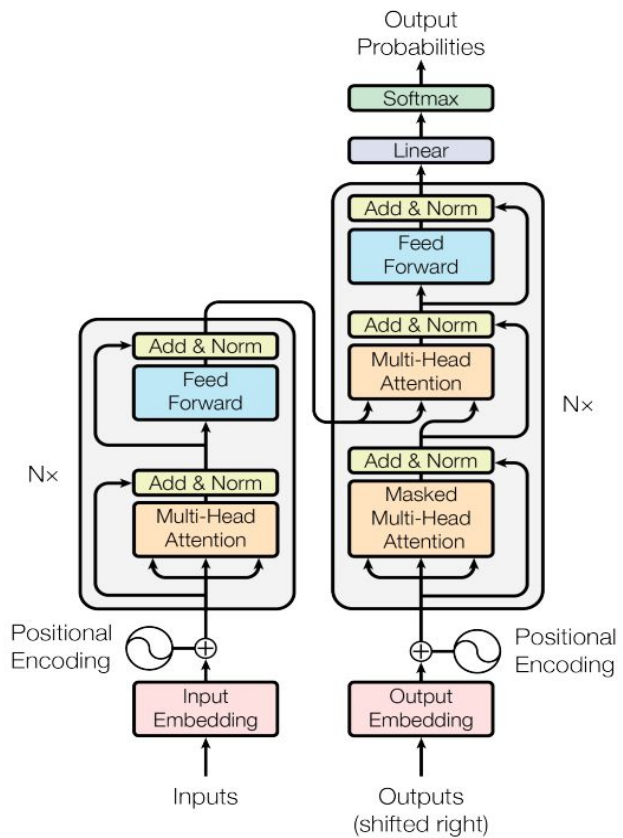


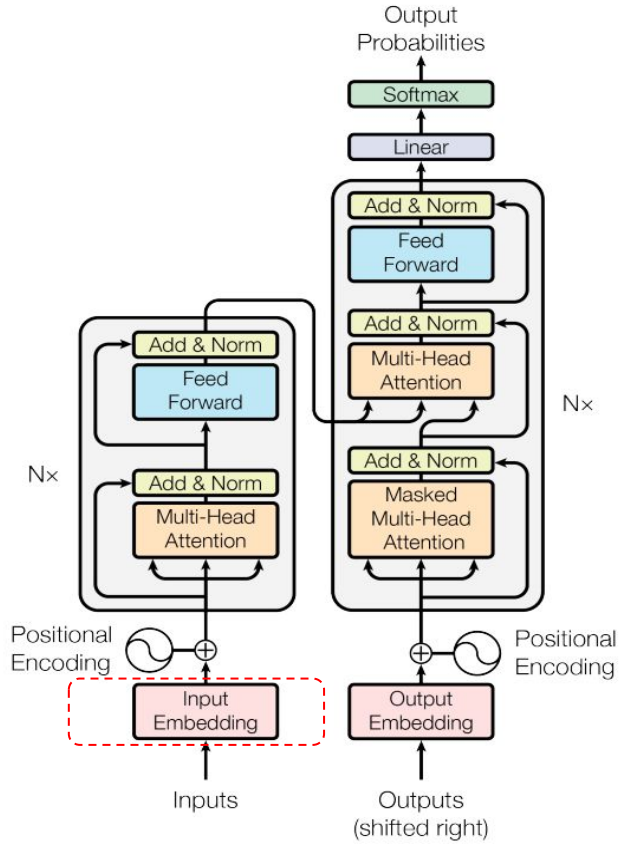
Transformers - Architecture

1. Introduction
2. The Architecture of the Transformer
 - 2.1. Input Embedding
 - 2.2. Positional Encoding
 - 2.3. Multi- Headed Attention
 - 2.4. Self-attention
 - 2.5. Layer Normalisation
 - 2.6. Point-wise feed forward

