

10/7/25.

Attention is all you need.

Problems with RNN

- 1) Slow computation
- 2) Vanishing or exploding gradients

$$x \rightarrow f(x,y) = x \cdot y \rightarrow g(z) = z^2$$

$$y \underbrace{\quad}_{0.5} \underbrace{\quad}_{0.5} \rightarrow 0.25$$

$$\frac{dg}{dx} = \frac{dg}{df} \times \frac{df}{dx} \quad (\text{Chain rule})$$

Computation chain is long, too much.

Either very big no. or very small no.  
 $> 1$  multiplied       $< 1$  multiplied

- 3) Difficulty in accessing info from long time ago.

Introducing the Transformer.

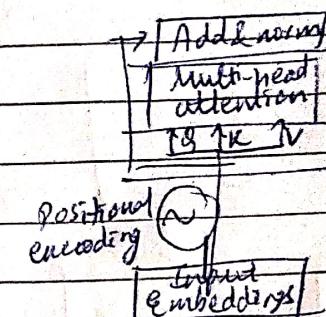
Encoder      Decoder

Notations

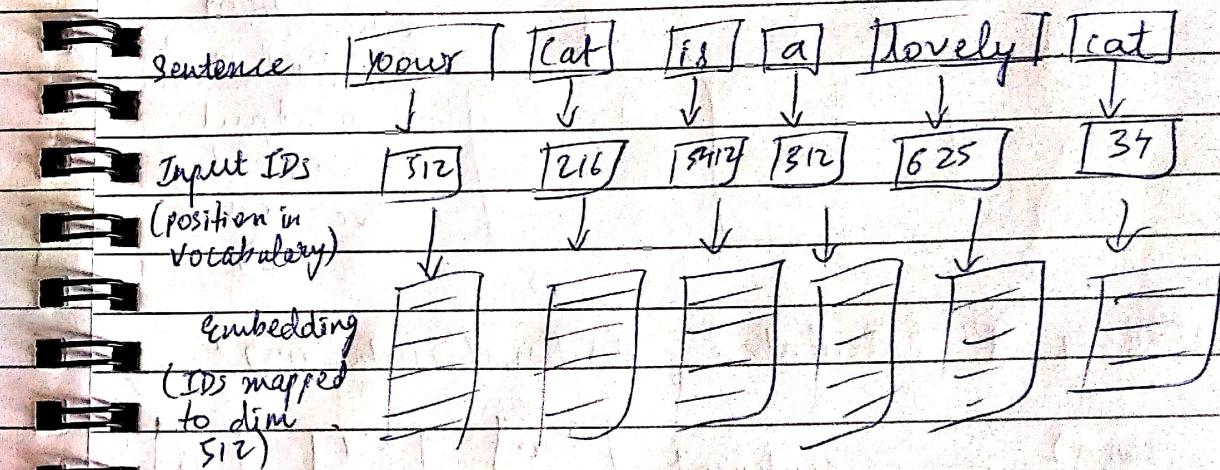
Input matrix sequence ( $d_{\text{model}}$ )

	(6, 512)	A B C D E F	A B C D E F
A	512	: : : : : :	A A A A A A
B	x	: : : : : :	B B B B B B
C		: : : : : :	C C C C C C
D		: : : : : :	D D D D D D
E		: : : : : :	E E E E E E
F		: : : : : :	F F F F F F

(512, 6)



What is an input embeddings?

