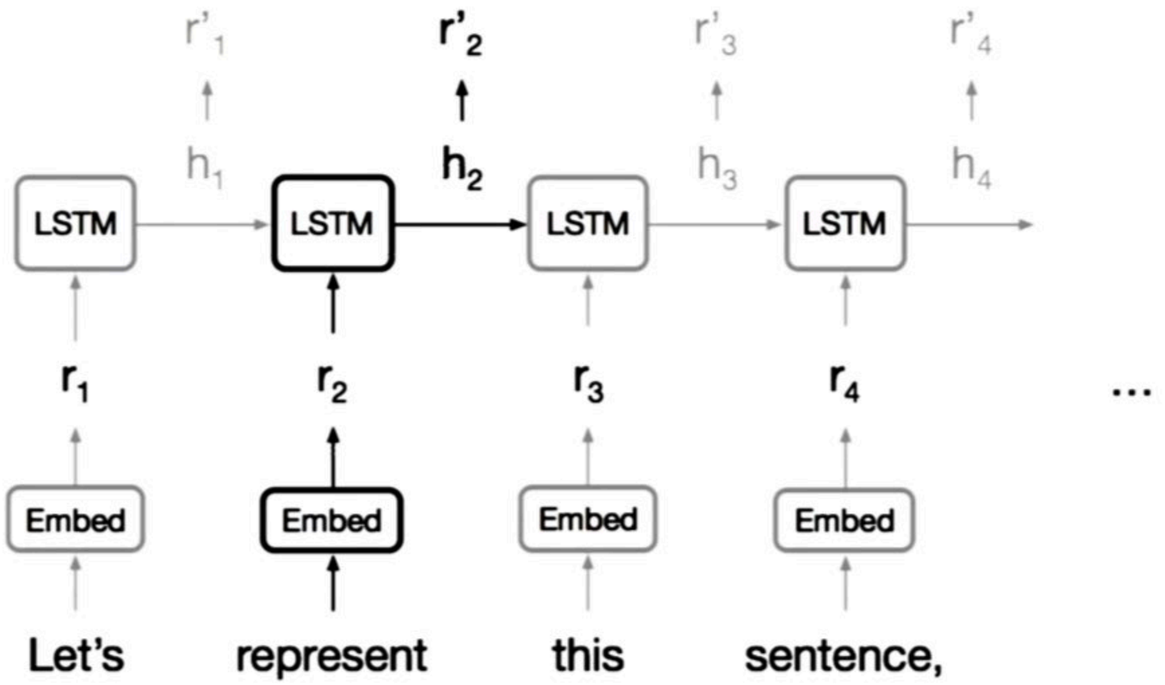
Transformers - 1

Introduction to NLP

Rahul Mishra
IIIT-Hyderabad
March 15, 2024

Motivation for a New Architecture



Motivation for a New Architecture

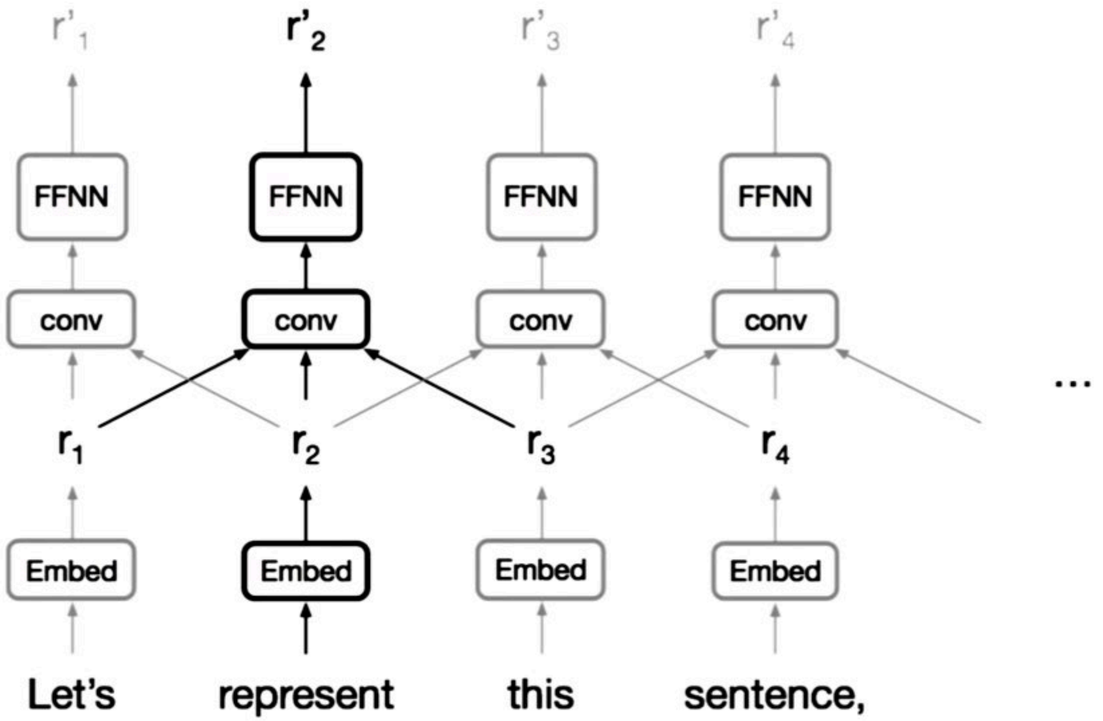
Sequential computation inhibits parallelization.

No explicit modeling of long and short range dependencies.

We want to model hierarchy.

RNNs (w/ sequence-aligned states) seem wasteful!

Motivation for a New Architecture



Motivation for a New Architecture

Trivial to parallelize (per layer).

Exploits local dependencies

‘Interaction distance’ between positions linear or logarithmic.

Long-distance dependencies require many layers.

Motivation for a New Architecture

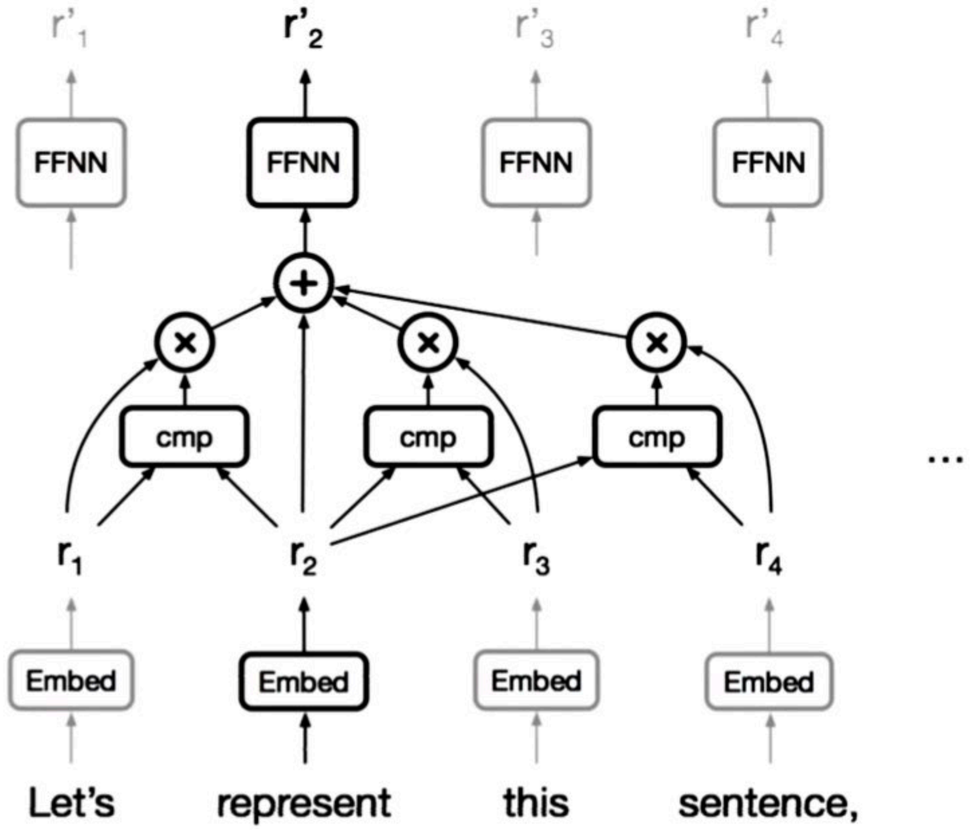
- Vanishing gradient problem
- Parallel Processing
- No Recurrent Connections
- Better Capturing Long-Term Dependencies

Attention is All You Need:Transformers

@grownupgaming 1 year ago

I would be scared for my life if I was Ashish. Someone from the future might travel back in time and kill me.

