Deep Learning

Transformers

Dr. Mehran Safayani

safayani@iut.ac.ir

safayani.iut.ac.ir

https://www.aparat.com/mehran.safayani

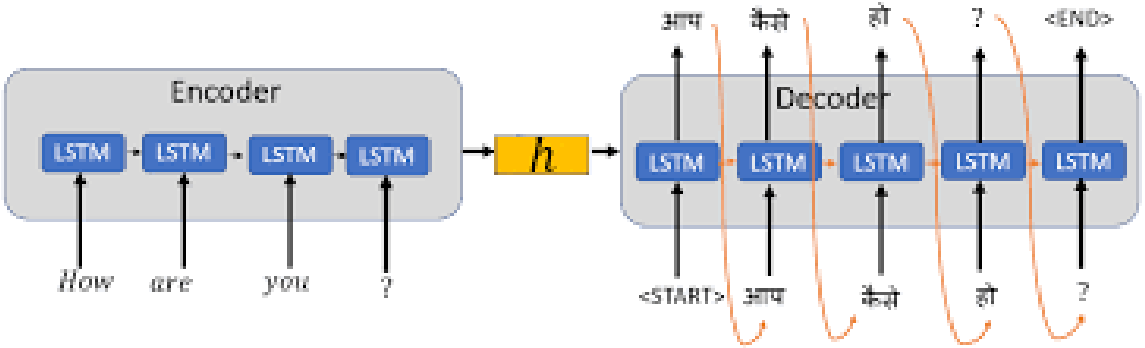
https://github.com/safayani/deep_learning_course

Department of Electrical and computer engineering, Isfahan university of technology, Isfahan, Iran

Attention model

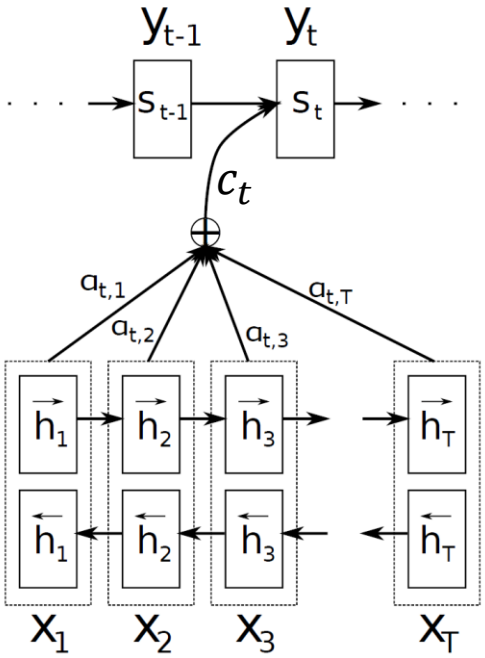
A potential issue with this encoder–decoder approach is that a neural network needs to be able to compress all the necessary information of a source sentence into a fixed-length vector.

This may make it difficult for the neural network to cope with long sentences, especially those that are longer than the sentences in the training corpus.



Dzmitry Bahdanau,etal, NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE, ICLR 2015

Attention model



$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

$$e_{ij} = a(s_{i-1}, h_j)$$

a is a feed forward NN

$$s_i = f(s_{i-1}, y_{i-1}, c_i).$$

$$p(y_i | y_1, \ldots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i),$$

Attention model summary

Each time the proposed model generates a word in a translation, it (soft-)searches for a set of positions in a source sentence where the most relevant information is concentrated. The model then predicts a target word based on the context vectors associated with these source positions and all the previous generated target words.

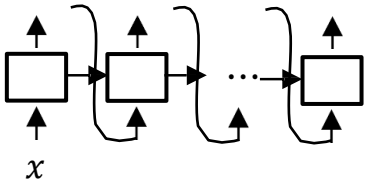
The most important distinguishing feature of this approach from the basic encoder–decoder is that it does not attempt to encode a whole input sentence into a single fixed-length vector. Instead, it encodes the input sentence into a sequence of vectors and chooses a subset of these vectors adaptively while decoding the translation.

Dzmitry Bahdanau,etal, NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE, ICLR 2015