Transformers



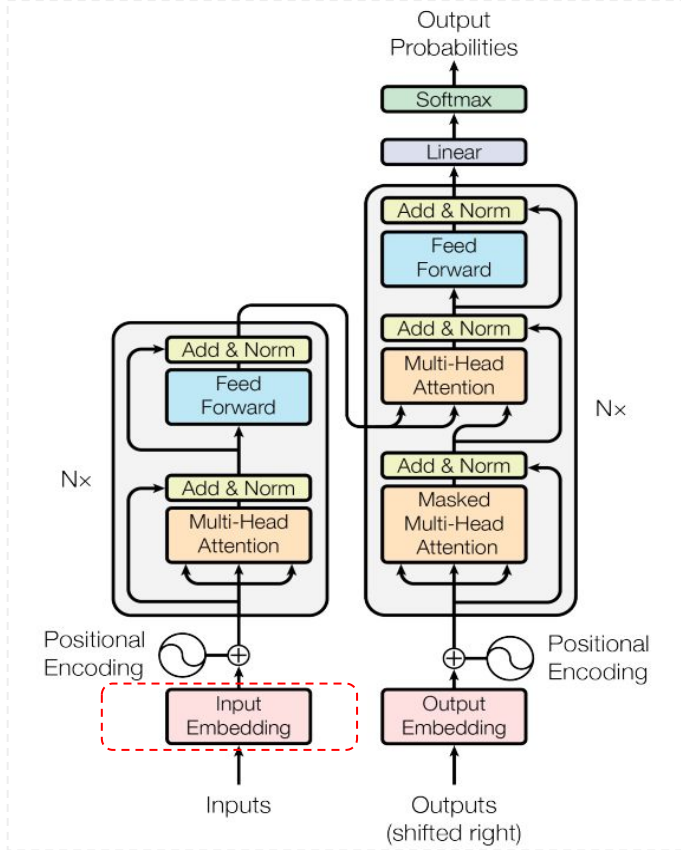
Encoder



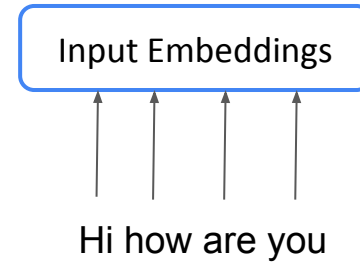
1. Input Embedding

Hi how are you

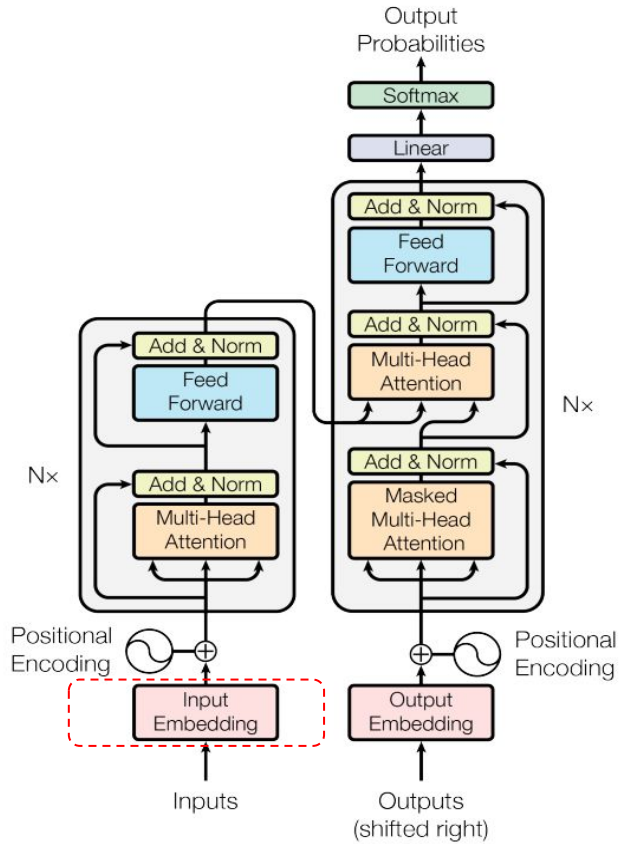
Encoder



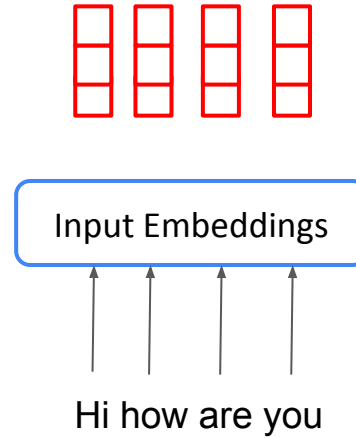
1. Input Embedding



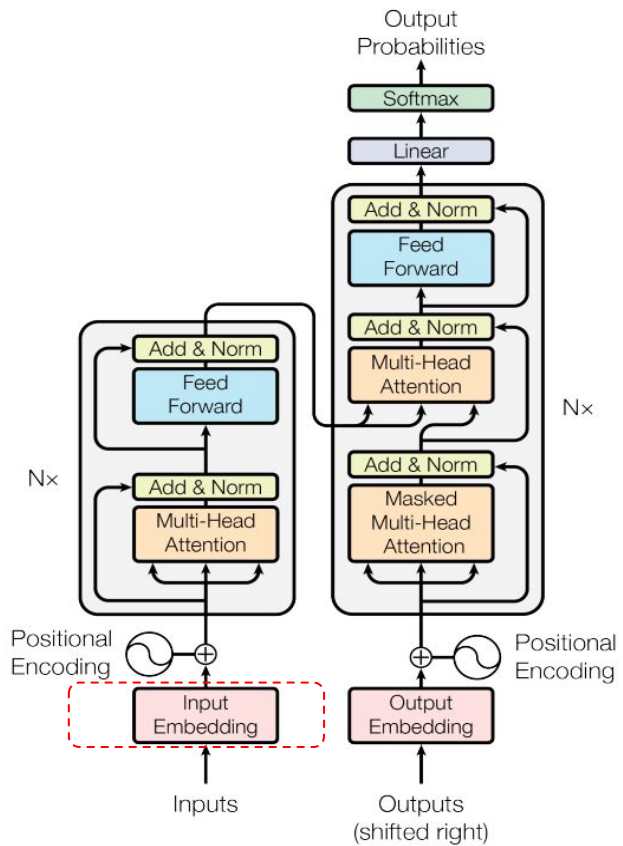
Encoder



1. Input Embedding

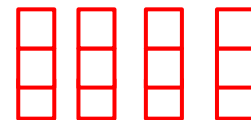


Encoder



1. Input Embedding

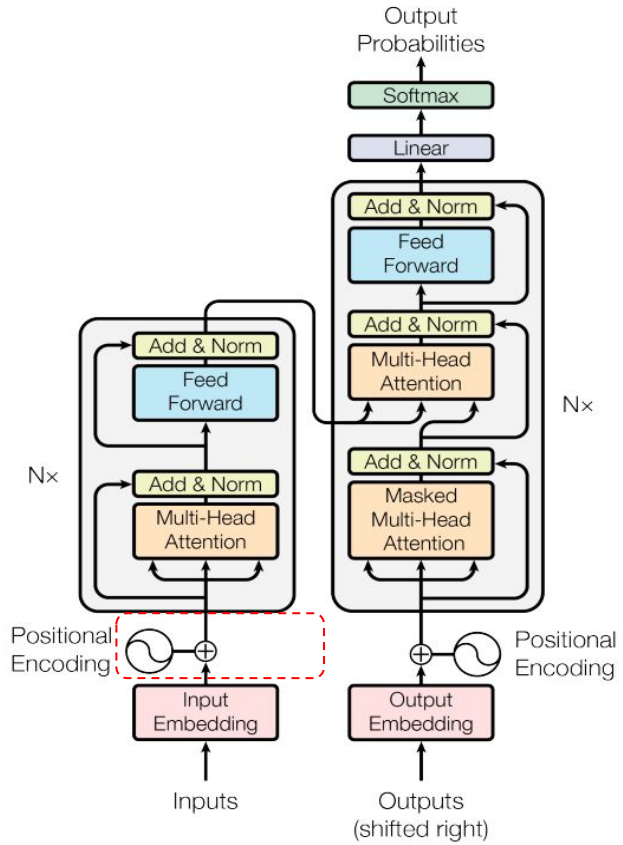
$$H_i = \begin{bmatrix} 0.1 \\ 0.8 \\ 0.4 \end{bmatrix}$$