Self-Attention



Self-Attention

Classification & regression with self-attention:

Parikh et al. (2016), Lin et al. (2016)

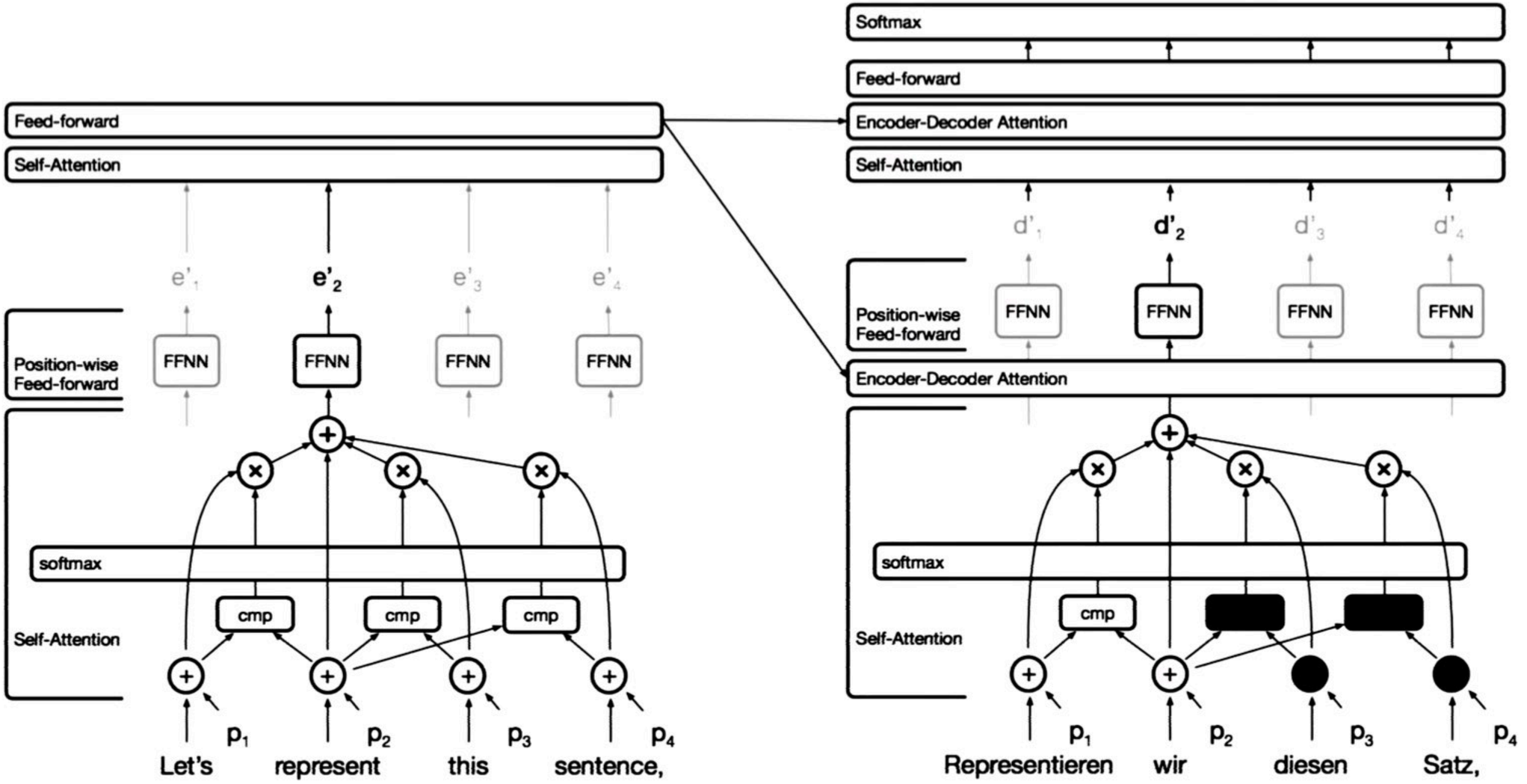
Self-attention with RNNs:

Long et al. (2016), Shao, Gows et al. (2017)

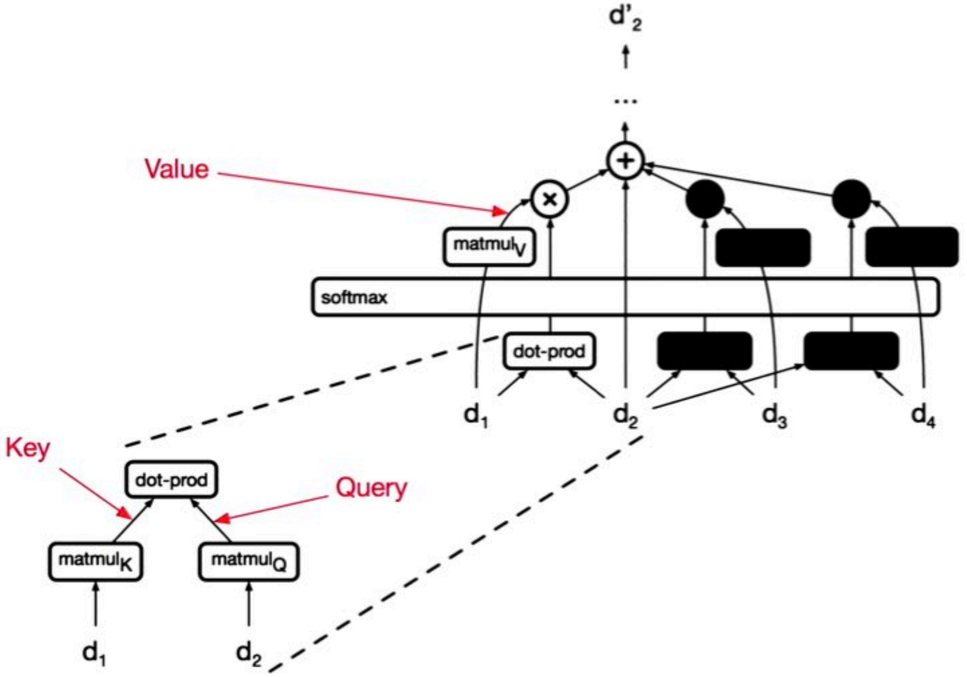
Recurrent attention:

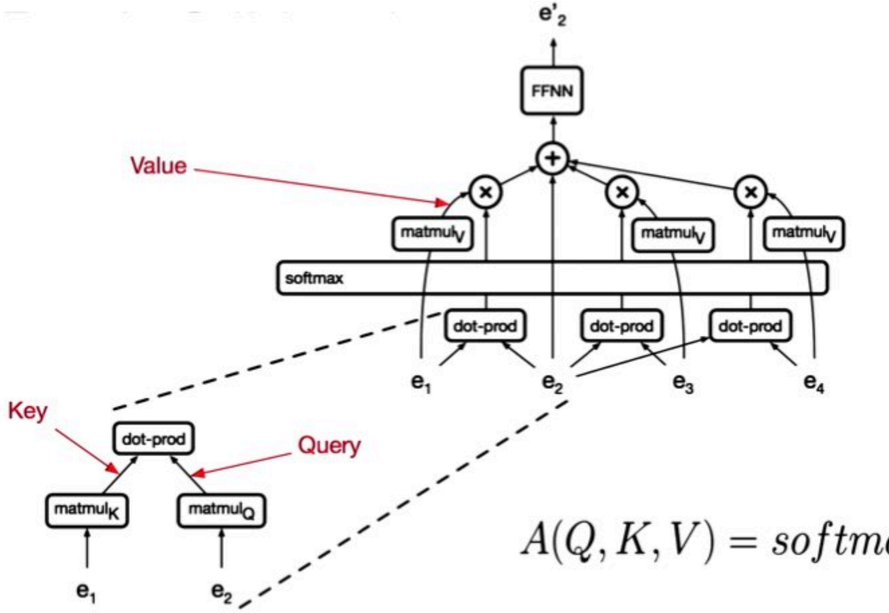
Sukhbaatar et al. (2015)