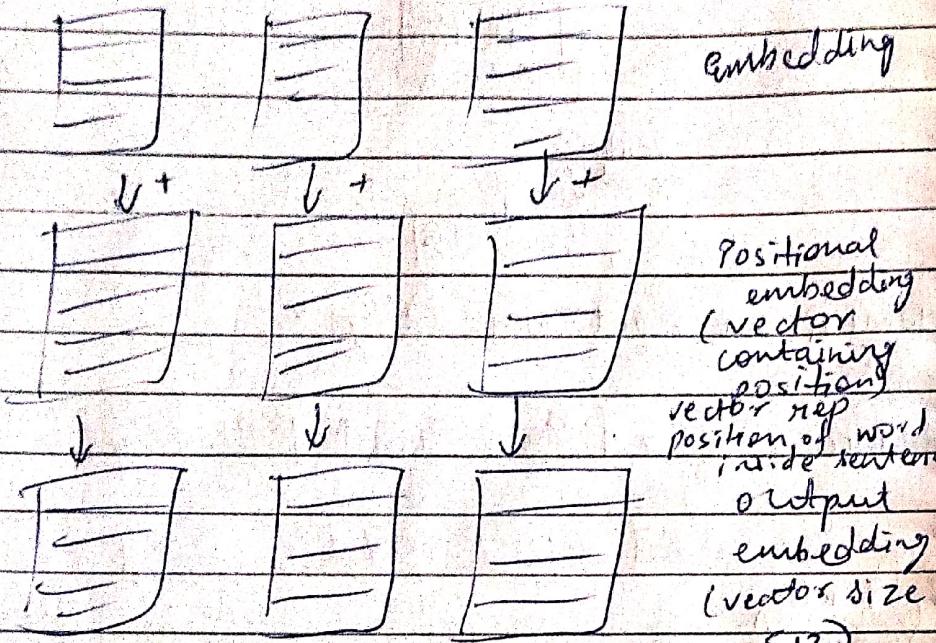


$$d_{\text{model}} = 512$$

What is positional encoding?

- We want each word to carry some information about its position in the sentence.
- Words appear close  $\rightarrow$  close model  
 $\dots \dots \dots$   
 distant  $\rightarrow$  distant model

- Positional encoding represent patterns for model to learn.



How is it calculated, positional encoding?

(even) sentence Your cat is

$$PE(pos, 2i) = \sin \frac{pos}{10000^{\frac{2i}{d_{model}}}}$$

$$PE(pos, 2i+1) = \cos \frac{pos}{10000^{\frac{2i+1}{d_{model}}}}$$

(odd)

Why trig func?

Why cos ( sin )

What is self-attention?

→ Self attention allows the model to relate words to each other

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

→ In this simple case, we consider the sequence length seq=6 and  $d_{model} + d_v = 512$ .

→ The matrices  $\mathbf{Q}, \mathbf{K}, \mathbf{V}$  are just the input sentence.

$$\begin{array}{c|c|c|c|c} \mathbf{Q} & \times & \mathbf{K}^T & & \text{softmax} = \\ (6, 512) & & (512, 6) & & \\ \hline & \sqrt{512} & & & (6, 6) \end{array}$$

sums up  
every value  
to 1

(scale it  
down)

$$\underbrace{\begin{bmatrix} Q \\ (seq, d_{model}) \end{bmatrix}}_{(seq, d_{model})} \times \underbrace{\begin{bmatrix} W_Q \\ (d_{model}, d_{model}) \end{bmatrix}}_{(d_{model}, d_{model})} = \underbrace{\begin{bmatrix} Q' \\ (seq, d_{model}) \end{bmatrix}}_{(seq, d_{model})}$$

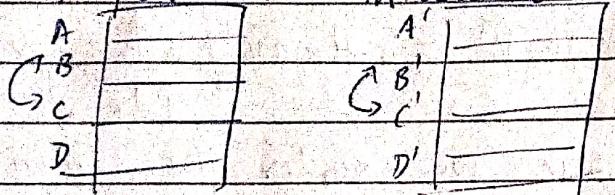
$$X \begin{bmatrix} \sqrt{6} \\ (6, 512) \end{bmatrix} = \begin{bmatrix} 6 \\ (6, 512) \end{bmatrix} \xrightarrow{512} \text{Attention}$$

$$\text{Input} \begin{bmatrix} K \\ (seq, d_{model}) \end{bmatrix} \times \begin{bmatrix} W_K \\ (512, d_{model}) \end{bmatrix} = \begin{bmatrix} K' \\ (512, 512) \end{bmatrix}$$

$(6, 6)$

Self - Attention in detail

1) It is permutation invariant.



$$(6, 512) \xrightarrow{d_{model}} \begin{bmatrix} V \\ (seq, d_{model}) \end{bmatrix} \times \begin{bmatrix} W_V \\ (512, 512) \end{bmatrix} = \begin{bmatrix} V' \\ (512, 512) \end{bmatrix}$$

2) Doesn't require any parameter.

3) We expect value along the diagonal to be maximum. (dot prod of each value with  $i^{th} \text{ col}$ )

$$\xrightarrow{d_K \quad h=4} \begin{bmatrix} Q_1 & Q_2 & Q_3 & Q_4 \end{bmatrix}_{\text{seq}} \quad \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right)V$$

4) If we don't want two words to interact, then substitute  $-\infty$  in their value.