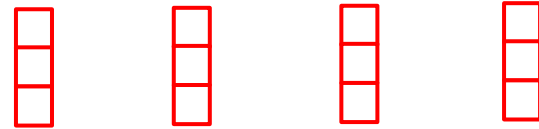
Input Embeddings

Hi how are you

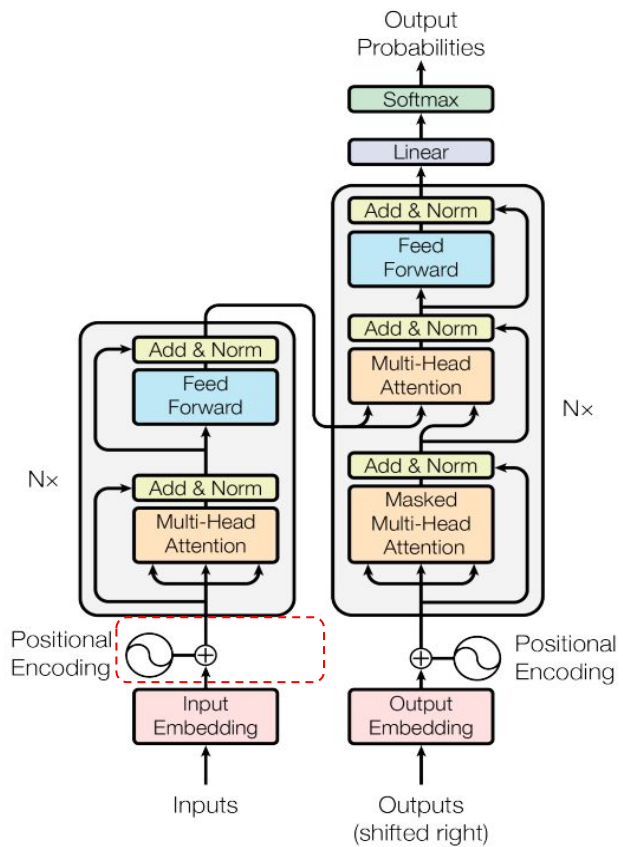
Encoder



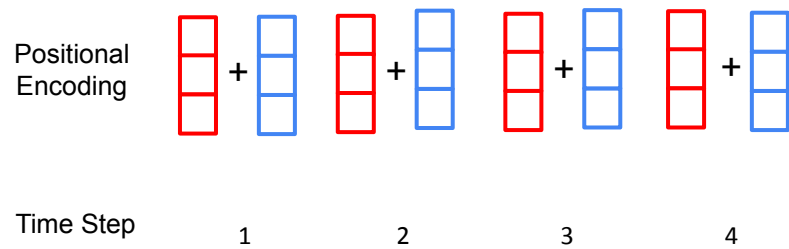
2. Positional Encoding



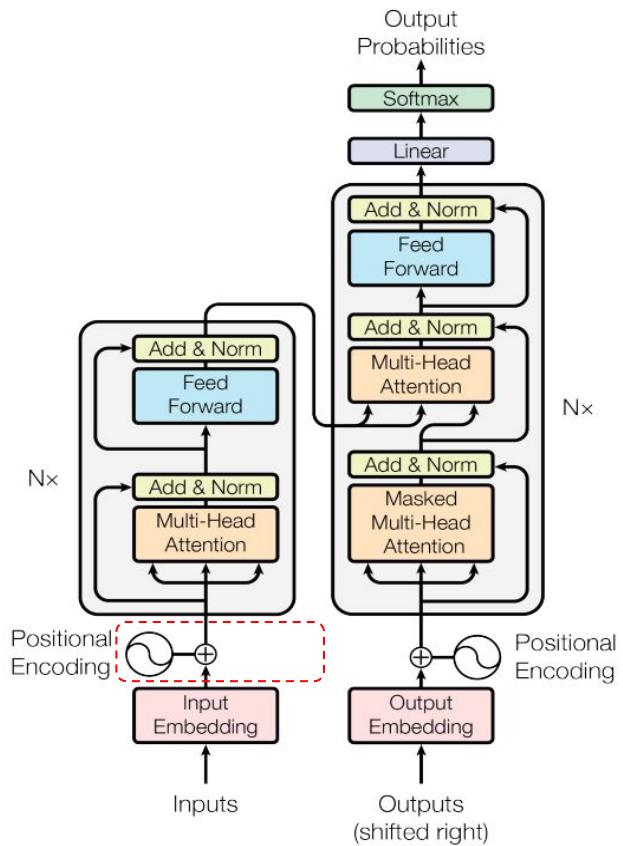
Transformers



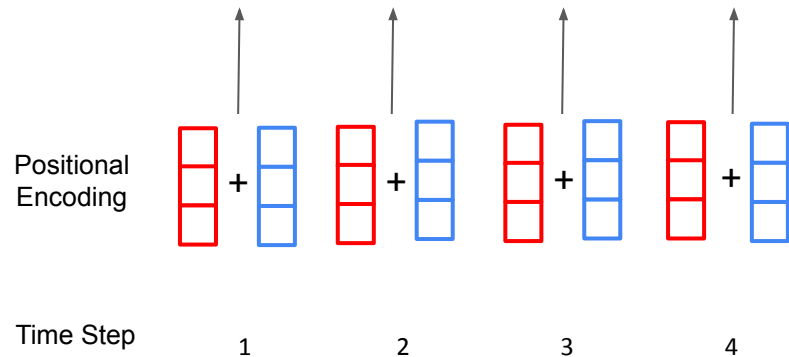
2. Positional Encoding



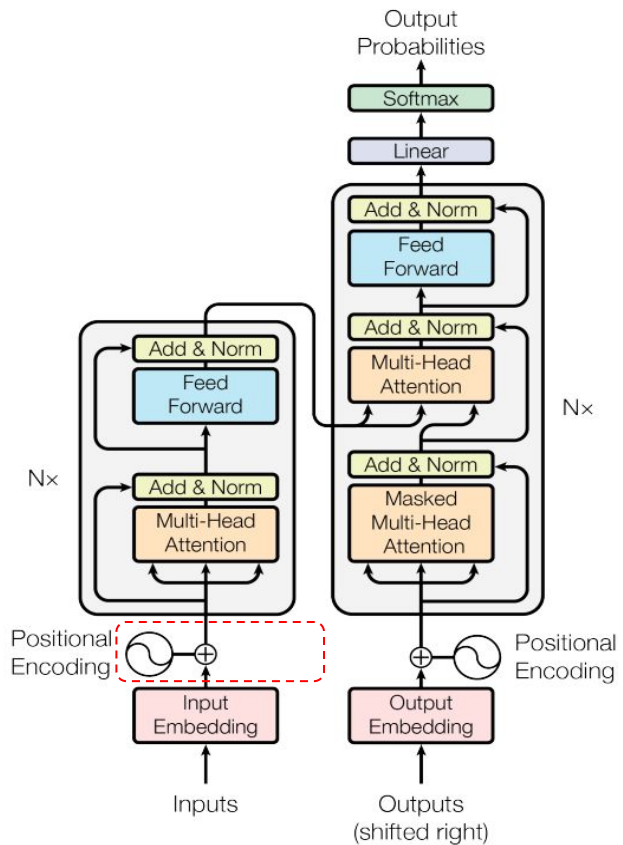
Transformers



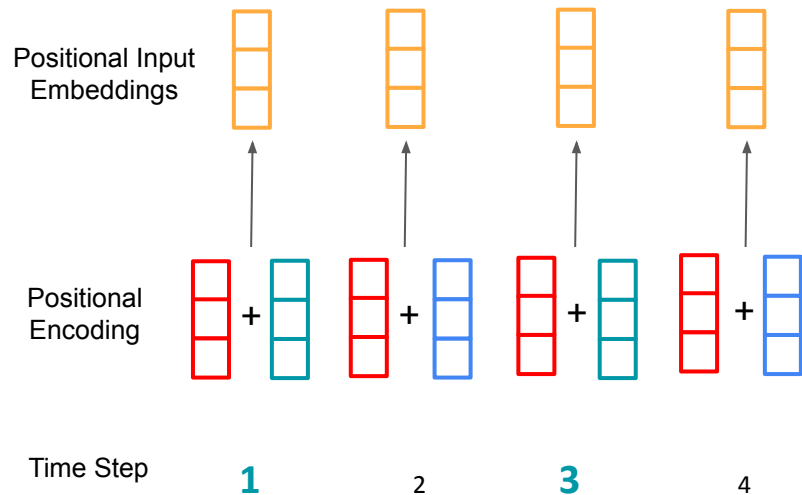
2. Positional Encoding



Encoder

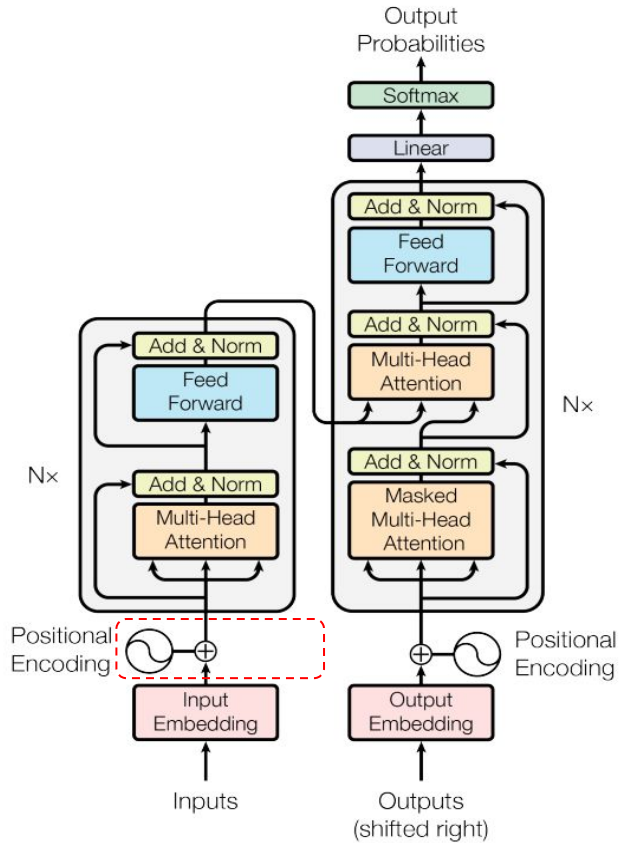


2. Positional Encoding

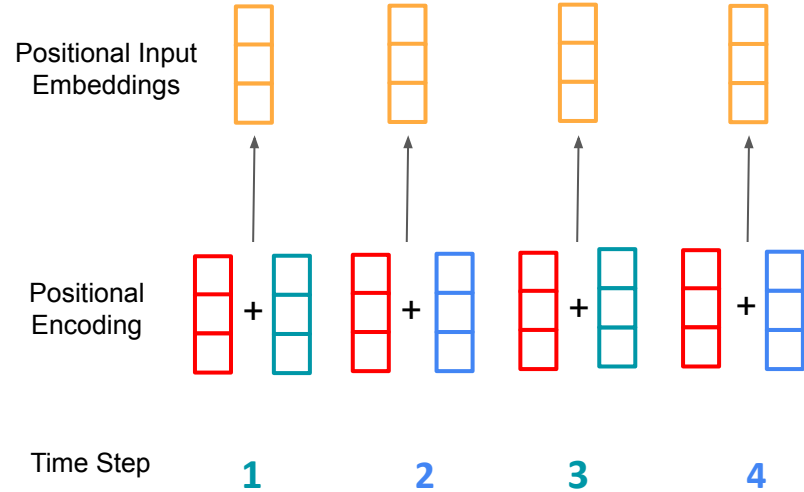


$$PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{model}})$$

Encoder



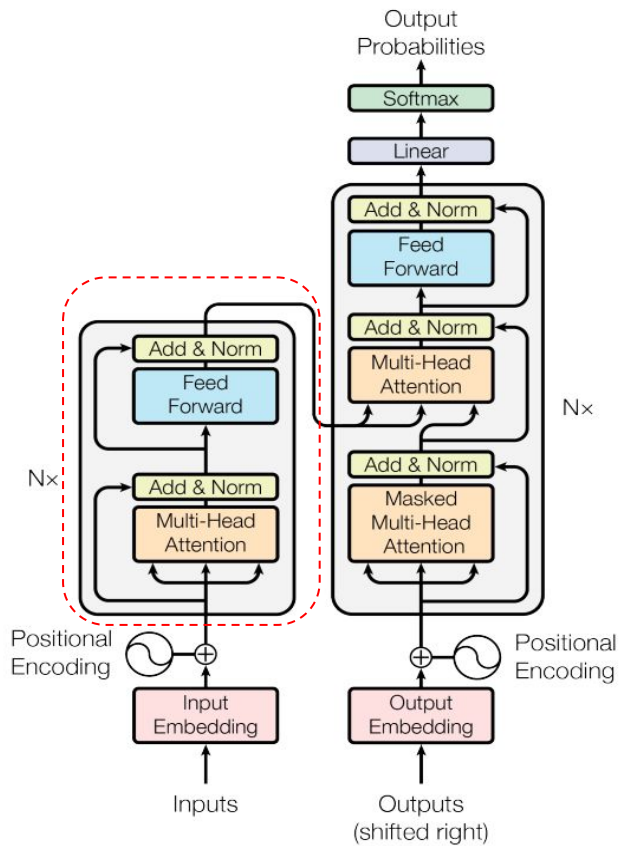
2. Positional Encoding



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

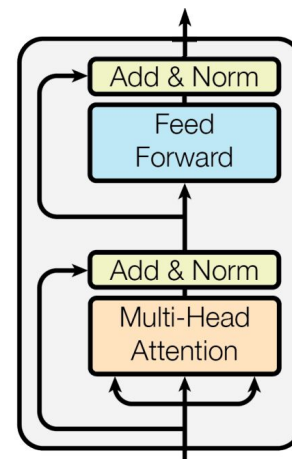
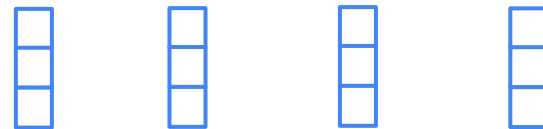
$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$