Rough Idea of Transformers



Rough Idea of Transformers



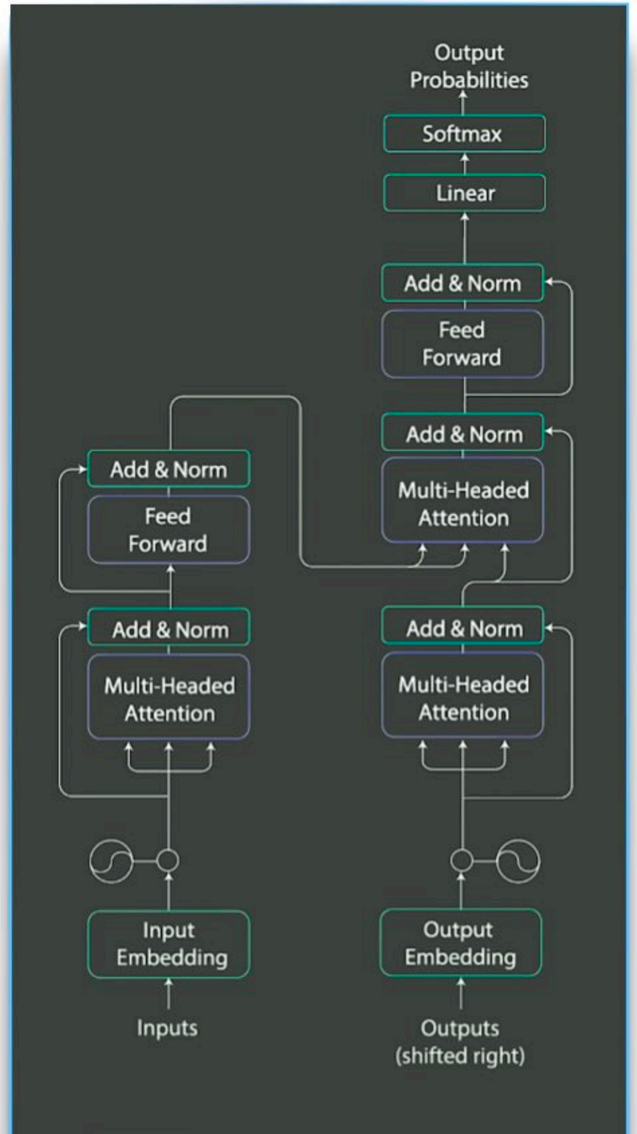


$$A(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

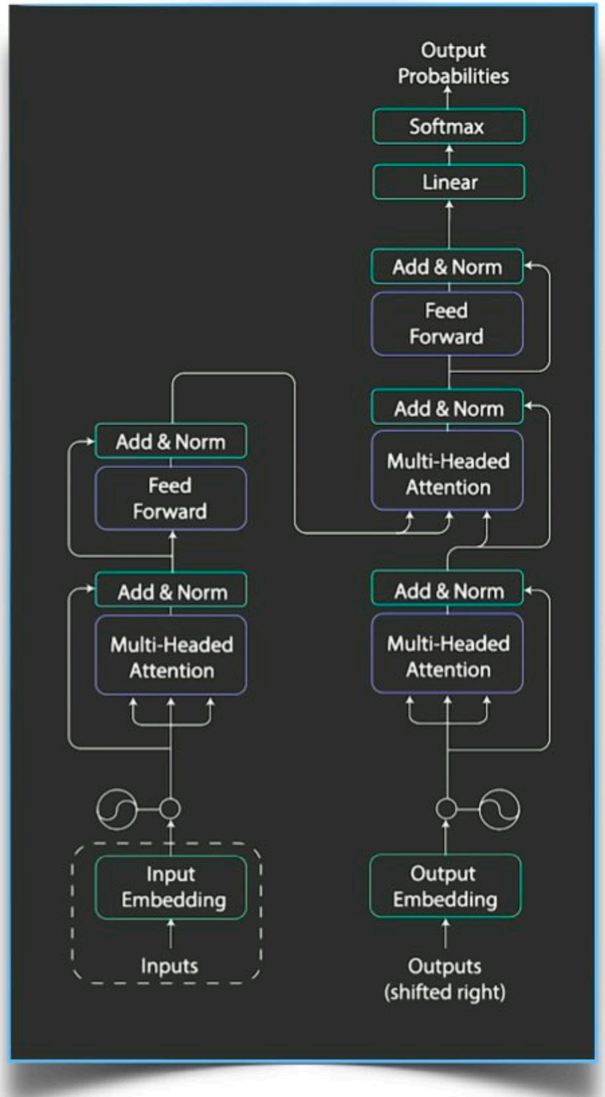
Is it complex? Yes and No

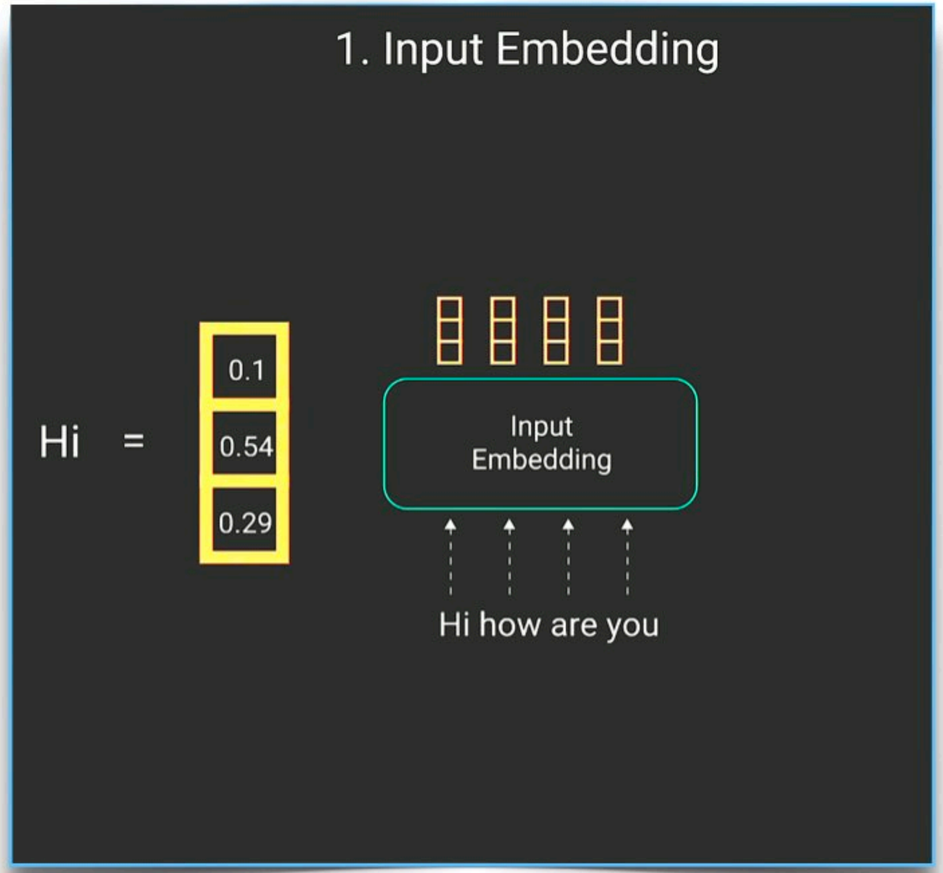
	FLOPs
Self-Attention	$O(\text{length}^2 \cdot \text{dim})$
RNN (LSTM)	$O(\text{length} \cdot \text{dim}^2)$
Convolution	$O(\text{length} \cdot \text{dim}^2 \cdot \text{kernel\_width})$