Transformers Motivation

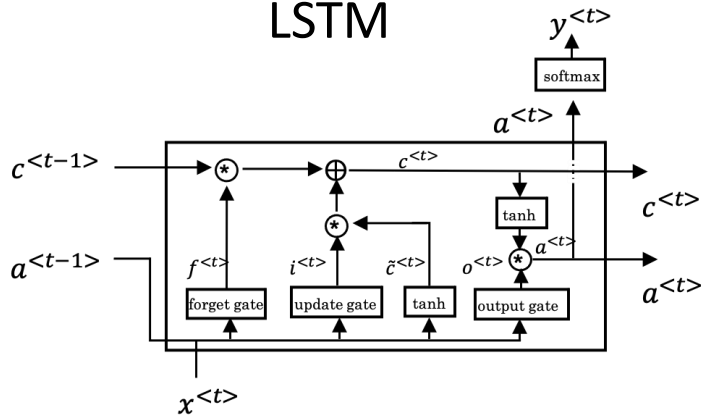
Increased complexity, sequential

RNN



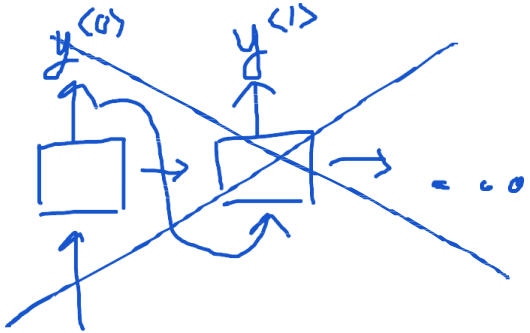
GRU

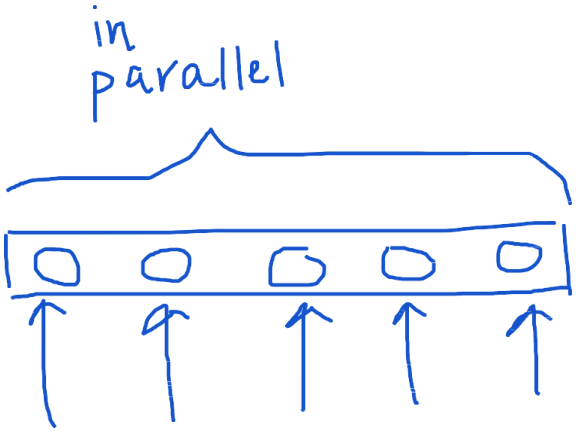
LSTM



Transformers Intuition

- Attention + CNN
  - Self-Attention
  - Multi-Head Attention





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

Self-Attention Intuition

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

RNN Attention

$$\alpha^{<t,t'>} = \frac{\exp(<t,t'>)}{\sum_{t'=1}^{T_x} \exp(<t,t'>)}$$

Transformers Attention

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

$$q^t = W^Q x^t \qquad k^t = W^K x^t \qquad v^t = W^V x^t$$

Query, Key, Value

The key/value/query concept is analogous to retrieval systems. For example, when you search for videos on internet, the search engine will map your **query** (text in the search bar) against a set of **keys** (video title, description, etc.) associated with candidate videos in their database, then present you the best matched videos (**values**).

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

Self-Attention Intuition

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

RNN Attention

$$\alpha^{<t,t'>} = \frac{\exp(<t,t'>)}{\sum_{t'=1}^{T_x} \exp(<t,t'>)}$$

Transformers Attention

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

$$q^t = W^Q x^t \qquad k^t = W^K x^t \qquad v^t = W^V x^t$$

$q^3 intuition$: what's happening in Africa?

$x^{<1>}$	$x^{<2>}$	$x^{<3>}$	$x^{<4>}$	$x^{<5>}$
Jane	visite	l'Afrique	en	septembre

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

Self-Attention Intuition

$q^t = W^Q x^t$ $k^t = W^K x^t$ $v^t = W^V x^t$

q^3 : what's happening in Africa?

$k^{<1>}$: person $v^{<1>}$: jane embedding

$k^{<2>}$: action $v^{<2>}$: visit embedding

Query (Q)	Key (K)	Value (V)
$q^{<1>}$	$k^{<1>}$	$v^{<1>}$
$q^{<2>}$	$k^{<2>}$	$v^{<2>}$
$q^{<3>}$	$k^{<3>}$	$v^{<3>}$
$q^{<4>}$	$k^{<4>}$	$v^{<4>}$
$q^{<5>}$	$k^{<5>}$	$v^{<5>}$

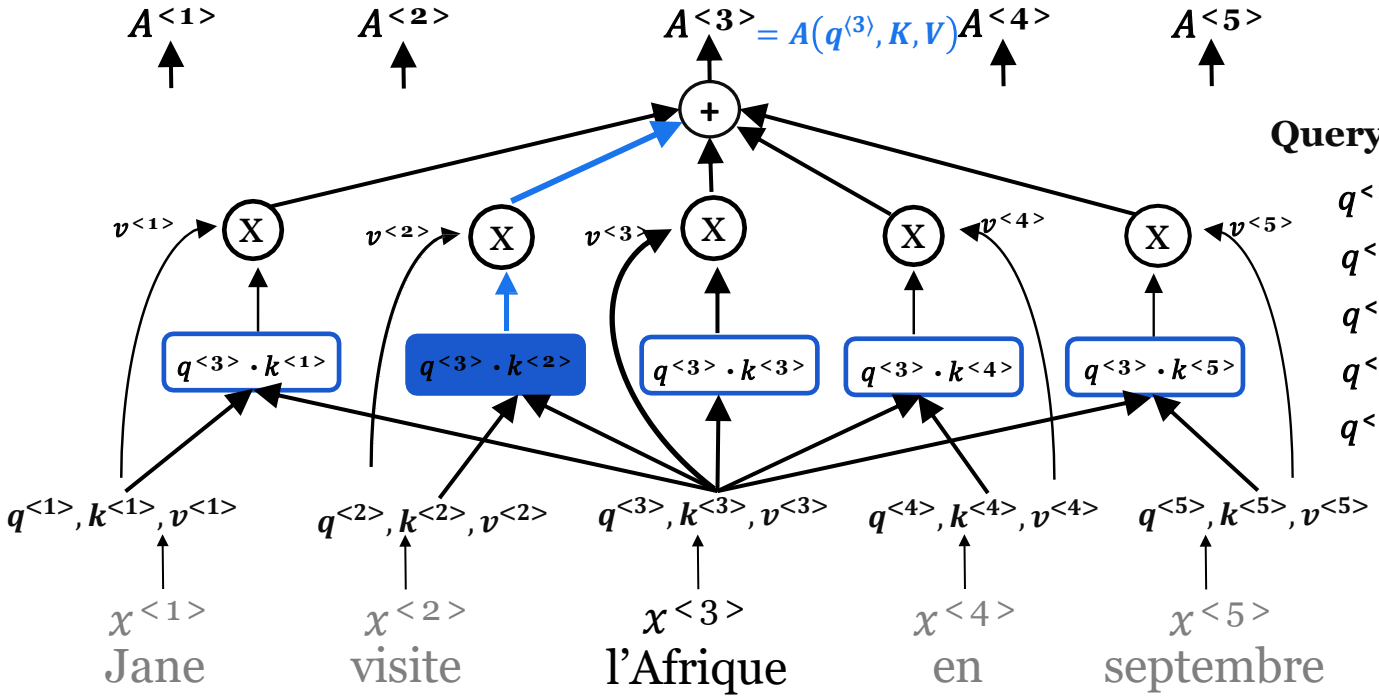
$x^{<1>}$	$x^{<2>}$	$x^{<3>}$	$x^{<4>}$	$x^{<5>}$
Jane	visite	l'Afrique	en	septembre

Self-Attention

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

If you put all of these computation together:

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



Contextual Embedding

Query (Q)	Key (K)	Value (V)
$q^{<1>}$	$k^{<1>}$ person	$v^{<1>}$
$q^{<2>}$	$k^{<2>}$ action	$v^{<2>}$
$q^{<3>}$ What's happening there	$k^{<3>}$	$v^{<3>}$
$q^{<4>}$	$k^{<4>}$	$v^{<4>}$
$q^{<5>}$	$k^{<5>}$	$v^{<5>}$

$$q^{\langle 3 \rangle} = W^Q \cdot x^{\langle 3 \rangle}$$
$$k^{\langle 3 \rangle} = W^k \cdot x^{\langle 3 \rangle}$$
$$v^{\langle 3 \rangle} = W^v \cdot x^{\langle 3 \rangle}$$