Encoder



2. Encoder Layer

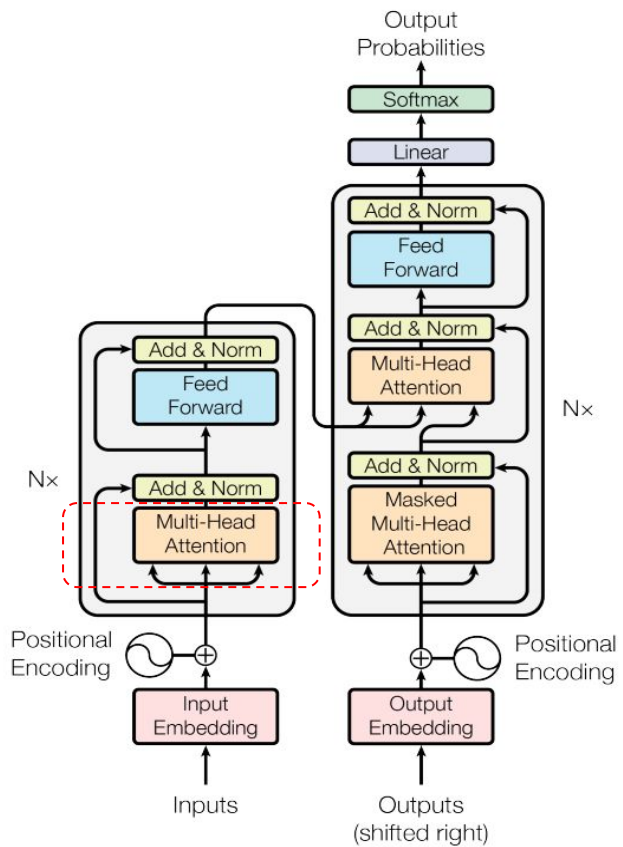
Encoder input Representation



Positional input Embeddings



Encoder

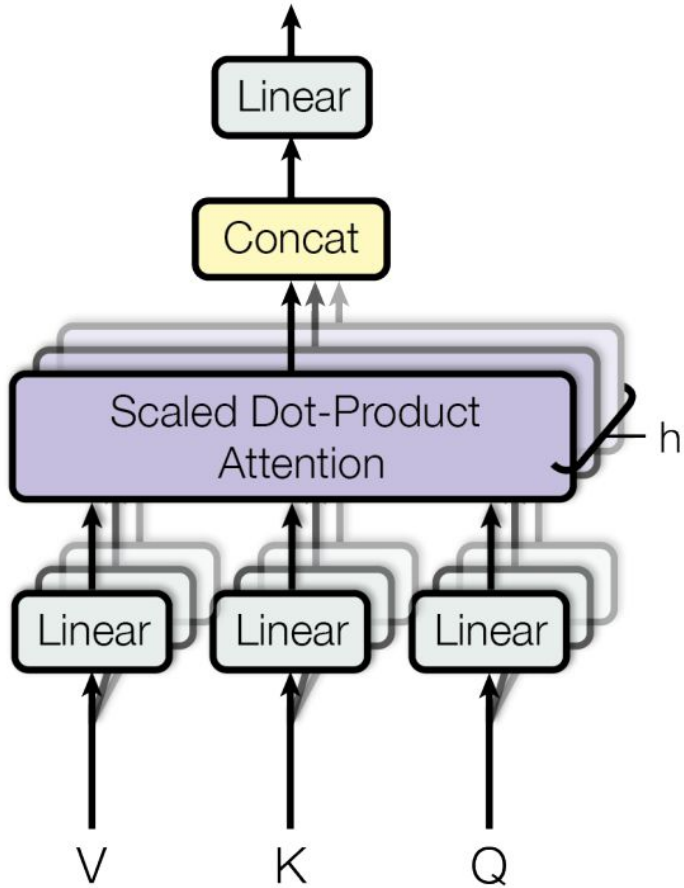


3. Multi-Headed Attention

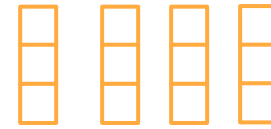


Hi how are you

Encoder

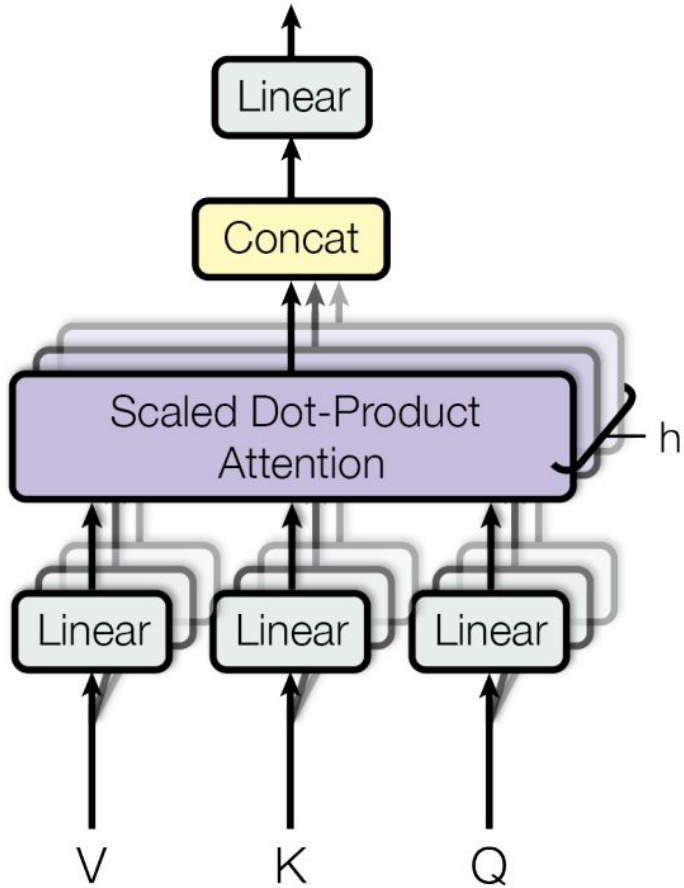


3. Multi-Headed Attention

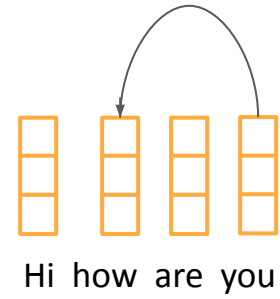


Hi how are you

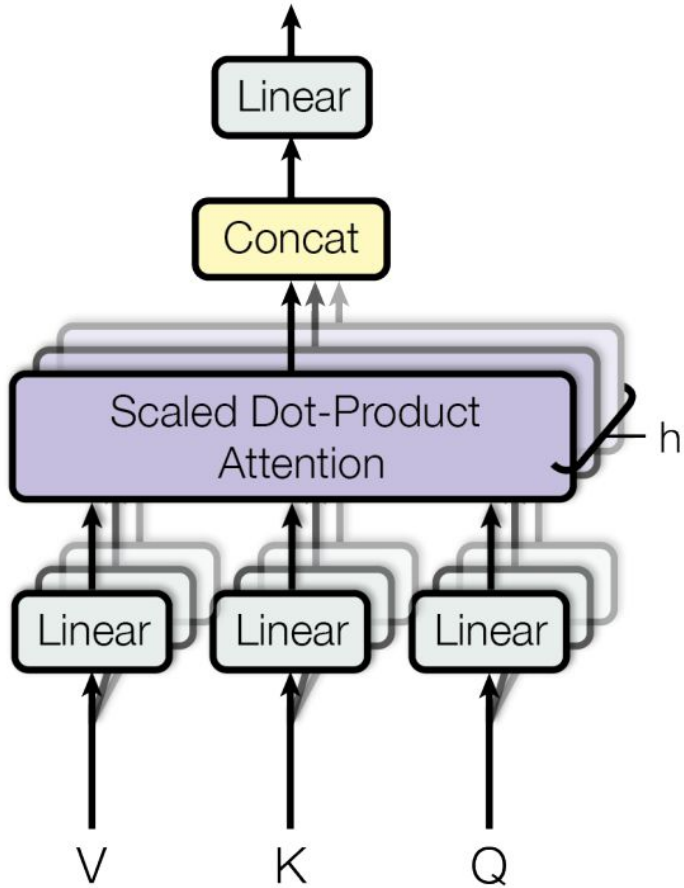
Encoder



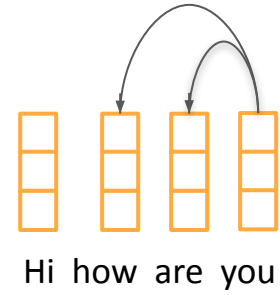