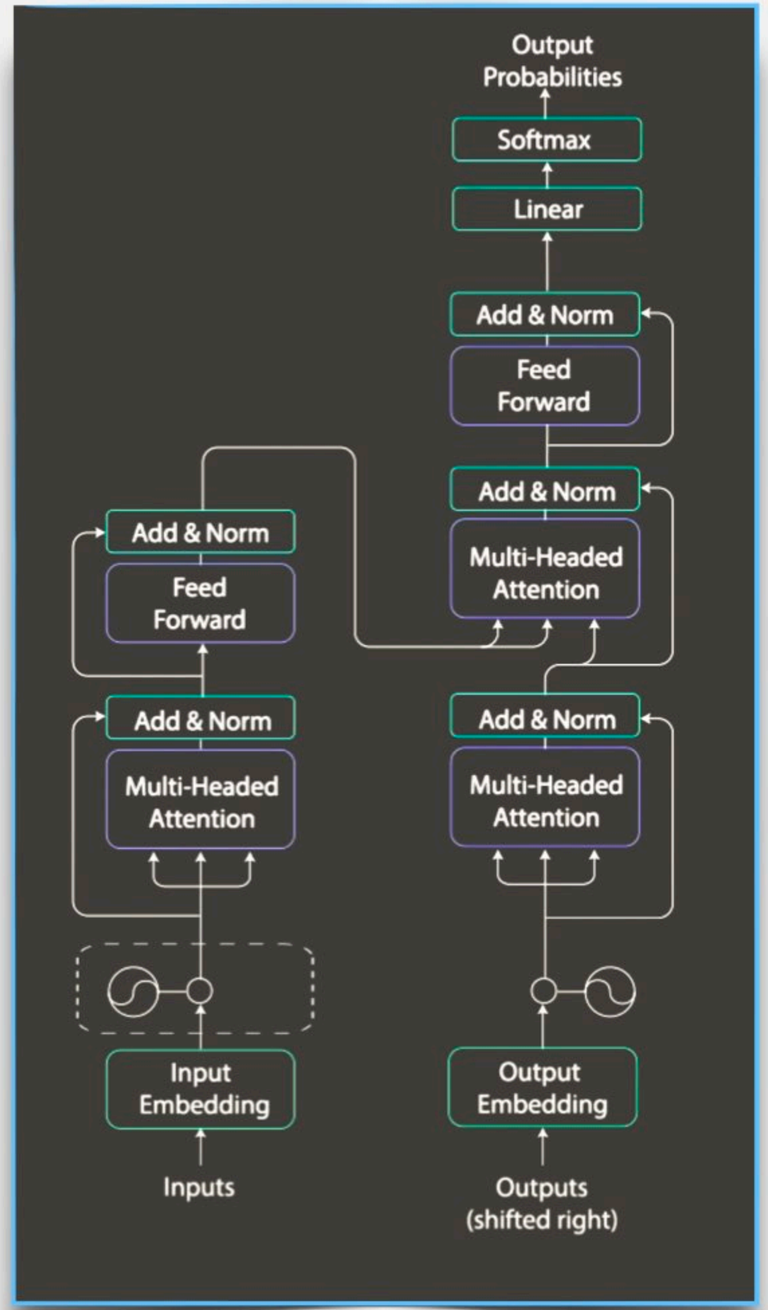
Transformers: Attention Is All You Need



Transformers: Attention Is All You Need







2. Positional Encoding

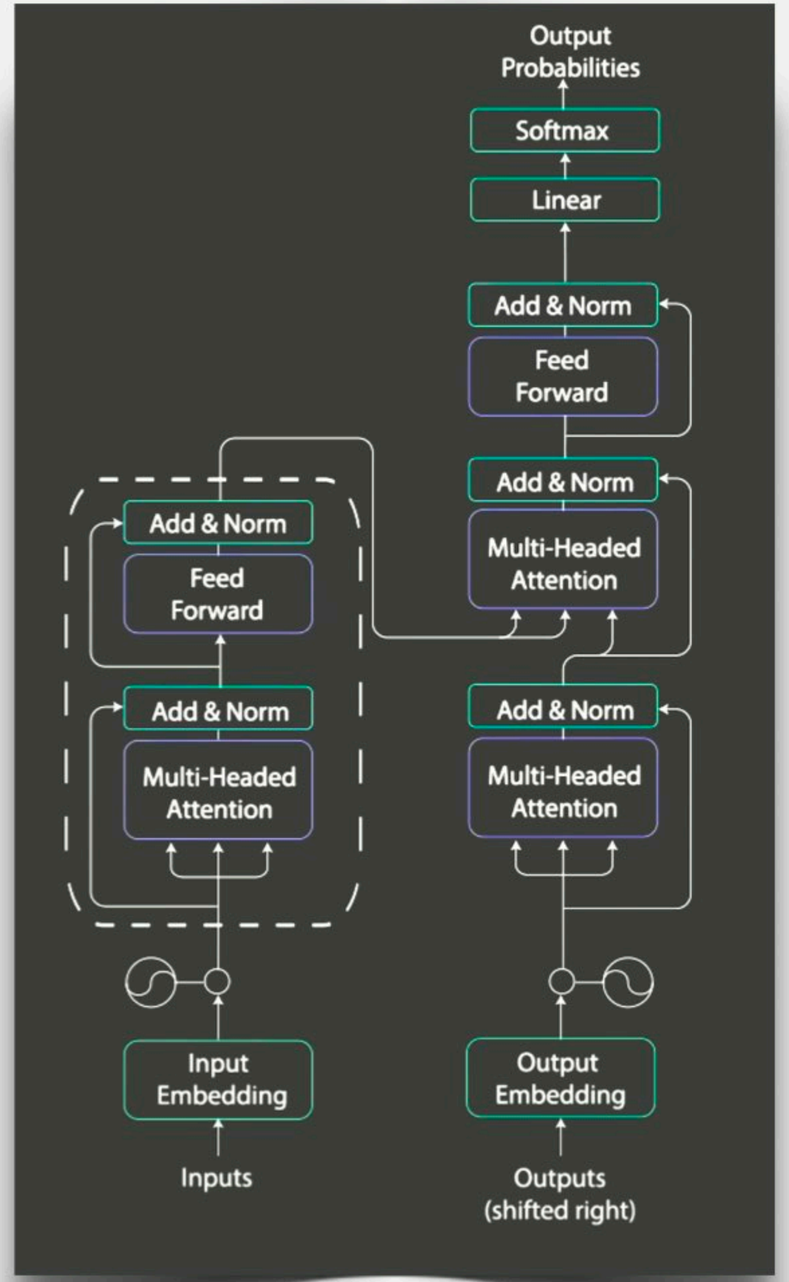
Positional Input Embeddings

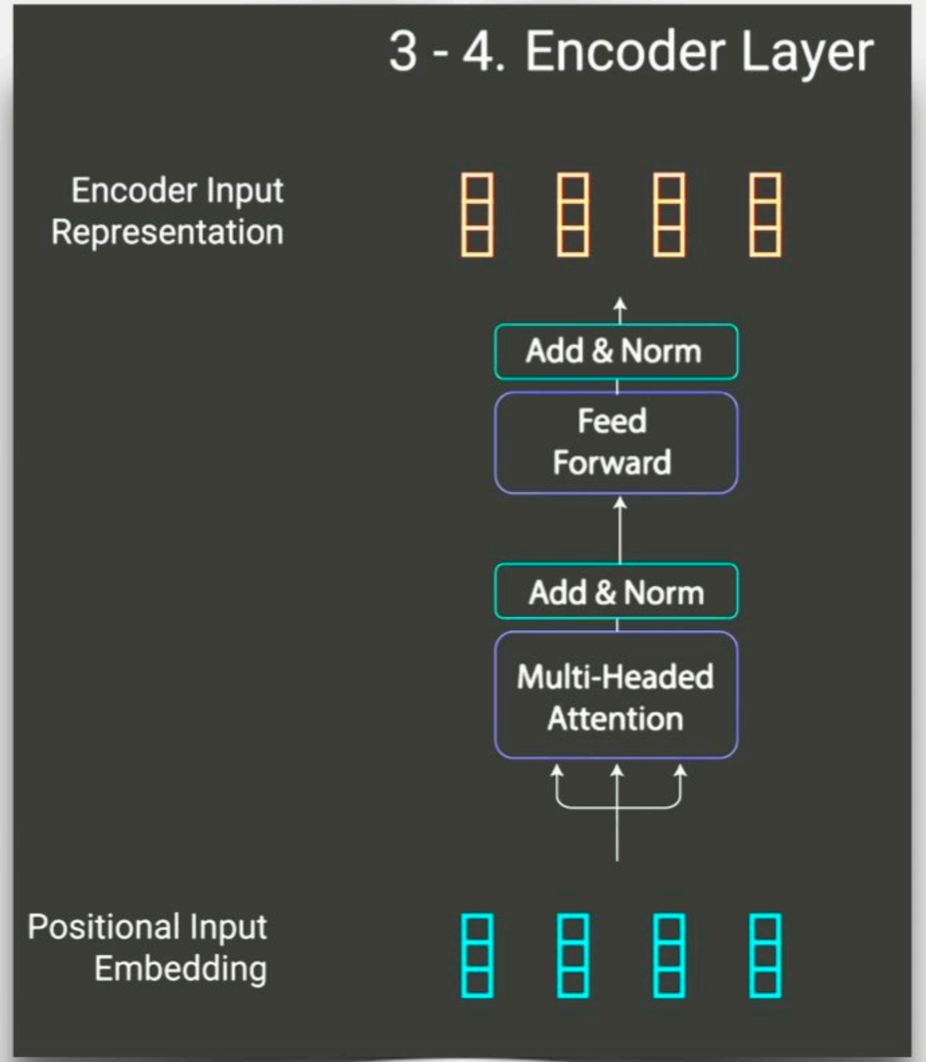
Positional Encoding

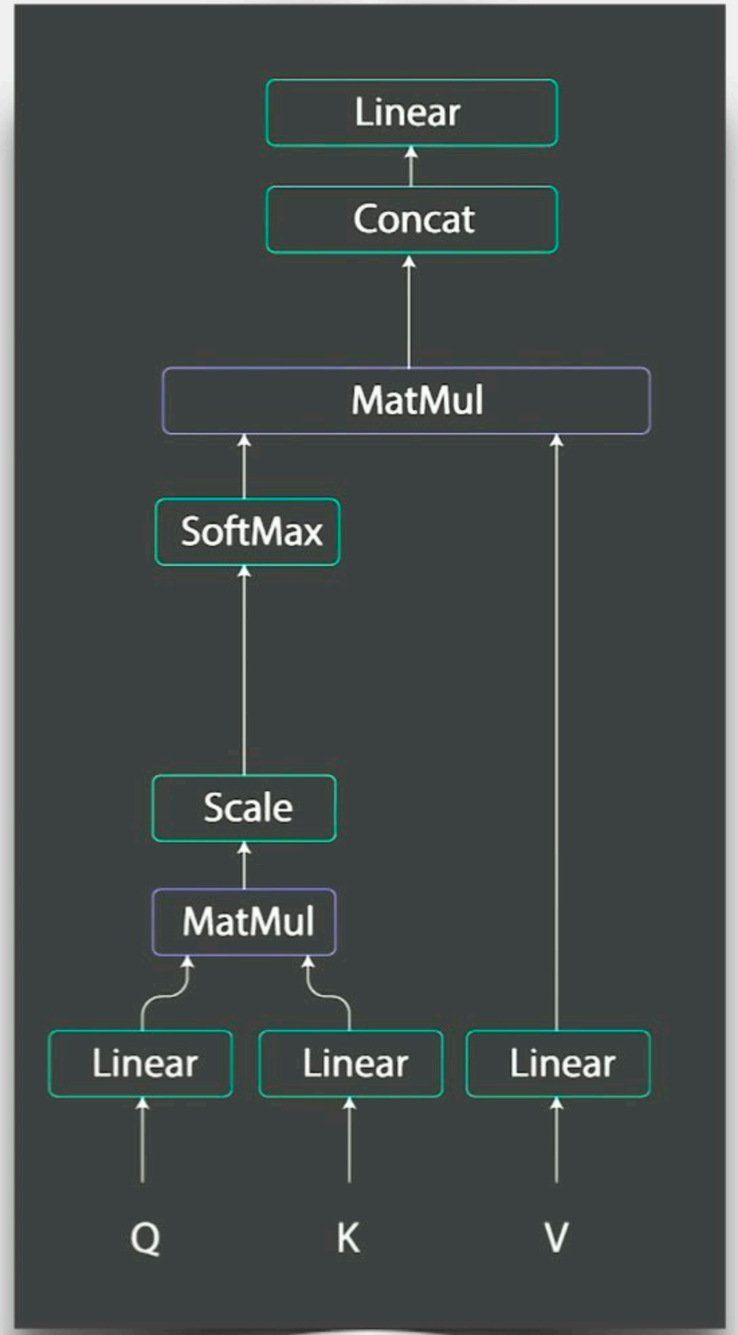
Time Step 1 2 3 4