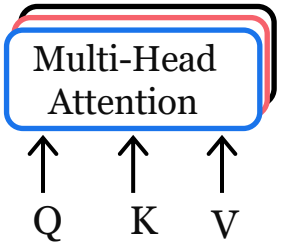
[Vaswani et al. 2017, Attention

Multi-Head Attention

first head: W_1^Q, W_1^K, W_1^V **- What's happening in Africa?**

$$\textbf{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \textbf{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{\boldsymbol{d}_k}}\right)\boldsymbol{V}$$

MultiHead(Q,K,V)



$\textbf{Attention}(W_1^Q Q, W_1^K K, W_1^V V)$

$W_1^Q q^{<1>}, W_1^K k^{<1>}, W_1^V v^{<1>}$	$W_1^Q q^{<2>}, W_1^K k^{<2>}, W_1^V v^{<2>}$	$W_1^Q q^{<3>}, W_1^K k^{<3>}, W_1^V v^{<3>}$	$W_1^Q q^{<4>}, W_1^K k^{<4>}, W_1^V v^{<4>}$	$W_1^Q q^{<5>}, W_1^K k^{<5>}, W_1^V v^{<5>}$
$q^{<1>}, k^{<1>}, v^{<1>}$	$q^{<2>}, k^{<2>}, v^{<2>}$	$q^{<3>}, k^{<3>}, v^{<3>}$	$q^{<4>}, k^{<4>}, v^{<4>}$	$q^{<5>}, k^{<5>}, v^{<5>}$
$x^{<1>}$	$x^{<2>}$	$x^{<3>}$	$x^{<4>}$	$x^{<5>}$
Jane	visite	l'Afrique	en	septembre

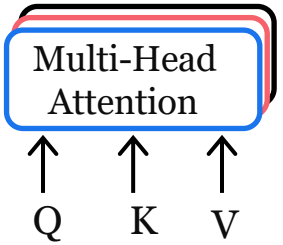
[Vaswani et al. 2017, Attention Is All You Need]