3. Multi-Headed Attention



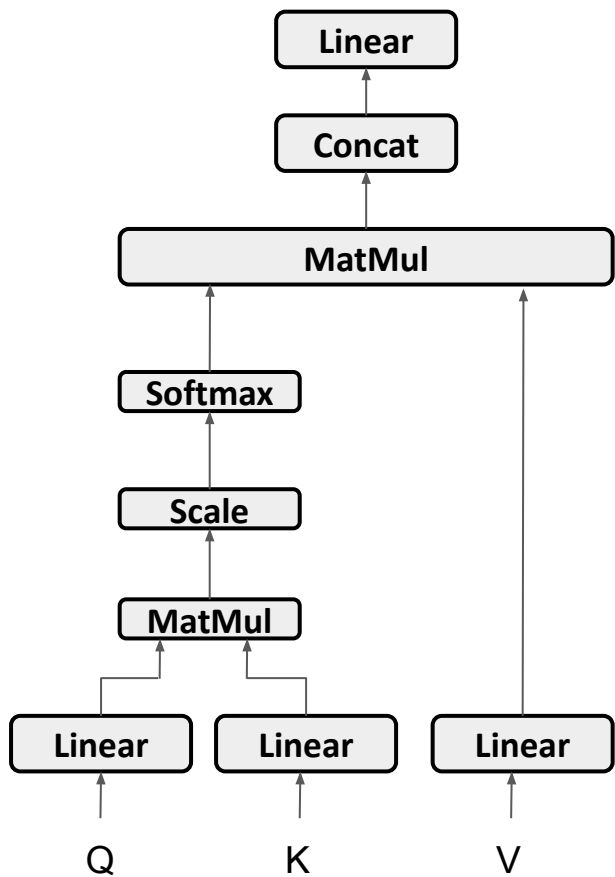
Encoder



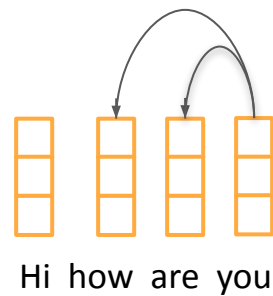
3. Multi-Headed Attention



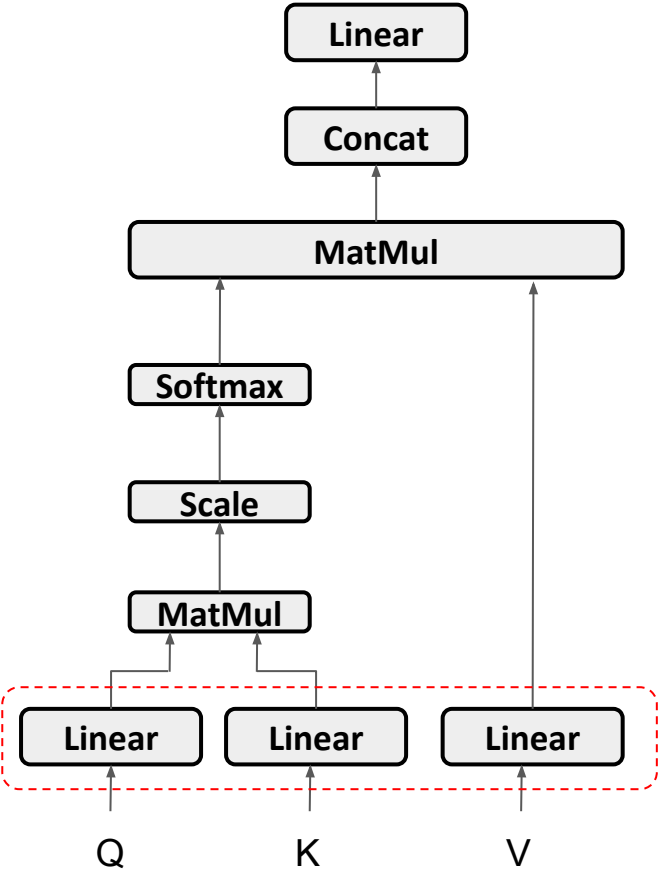
Encoder



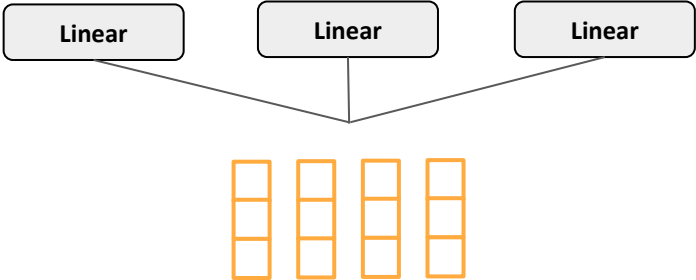
3. Multi-Headed Attention



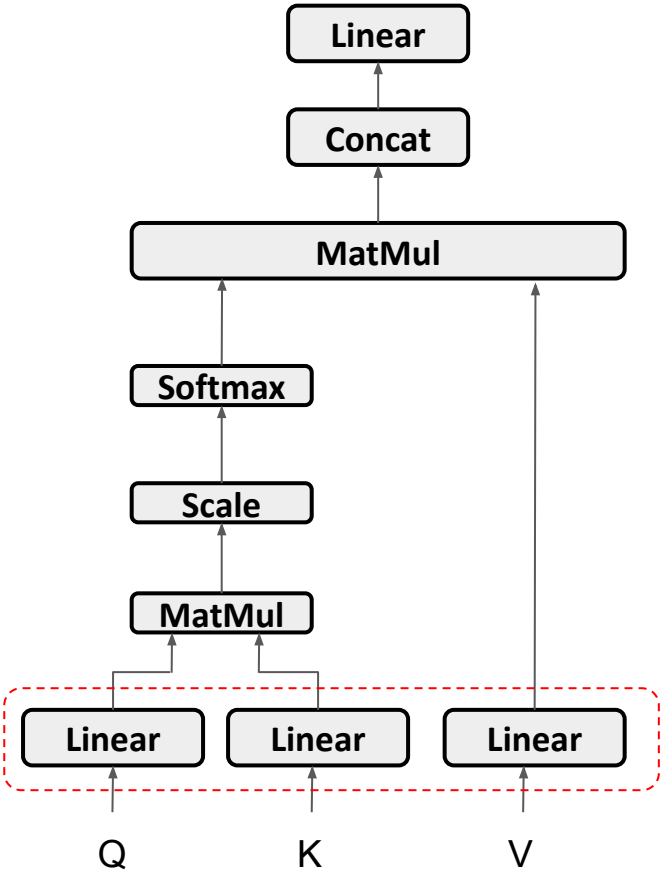
Encoder



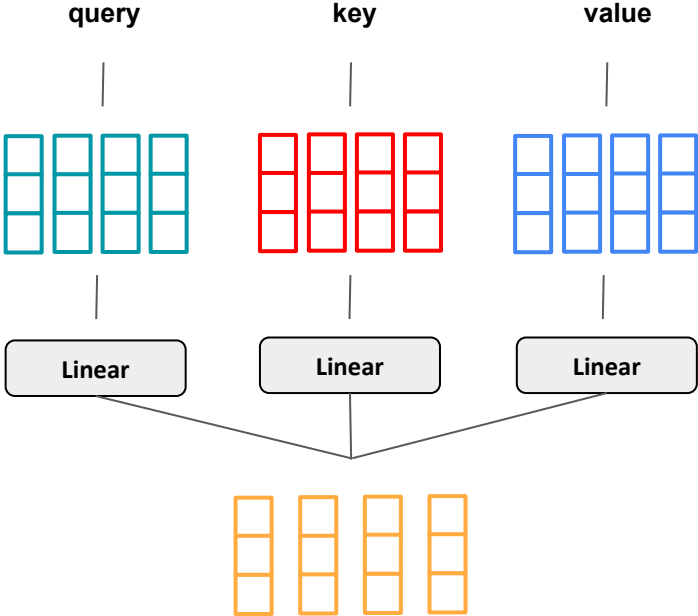
3. Multi-Headed Attention



Transformers

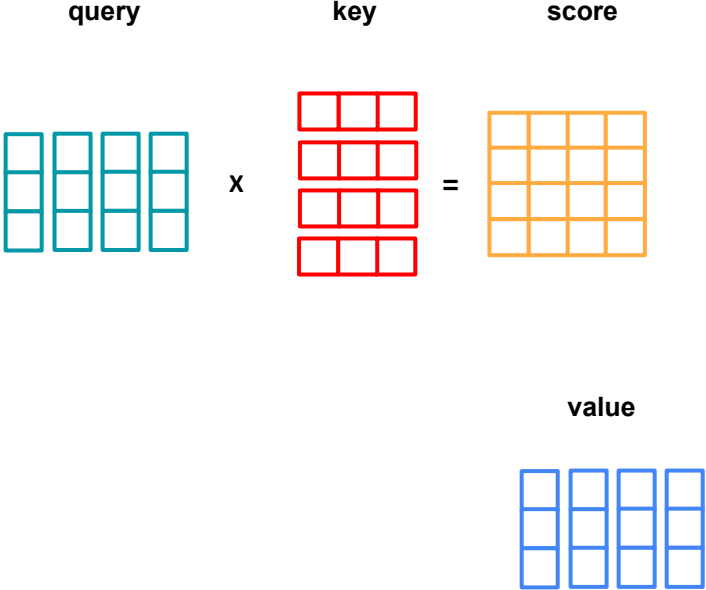
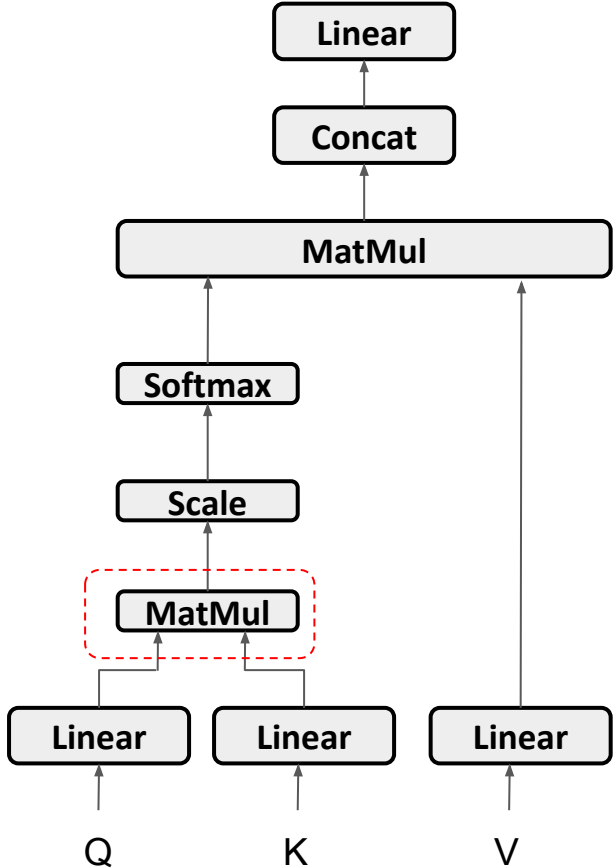