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

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

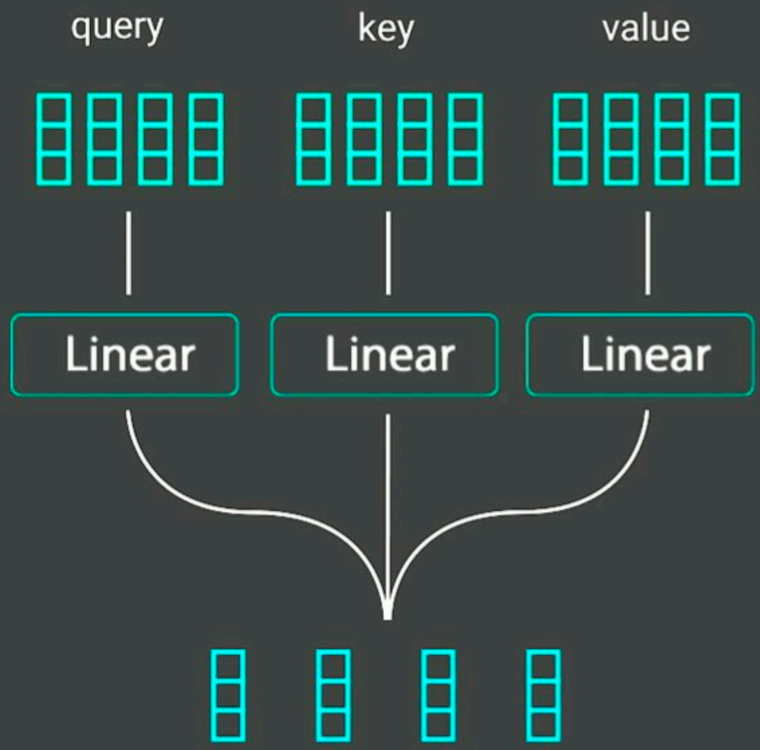

Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Headed Attention
Add & Norm
Multi-Headed Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Headed Attention
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