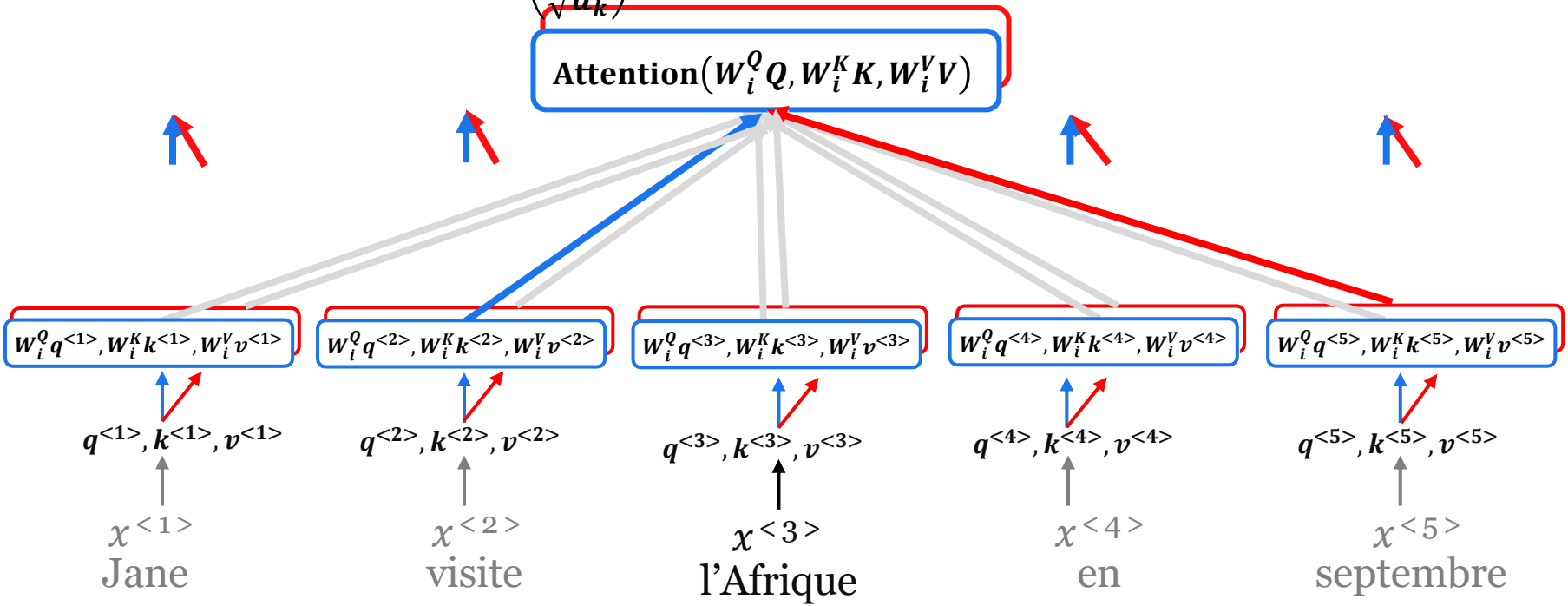
Multi-Head Attention

second head: W_2^Q, W_2^K, W_2^V **- When is sth happening in Africa?**

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

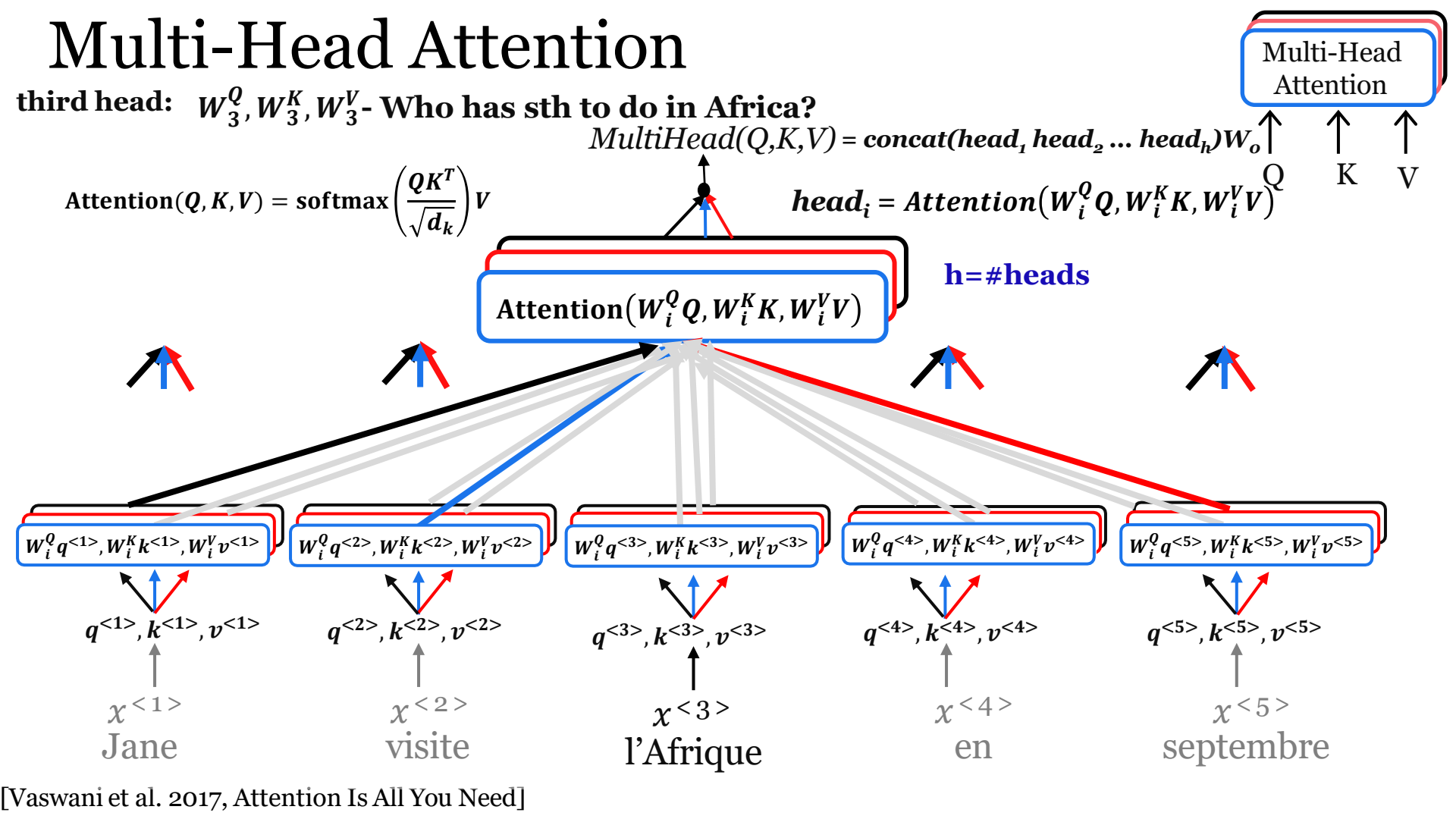
MultiHead(Q,K,V)





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

Multi-Head Attention



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

Multi-head attention Dimensions

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

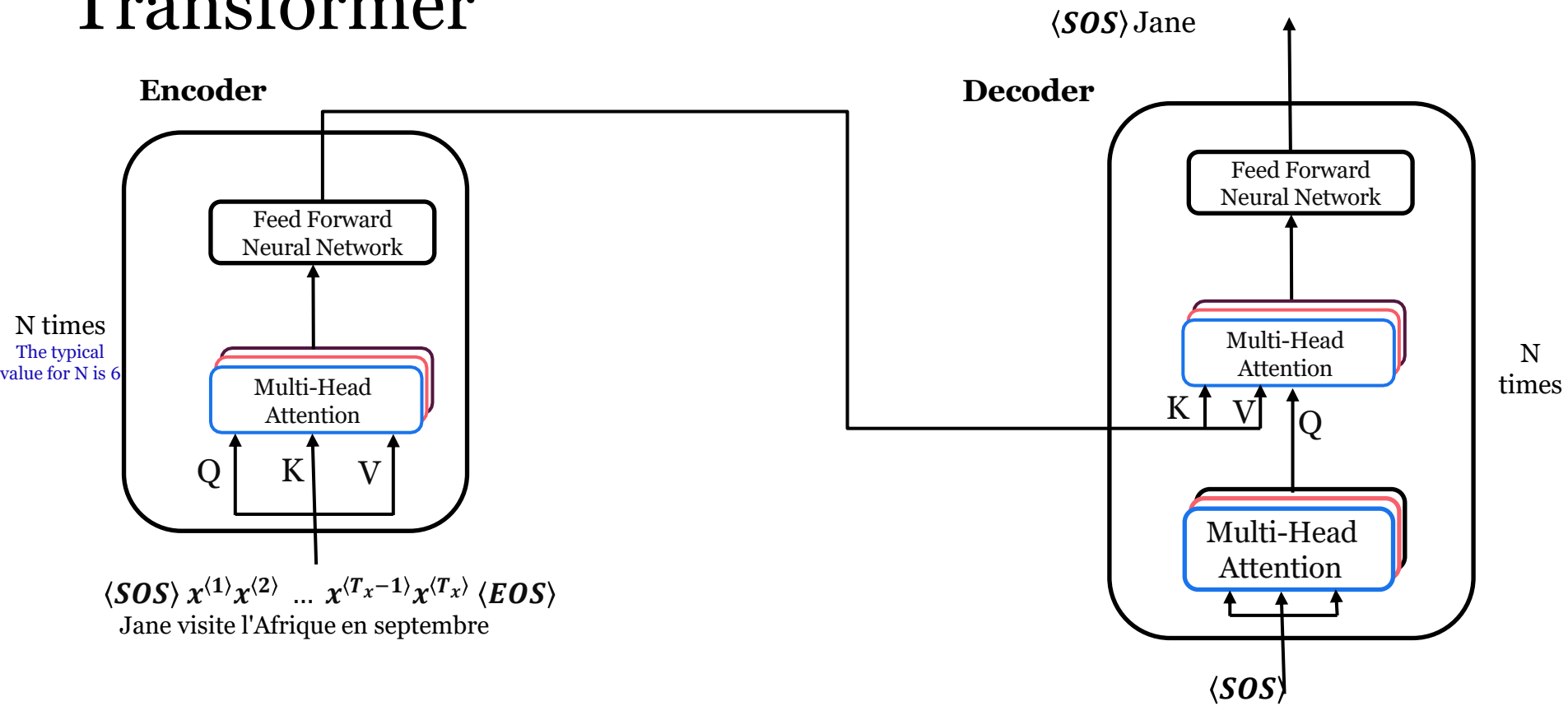
Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

In this work we employ $h = 8$ parallel attention layers, or heads. For each of these we use $d_k = d_v = d_{\text{model}}/h = 64$. Due to the reduced dimension of each head, the total computational cost is similar to that of single-head attention with full dimensionality.