3. Multi-Headed Attention

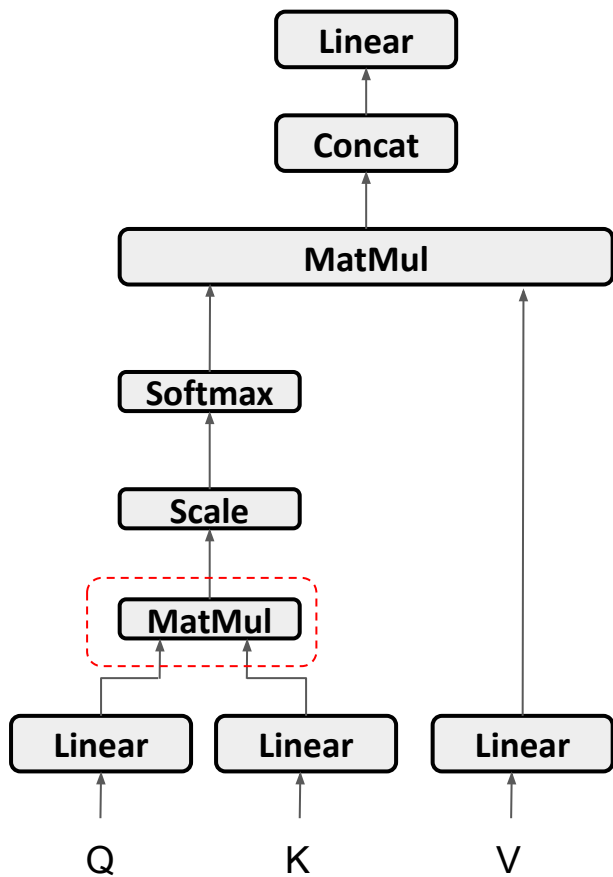


Transformers

3. Multi-Headed Attention



Encoder

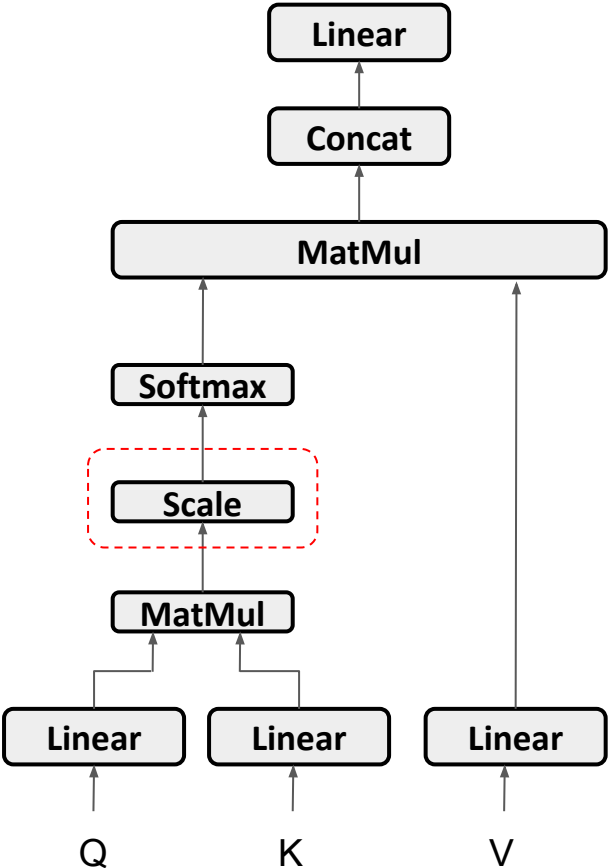


3. Multi-Headed Attention

	Hi	how	are	you
Hi	96	25	8	10
how	25	87	29	65
are	8	29	89	52
you	10	65	52	90

Transformers

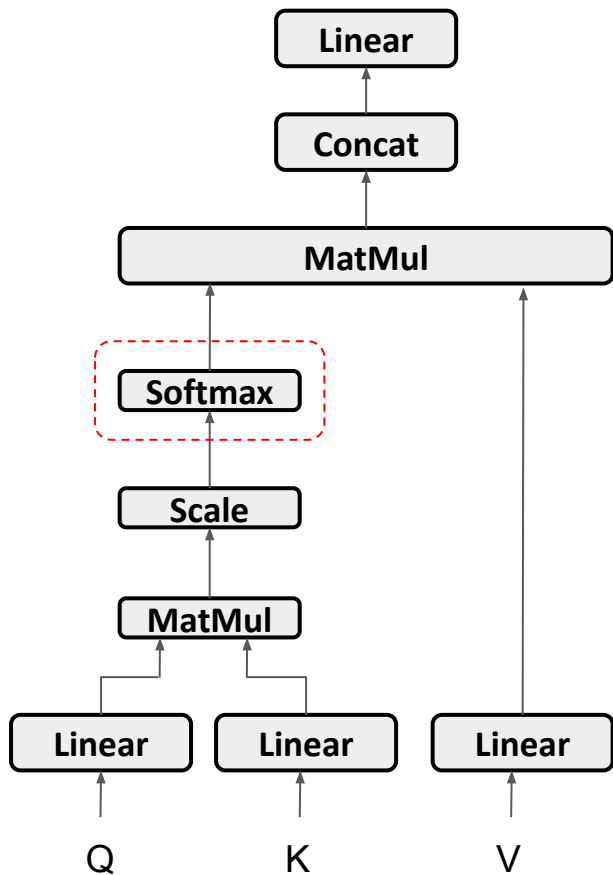
- 3. Multi- Headed Attention
 - 3.1 Self-Attention



The equation shows a 4x4 orange grid representing the output of the Q and K MatMul operation. This grid is divided by the square root of the dimensionality of the keys, $\sqrt{d_k}$. The result is a 4x4 teal grid labeled "Scaled scores".

Encoder

3. Multi-Headed Attention

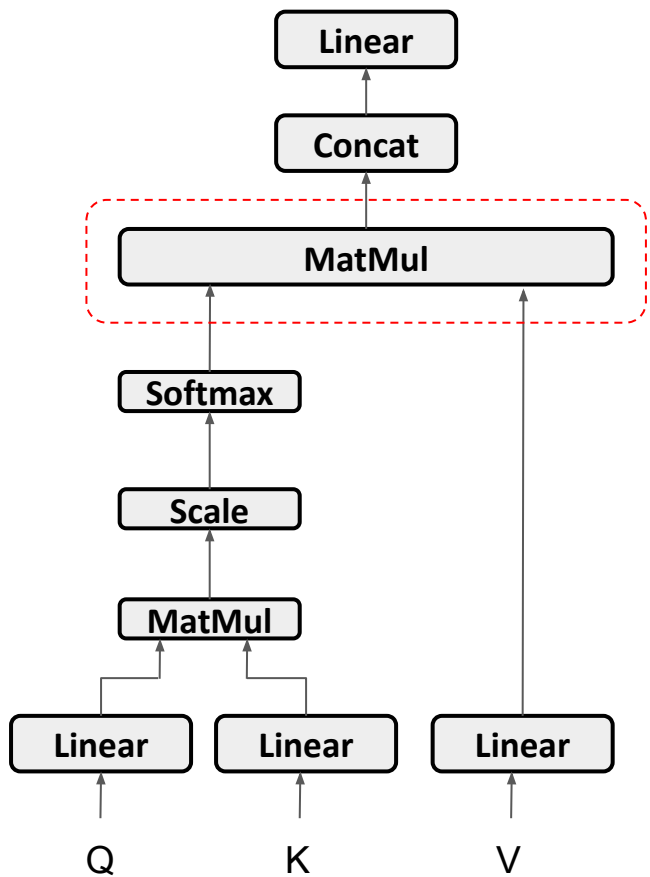


$$\text{Softmax}\left(\begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}\right) =$$

attention weights

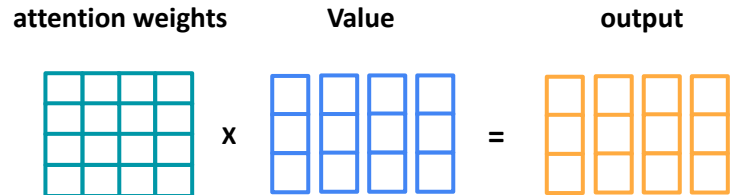
	Hi	how	are	you
Hi	0.7	0.1	0.1	0.1
how	0.2	0.5	0.1	0.2
are	0.1	0.2	0.6	0.1
you	0.1	0.2	0.4	0.3

