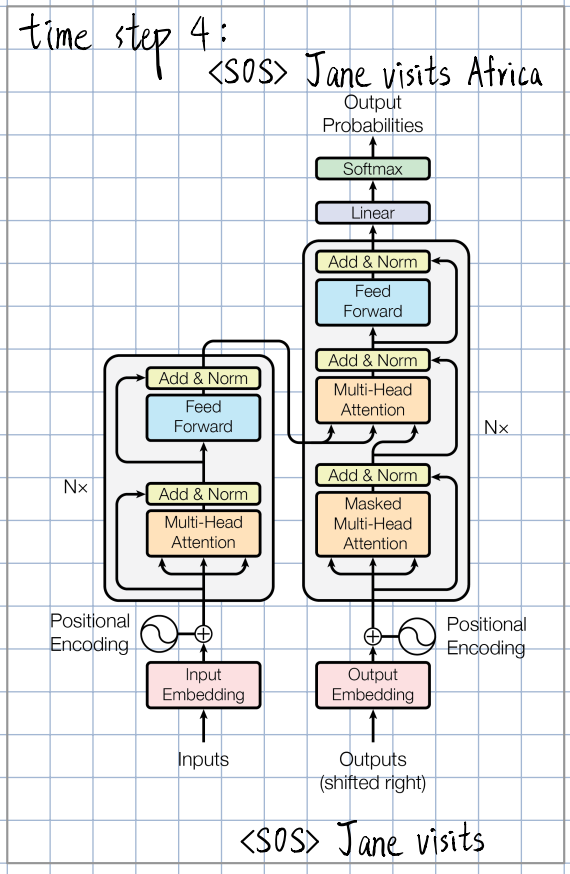
time step 2:



time step 3:



time step 4:



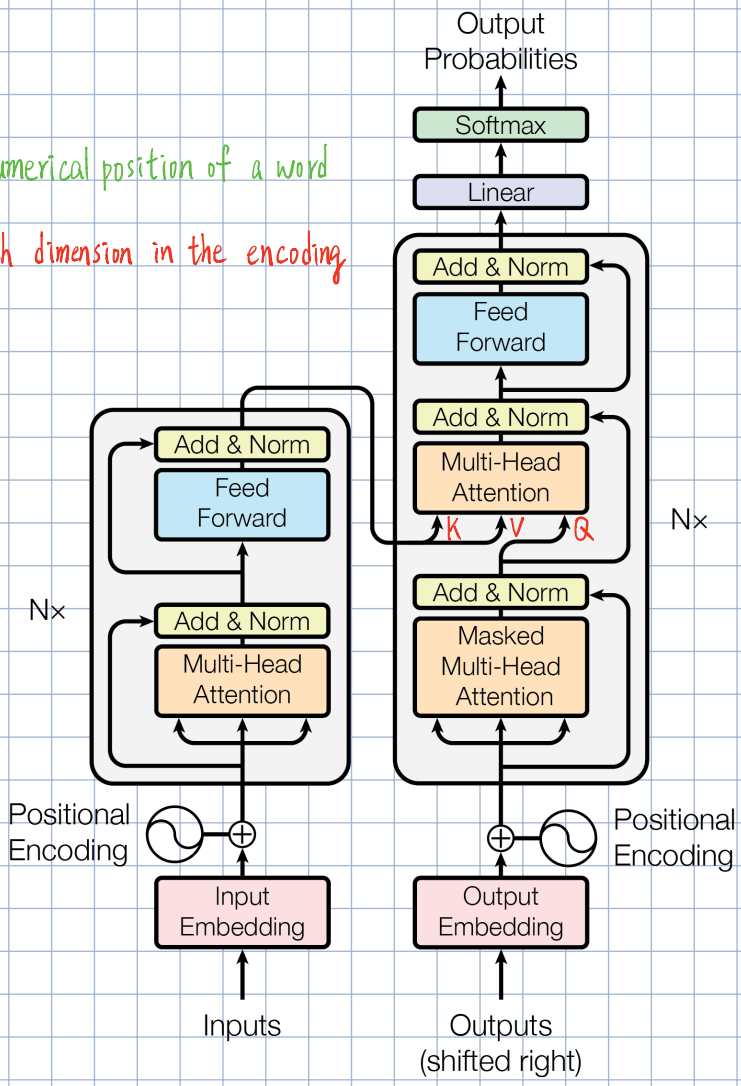
Positional Encoding

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

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

pos : the numerical position of a word

i : the i-th dimension in the encoding vector



"Jane visite l'Afrique en septembre."

For "Jane", pos is 1.

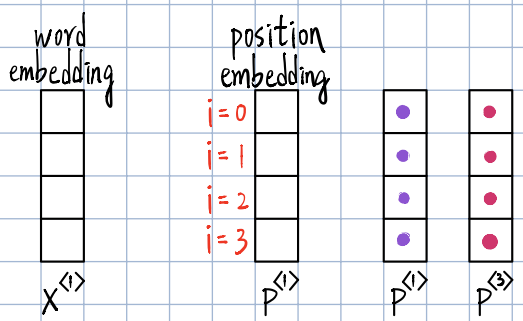
For "l'Afrique", pos is 3.

Criteria for a good positional encoding algorithm:

1. Output a unique encoding for each time step (word's position in a sentence)
2. Distance between any two time steps should be consistent for all sentence length.
3. The algorithm should be able to generalize to longer sentences.

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 $d_{model} = 4$



$x^{\langle 1 \rangle} + P^{\langle 1 \rangle}$ now has the semantic and positional information.