The dimensionality of input and output is $d_{\text{model}} = 512$,

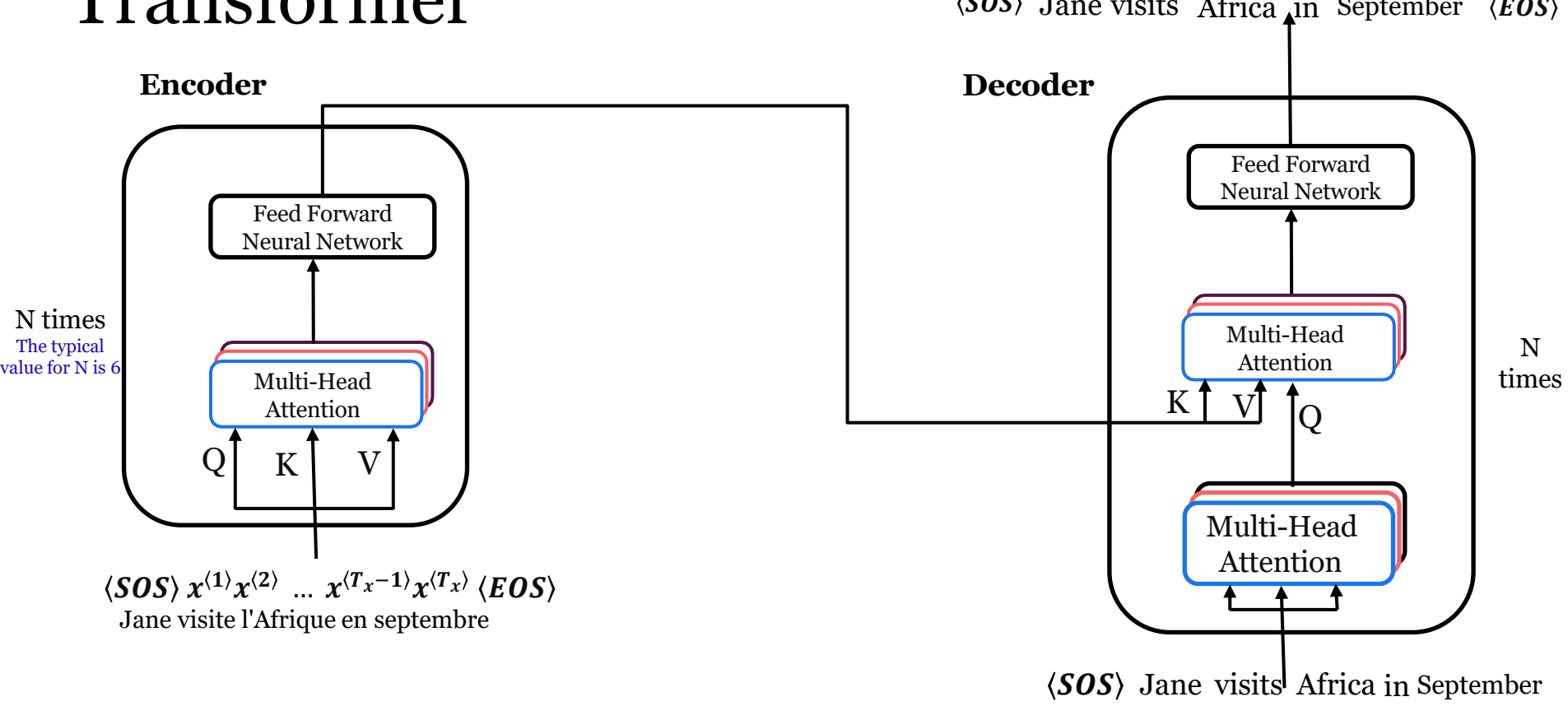
$$\textbf{Attention}(\boldsymbol{QW^Q}, \boldsymbol{KW^K}, \boldsymbol{VW^V}) = \textbf{softmax}\left(\frac{(\boldsymbol{QW^Q})(\boldsymbol{KW^K})^T}{\sqrt{\boldsymbol{d_k}}}\right)(\boldsymbol{VW^V}) \qquad Q, K, V \in R^{T \times d_{model}}$$

Transformer



- The translation will start with a start of sentence token(⟨***SOS***⟩) and so the start of sentence token gets fed into this multi-head attention block.
- Just this one token, ⟨***SOS***⟩ is used to compute Q, K and V for this multi-head attention block.

Transformer



Transformer Details: Positional Embedding

- Positional encoding is necessary to convey word positions in self-attention.
- Self-attention equations don't inherently indicate word positions.
- Position within a sentence holds significant importance.
- Sine and cosine equations are used for positional encoding.
- Encoding position helps capture meaningful context.
- Preserving positional information enhances sentence understanding.

Positional Encoding

- add a value in range [0,1] to the time steps in which 0 means the first word and 1 is the last time-step

Drawbacks:

- you can't figure out how many words are present within a specific range.
- time-step delta doesn't have consistent meaning across different sentences.

- Second idea is to assign a number to each time-step linearly. That is, the first word is given "1", the second word is given "2", and so on.