Encoder

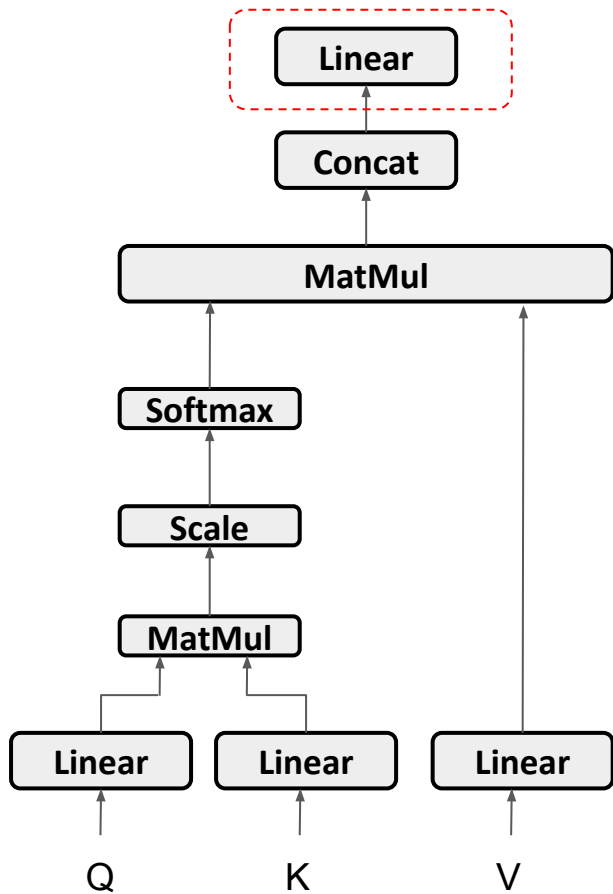


3. Multi-Headed Attention

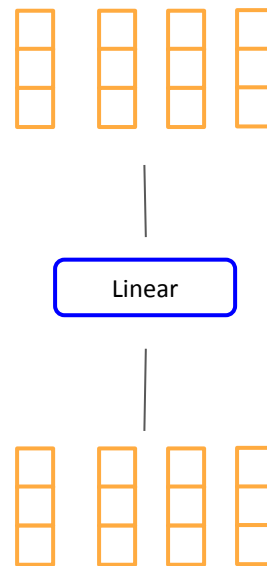
3.1 Self-Attention



Encoder

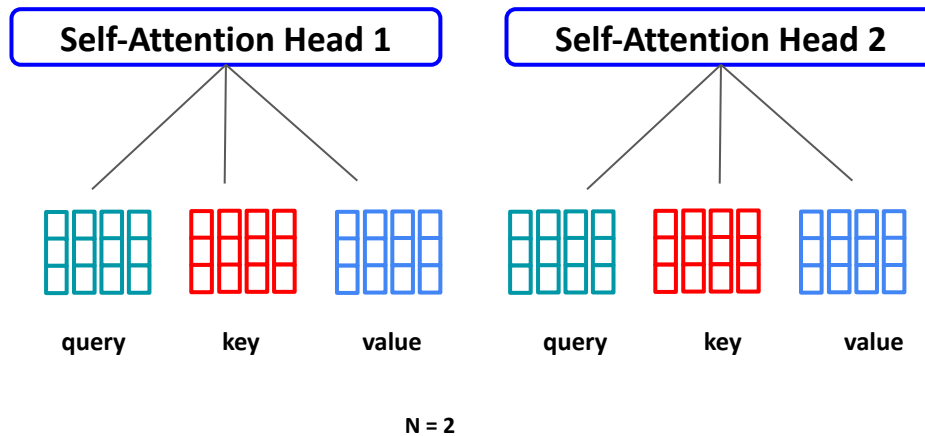
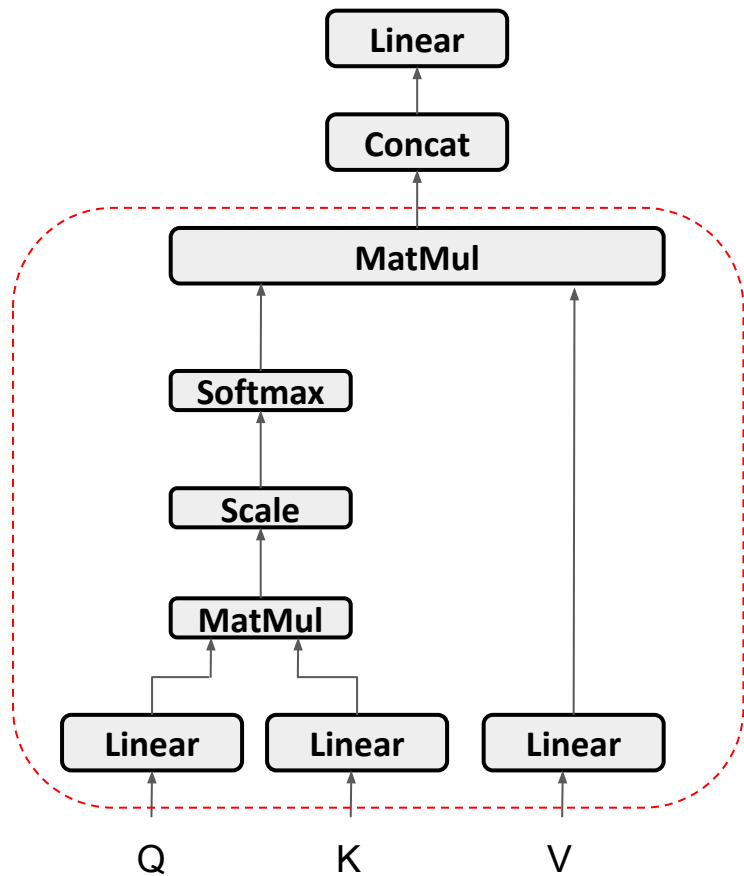


3. Multi-Headed Attention



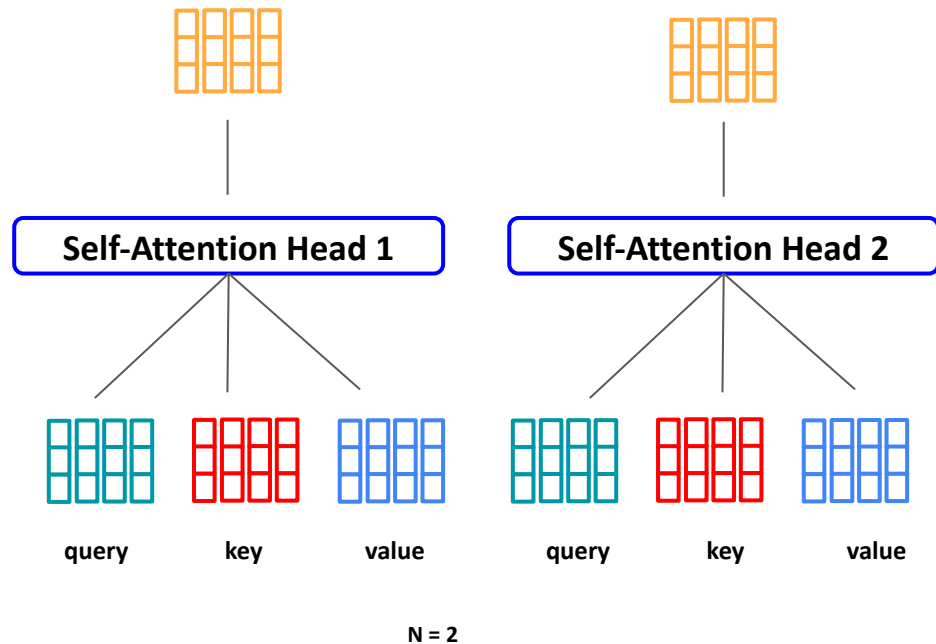
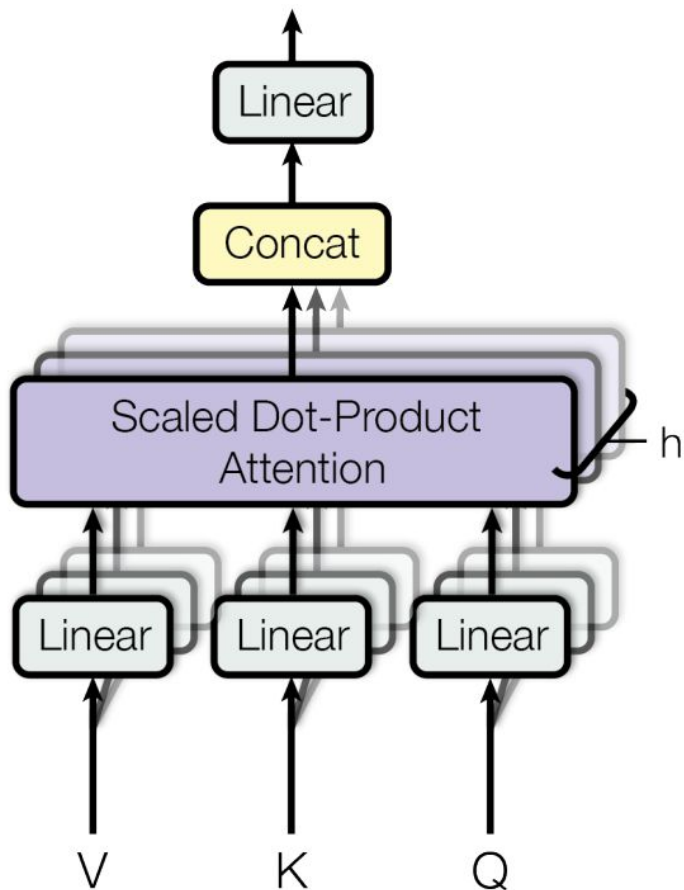
Encoder