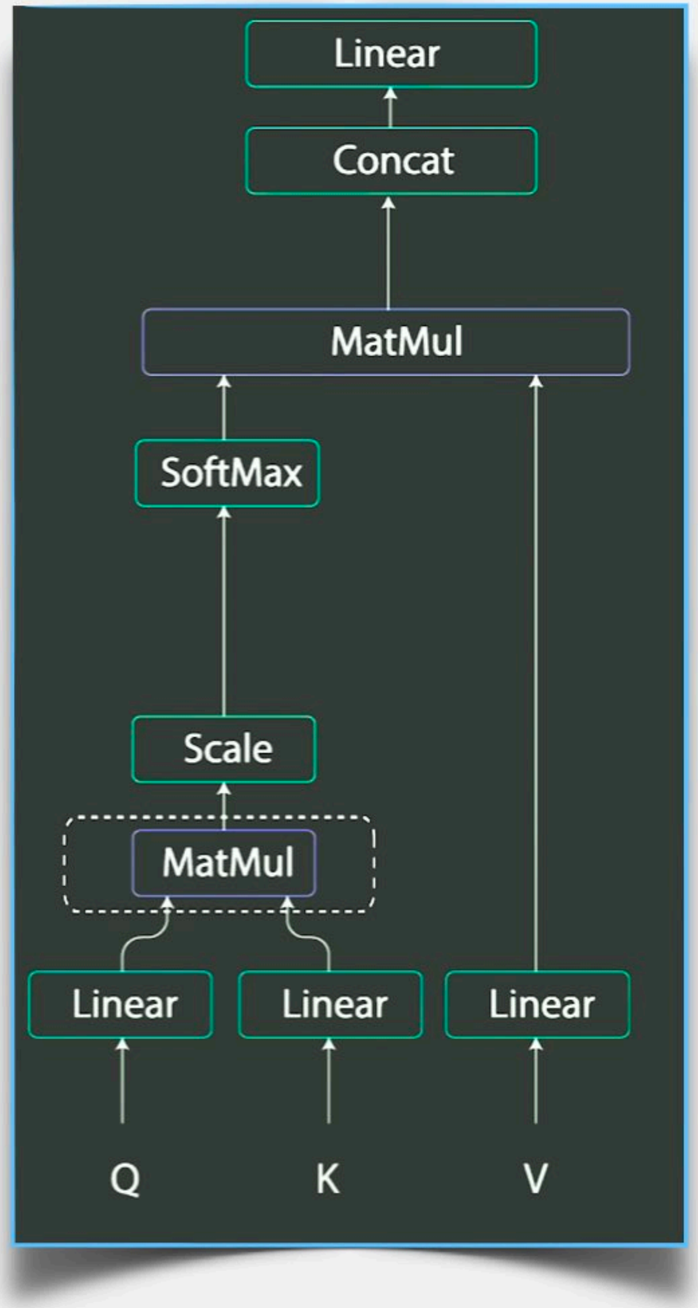

3 - 4. Encoder Layer
Encoder Input Representation
Add & Norm
Feed Forward
Add & Norm
Multi-Headed Attention
Positional Input Embedding