Drawbacks:

- The values could get quite large
- our model can face sentences longer than the ones in training.
- our model may not see any sample with one specific length which would hurt generalization of our model.

https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

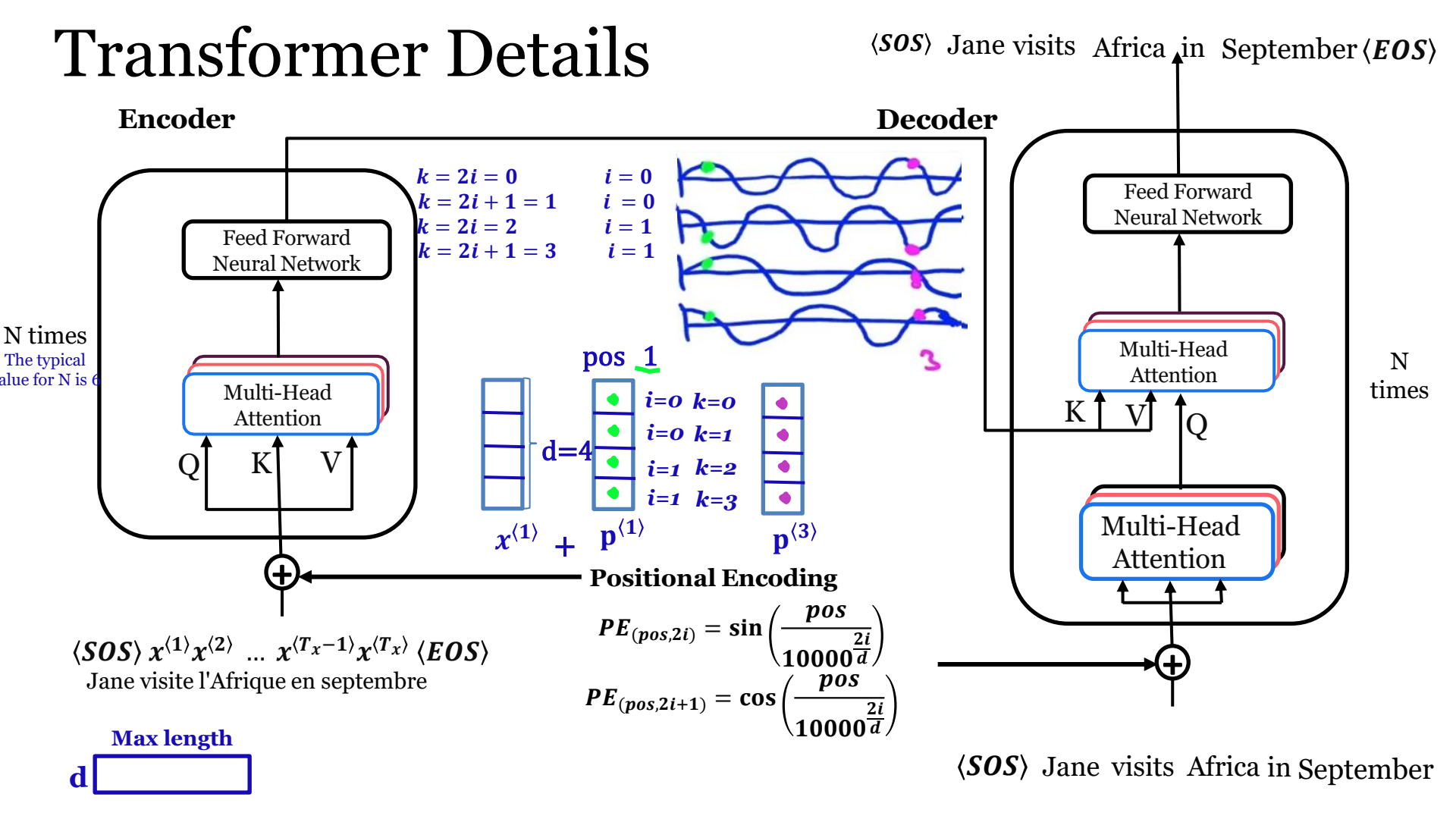
Positional Encoding

Ideally, the following criteria should be satisfied:

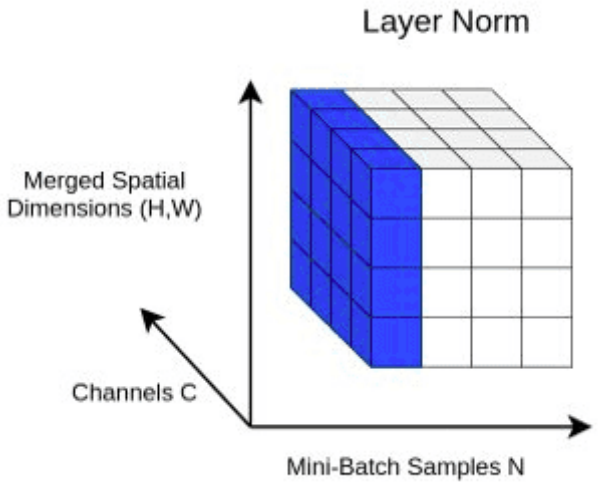
- It should output a unique encoding for each time-step (word's position in a sentence)
- Distance between any two time-steps should be consistent across sentences with different lengths.
- Our model should generalize to longer sentences without any efforts. Its values should be bounded.
- It must be deterministic

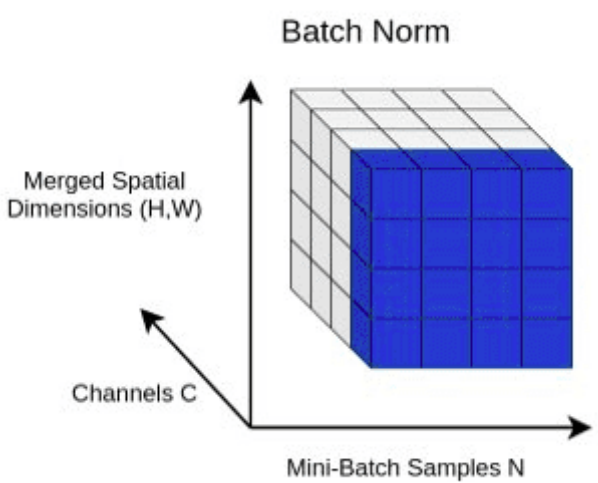
https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