3. Multi-Headed Attention

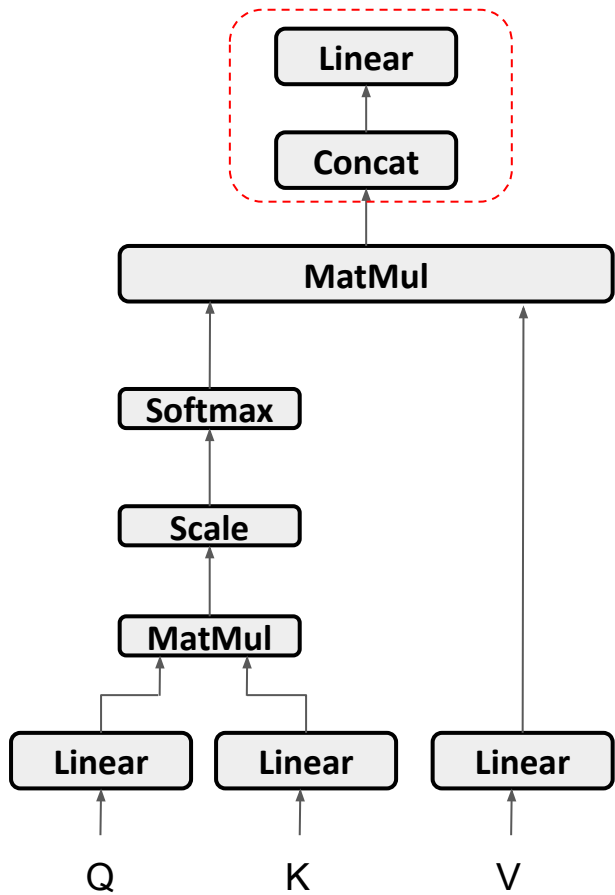


Encoder

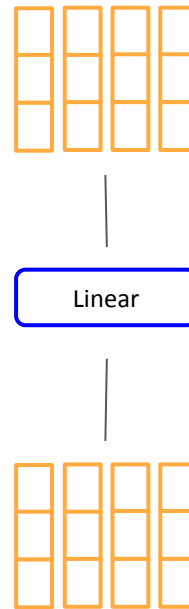
3. Multi-Headed Attention



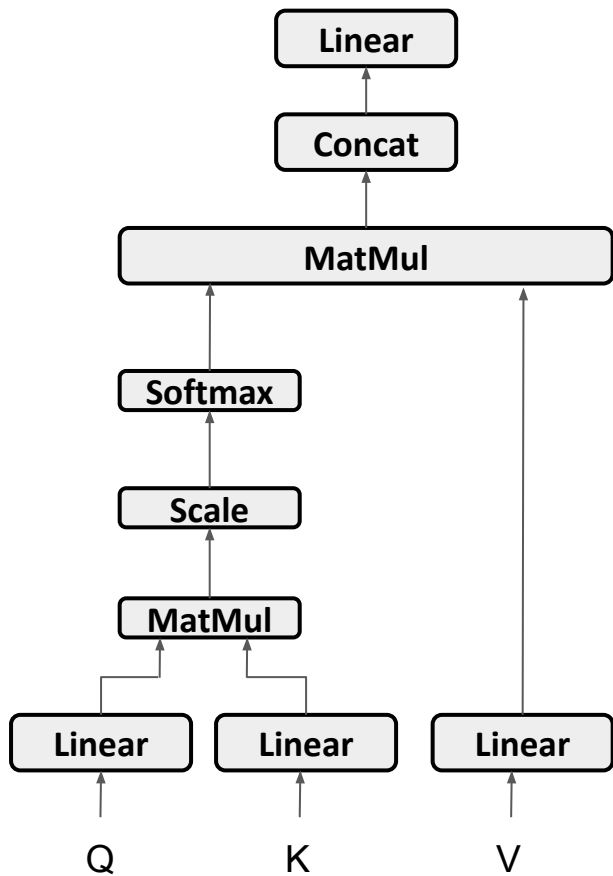
Encoder



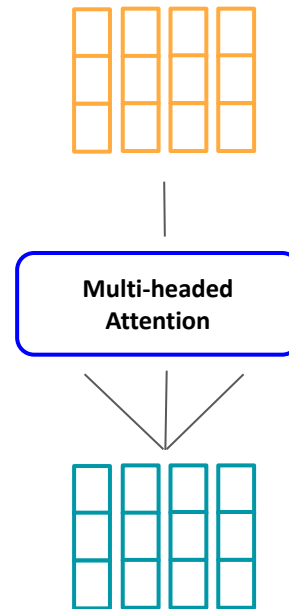
3. Multi-Headed Attention



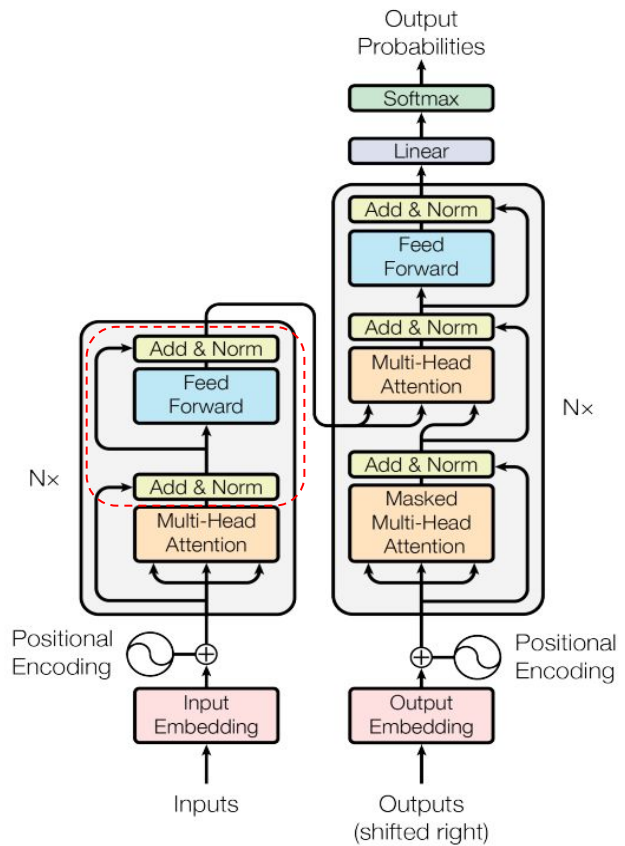
Encoder



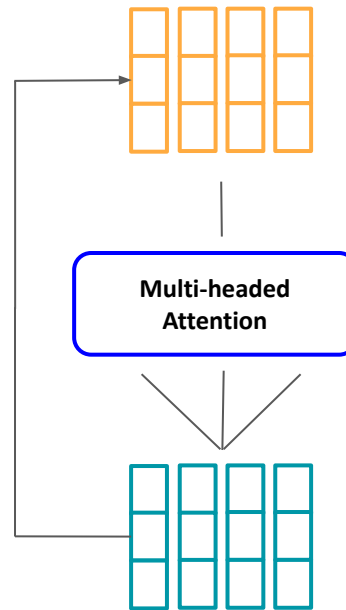
3. Multi-Headed Attention



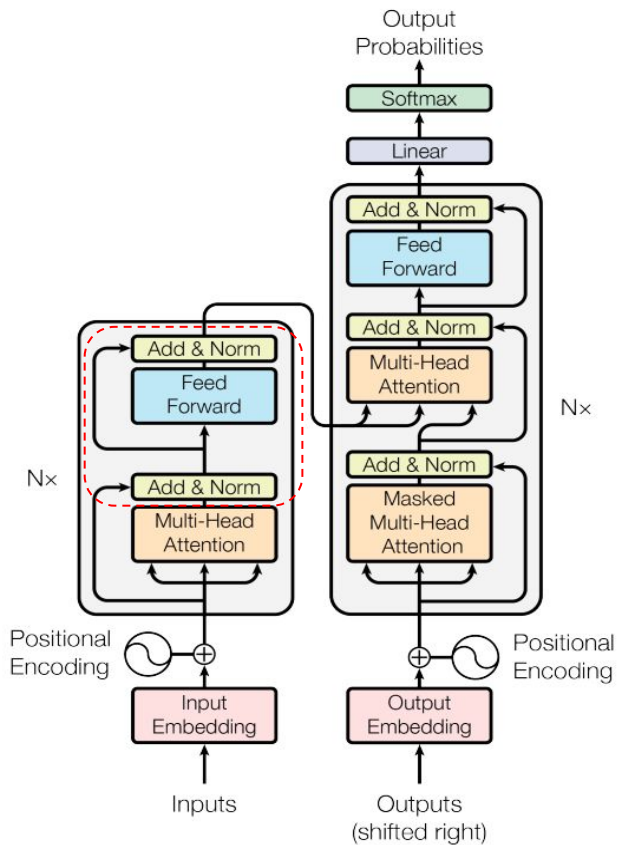
Encoder



4. Residual Connection

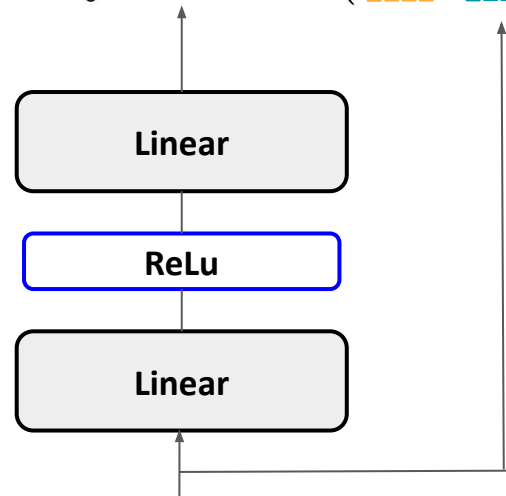


Encoder



5. Layer Normalisation and Point-wise feed forward

$$\text{LayerNorm}(\text{orange grid} + \text{teal grid})$$



$$\text{LayerNorm}(\text{orange grid} + \text{teal grid})$$