3. Multi-headed Attention

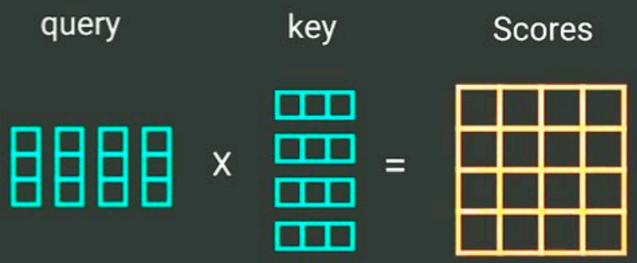
3.1. Self-Attention

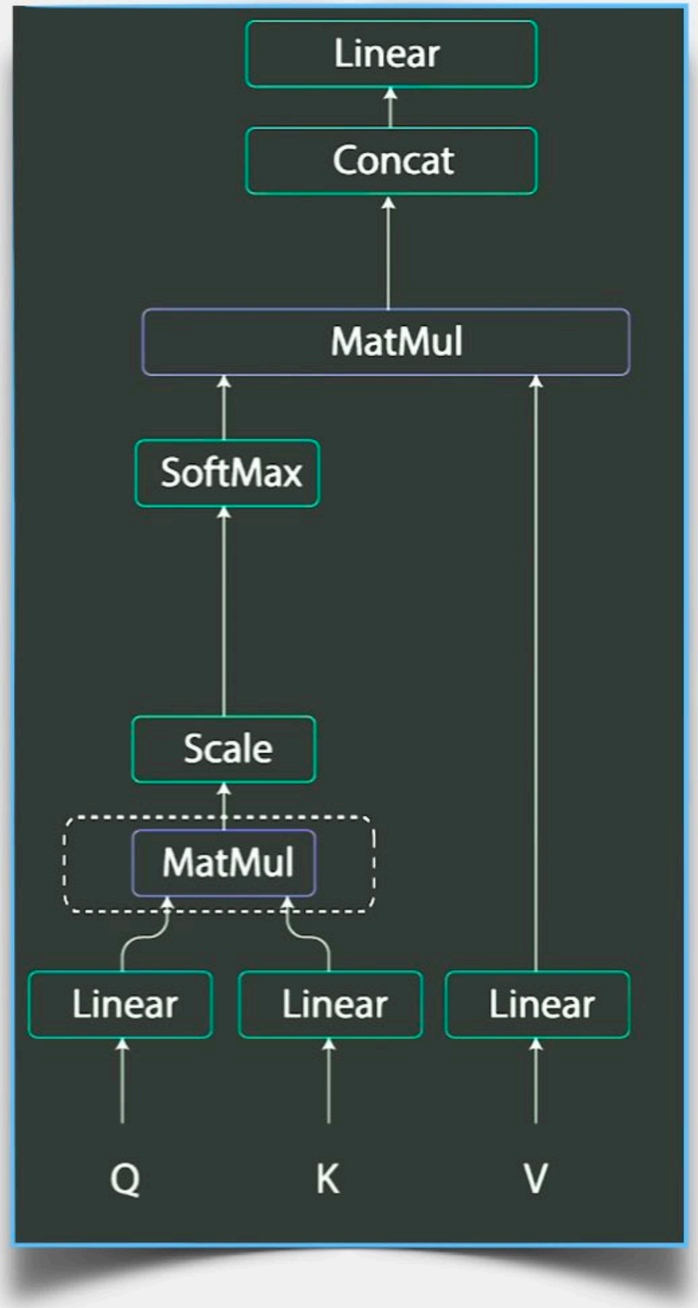




3. Multi-headed Attention

3.1. Self-Attention

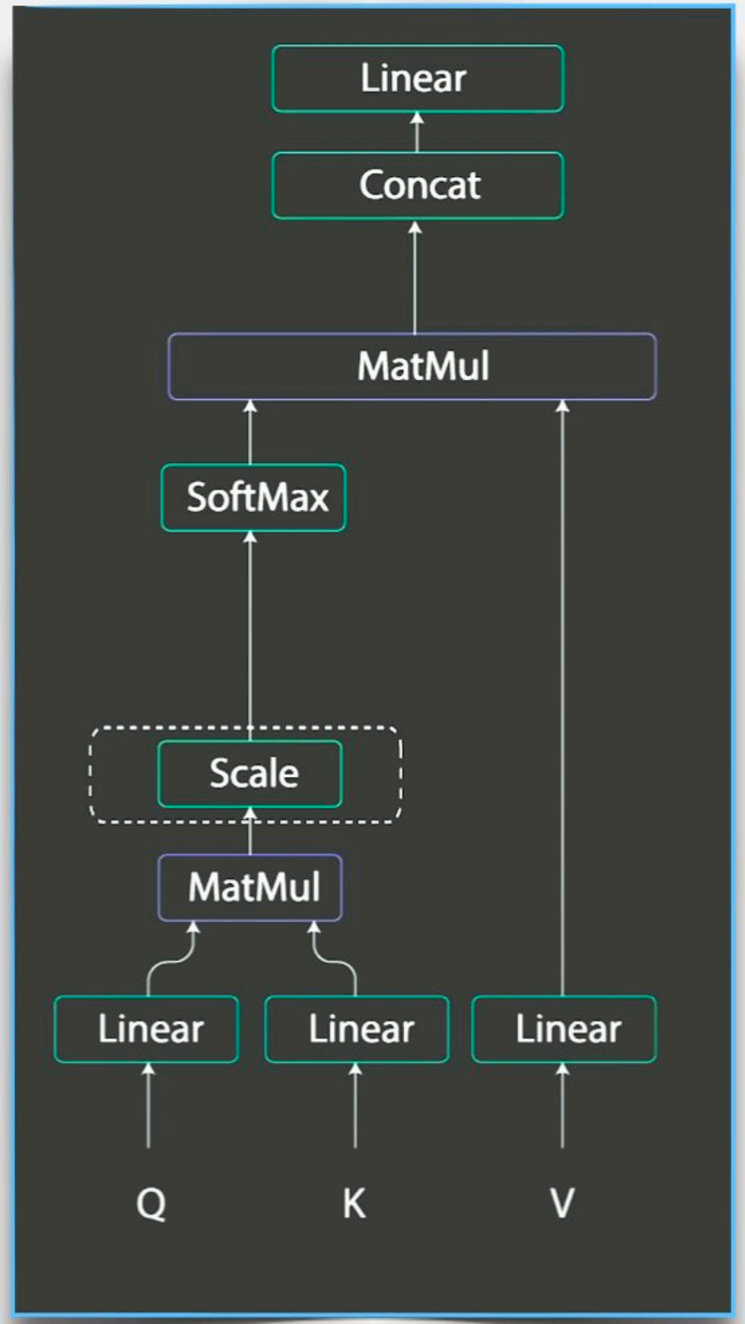




3. Multi-headed Attention

3.1. Self-Attention

	Hi	how	are	you
Hi	98	27	10	12
how	27	89	31	67
are	10	31	91	54
you	12	67	54	92

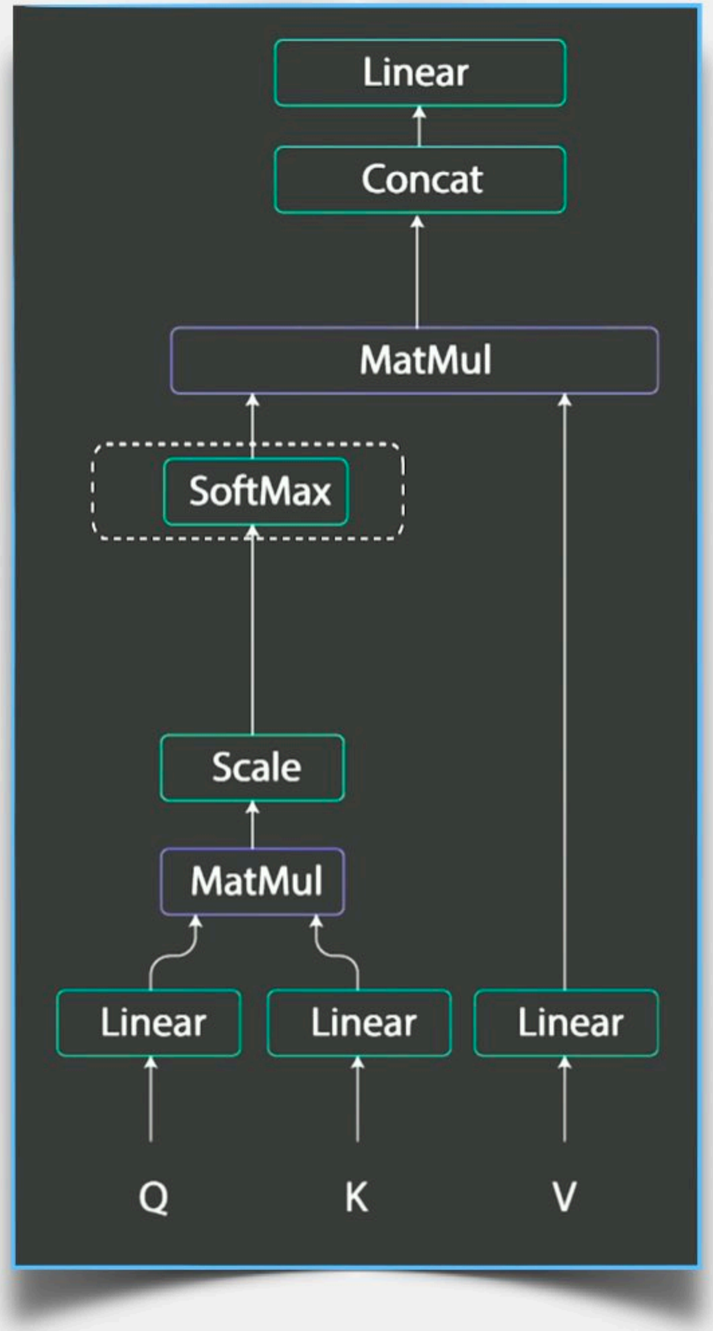


3. Multi-headed Attention

3.1. Self-Attention