Transformer Details



Batch norm vs Layer norm



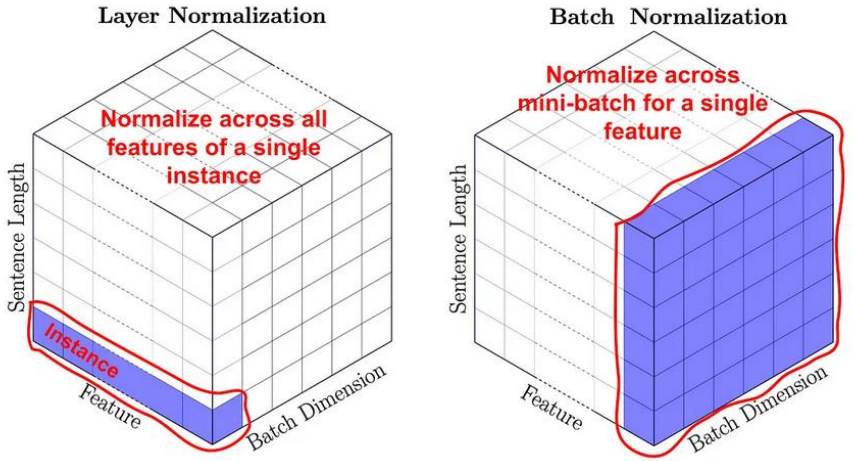


$$y = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \odot \gamma + \beta$$

Batch Normalization for RNN

Batch normalization drawbacks:

- Applying batch normalization to RNNs appears to require different statistics for different time-steps.
- BN cannot be applied to online learning tasks or to extremely large distributed models where the mini-batches have to be small.