Softmax replaces  $-\infty$  to zero (0).

$$\xrightarrow{V_1 \quad V_2 \quad V_3 \quad V_4} \begin{bmatrix} K_1 & K_2 & K_3 & K_4 \end{bmatrix}_{\text{head}} \quad \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

$$d_V = d_K = \frac{d_{\text{model}}}{h}$$

$$\text{Multihead}(Q, K, V) = \text{concat}(\text{head}_1, \dots, \text{head}_h) \times W_O$$

What is Multi-Head Attention?

seq = sequence length = 6

$d_{\text{model}}$  = size of embedding vector. = 512

$h$  = no. of heads

$$d_F = d_V = d_{\text{model}} / h$$

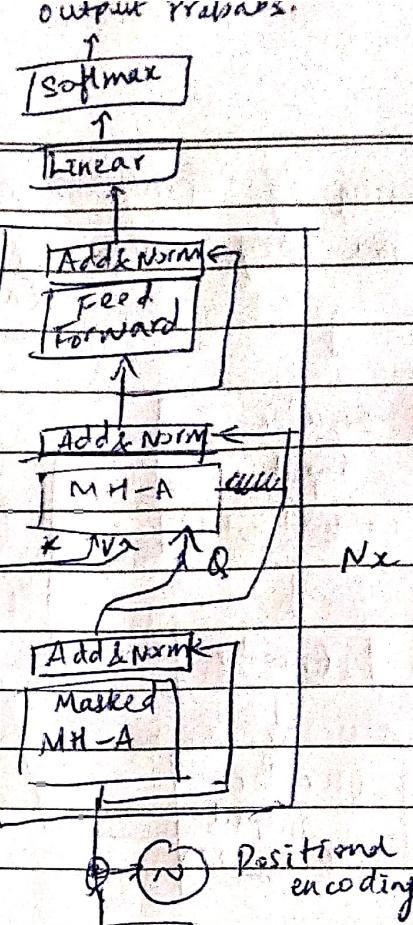
$$\xrightarrow{d_K = d_{\text{model}} / h} \begin{bmatrix} H_1 & H_2 & H_3 & H_4 \end{bmatrix}_{\text{seq}} \xrightarrow{d_V} \begin{bmatrix} (seq, d_V) \rightarrow H \\ (seq, h \cdot d_V) \end{bmatrix}_{(seq, h \cdot d_V)}$$

$$H \begin{pmatrix} \text{Seq}, h^d \end{pmatrix} \times W^0 \begin{pmatrix} h^d, d_{\text{model}} \end{pmatrix} = MH-A \begin{pmatrix} \text{Seq}, d_{\text{model}} \end{pmatrix}$$

→ Why Query, Keys and Values?

Keys & Values don't product query

Decoder.



→ Add & norm → What is layer normalization?

	item1	item2	item3
Batch of 3 items			
mean $\mu_1$	$\mu_2$	$\mu_3$	

variance	$\sigma_1^2$	$\sigma_2^2$	$\sigma_3^2$
----------	--------------	--------------	--------------

$$\hat{x}_j = x_j - \mu_j$$

$$\frac{\hat{x}_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

What is masked MH-A?

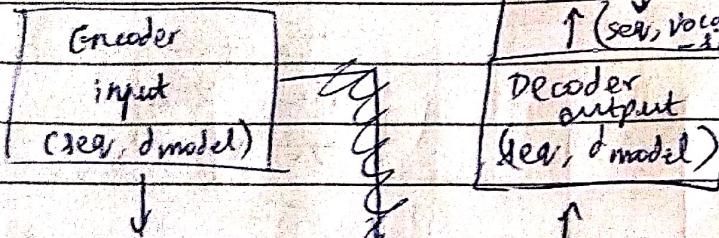
Our goal is to make the model causal: it means output at a certain position can only depend on the words on the prior positions. The model must not be able to see future words.

Inference & Training of Transformer Model  $\rightarrow$  Training

<SOS> I love you very much <EOS>

Start of sentence

↓  
End of sentence  
linear  
(seq, dmodel)



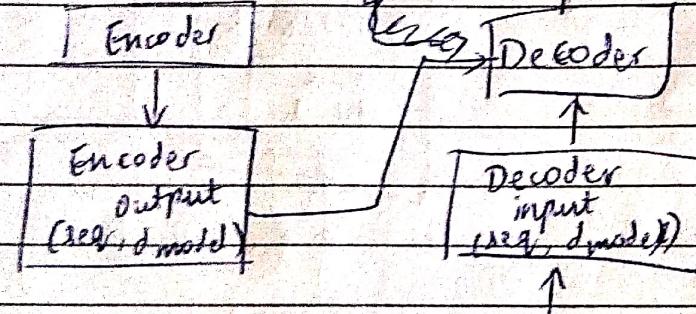
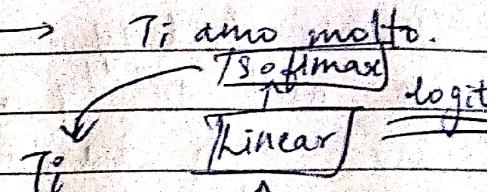
Time step = 1

It all happens in 1 time step!