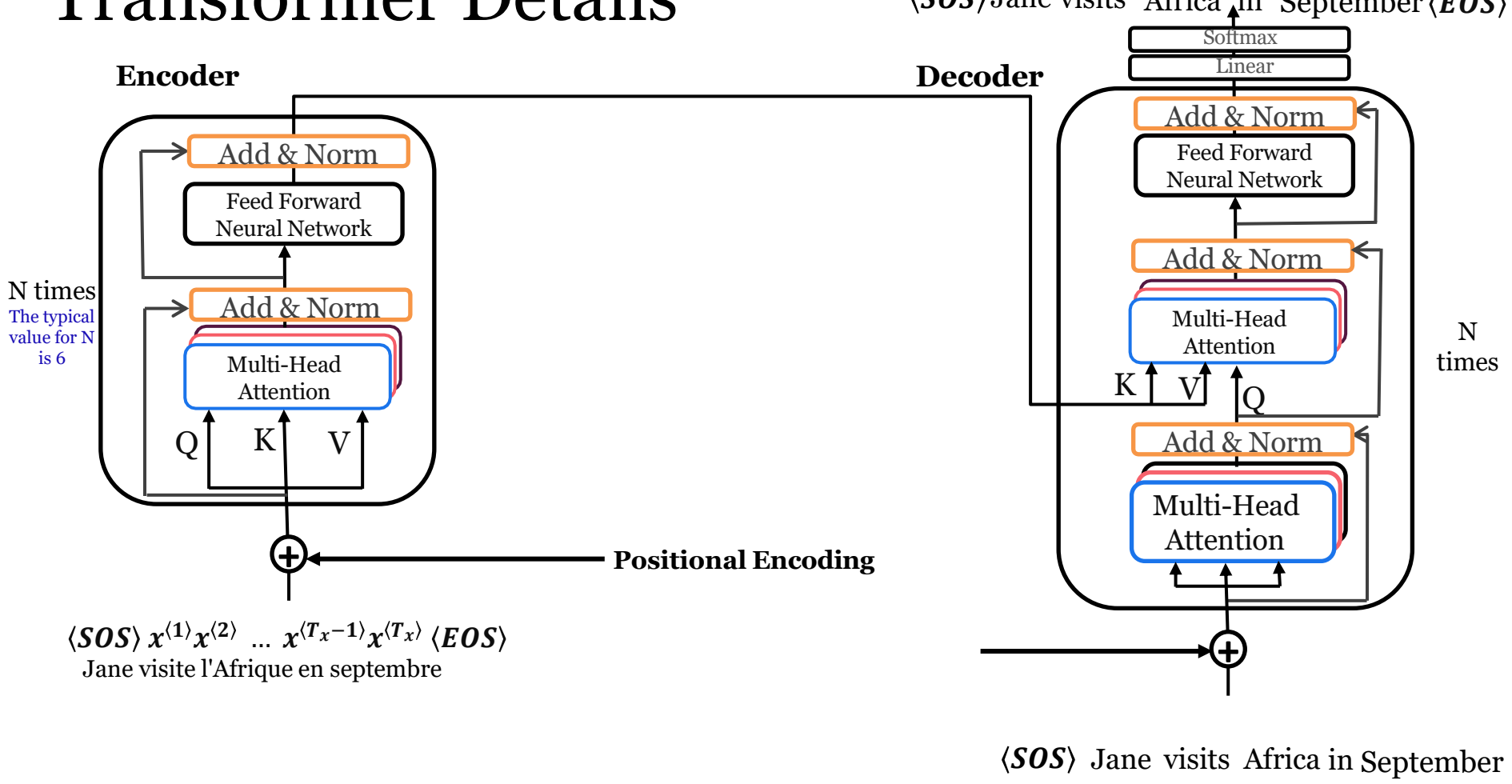


Transformer Details

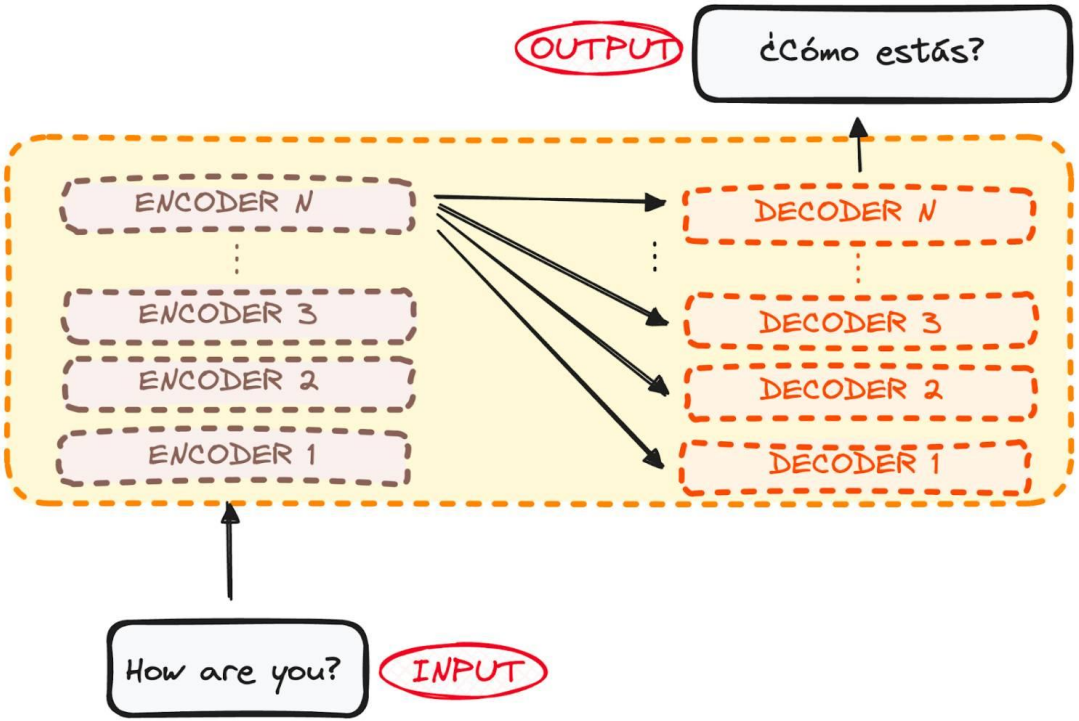
- The output of the encoding block contains contextual semantic embedding and positional encoding information

- Residual connections are used to pass positional information in the transformer architecture.
- “Add & Norm” is a layer similar to batch-norm, helping to speed up learning.
- This “Add & Norm” layer is repeated throughout the architecture.
- Positional encoding and linear layers are used in the transformer.
- A “SoftMax” layer predicts the next word in the decoder block, one word at a time.
- The transformer architecture is designed to handle sequential data.

Transformer Details



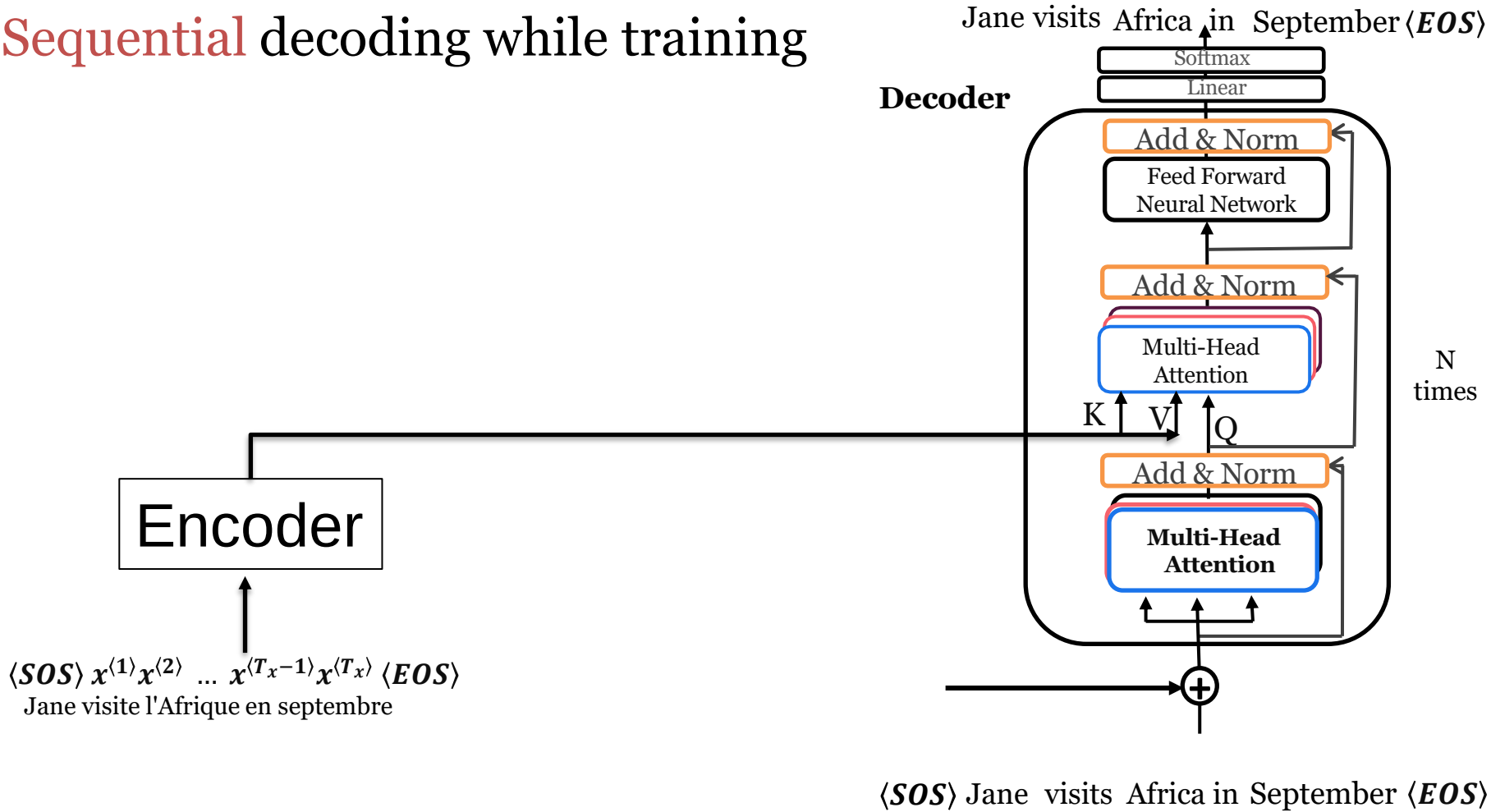
Transformer Details



Transformer Details: Masked Multi-Head Attention

- The **“Masked Multi-Head Attention”** is significant during the training process of the transformer network.
- It is used with datasets of correct French to English translations.
- The transformer network predicts one word at a time.
- Understanding how the transformer network is trained is important.
- “Masked Multi-head attention” enhances the transformer's ability to learn translation patterns.
- Masking is used during training to simulate prediction behavior.
- Masking is used during training to mimic test time or prediction scenarios.
- The blocked-out portion of the sentence helps the network predict the next word accurately.
- The aim is to see if the network can perform well given a perfect first part of the translation.

Sequential decoding while training



Parallel Decoding while training