### INFERENCE

I love you very much  $\rightarrow$  Ti amo molto.

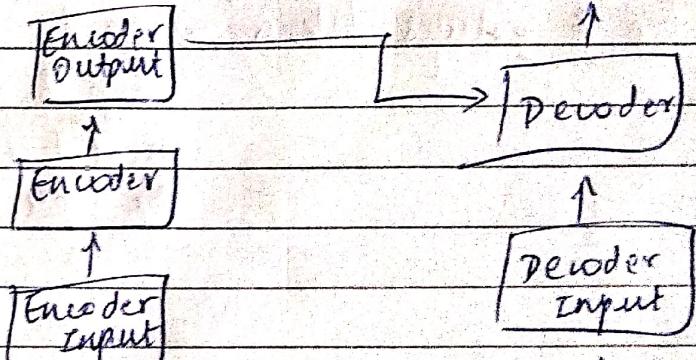
At time  
step = 1



<SOS> Ti amo molto

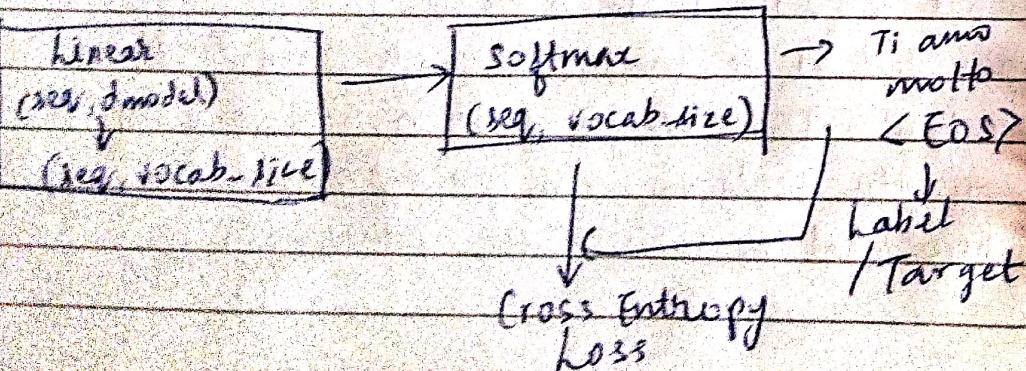
We prepend <SOS> token at the beginning.

That's why the paper says that the decoder input is shifted right



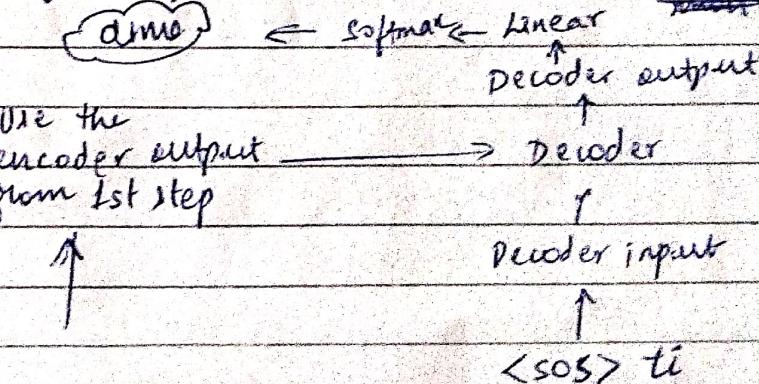
<SOS> I love you very much <EOS>

<SOS> Ti amo molto



At time  
step = 2

Use the  
encoder output  
from 1st step



<SOS> ti

Time step = 3      nmt-to.

Time step = 4       $\langle \text{EOS} \rangle$ .

### Inference strategy

→ We used greedy, at every step, the word with maximum softmax value.

(does not perform very well)

→ Beam Search is better.

↳ top  $B$  values taken → each of the top  $B$  values, next best words are predicted.