$$\frac{\square}{\sqrt{d_k}} = \square$$

Scaled Scores



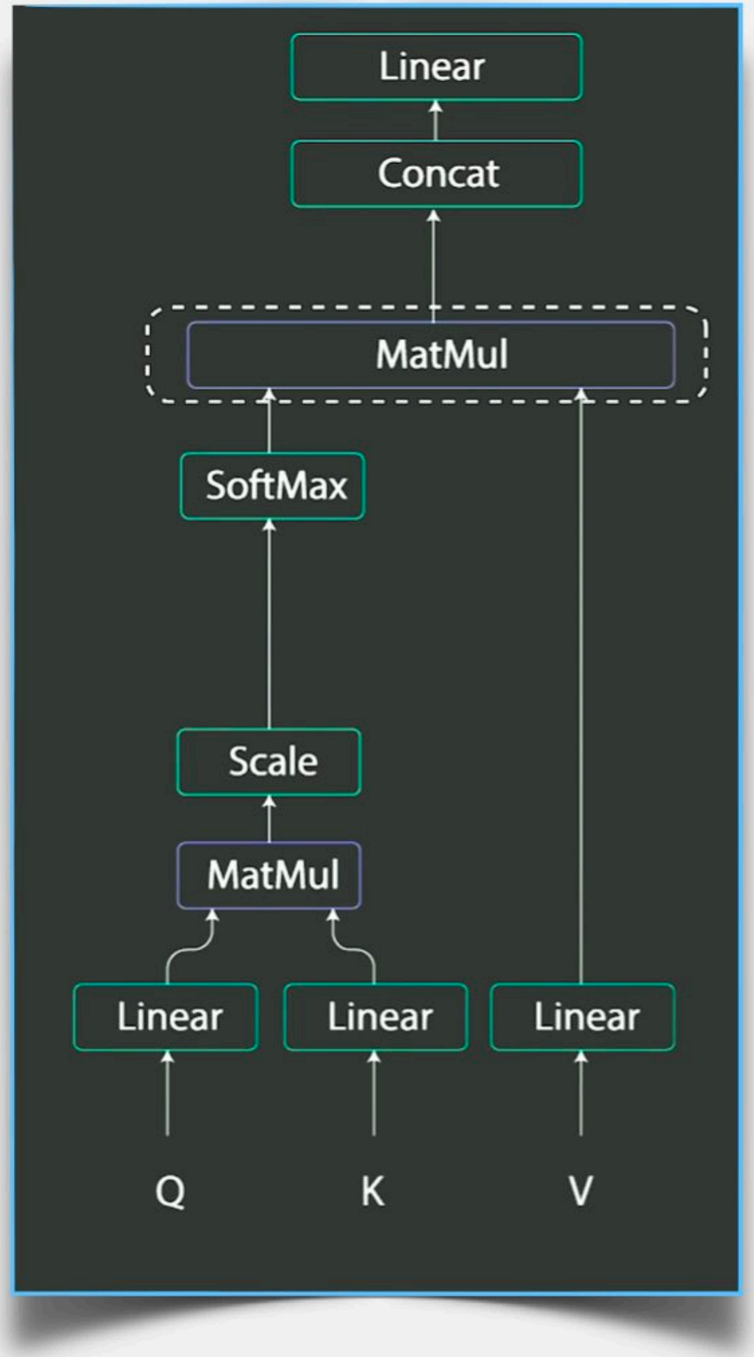
3. Multi-headed Attention

3.1. Self-Attention

Softmax(▦) =

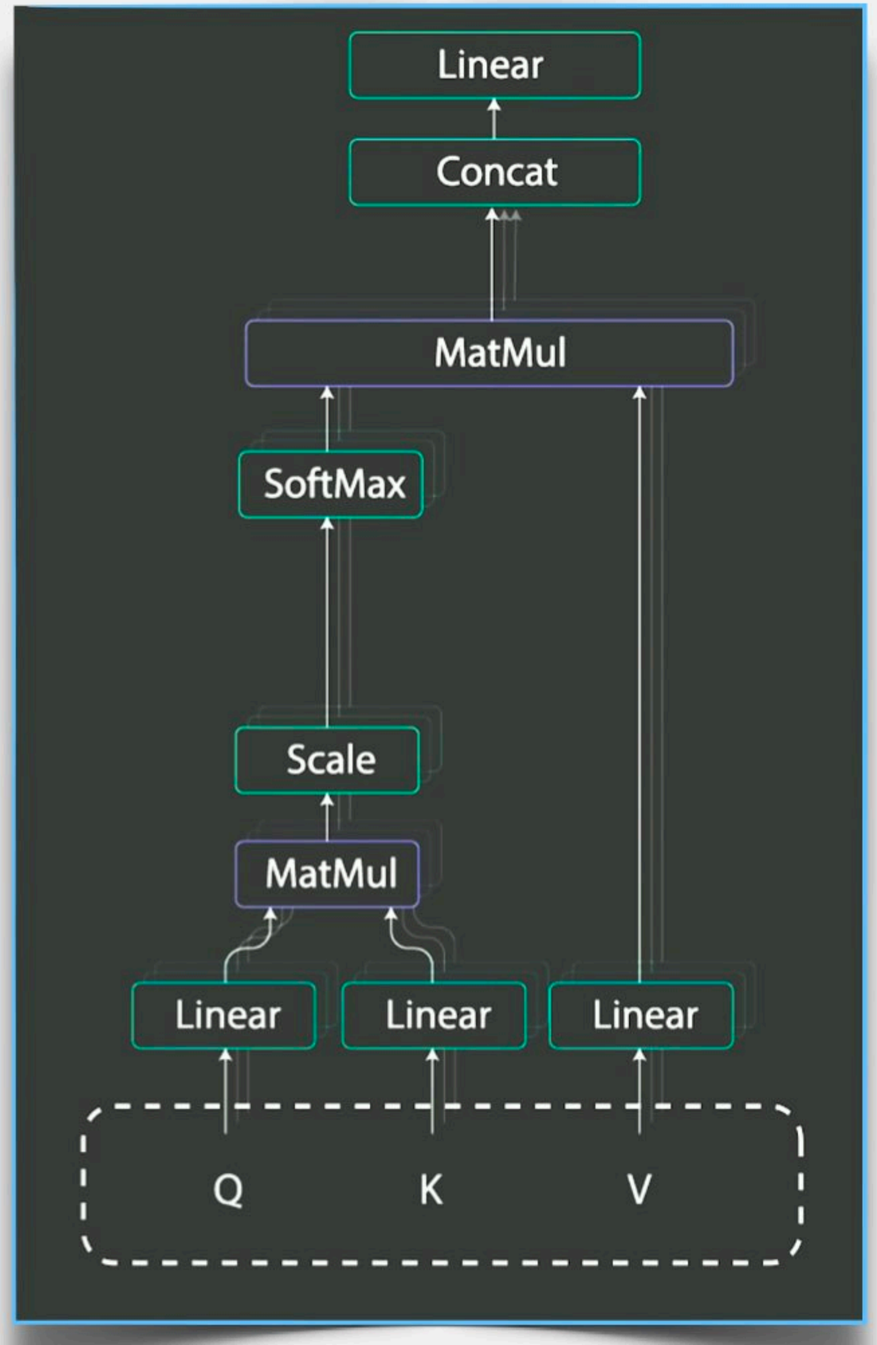
	Hi	how	are	you
Hi	0.7	0.1	0.1	0.1
how	0.1	0.6	0.2	0.1
are	0.1	0.3	0.6	0.1
you	0.1	0.3	0.3	0.3

$$softmax(x)_i = \frac{exp(x_i)}{\sum_j exp(x_j))}$$

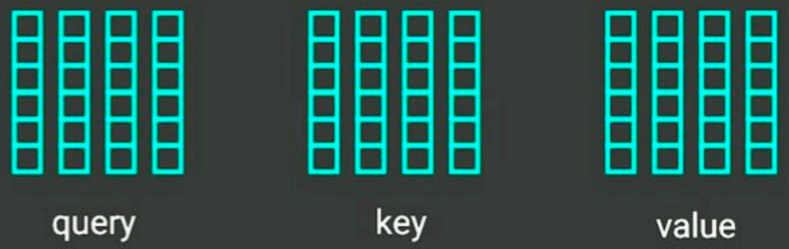


3. Multi-headed Attention

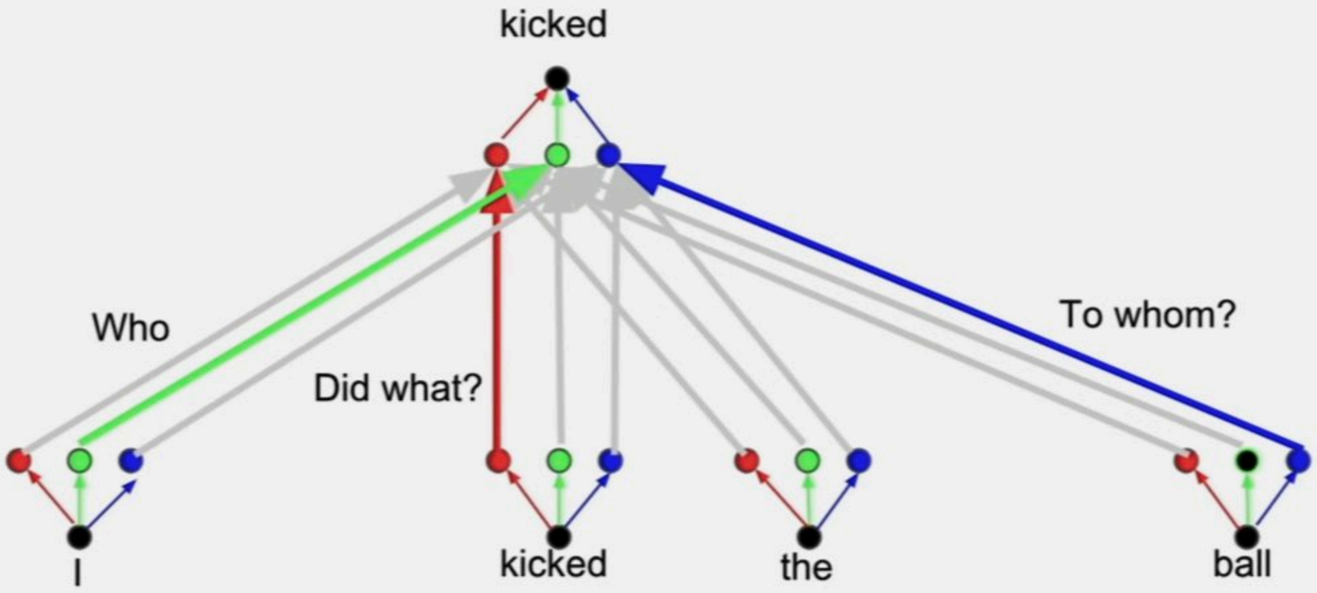
3.1. Self-Attention

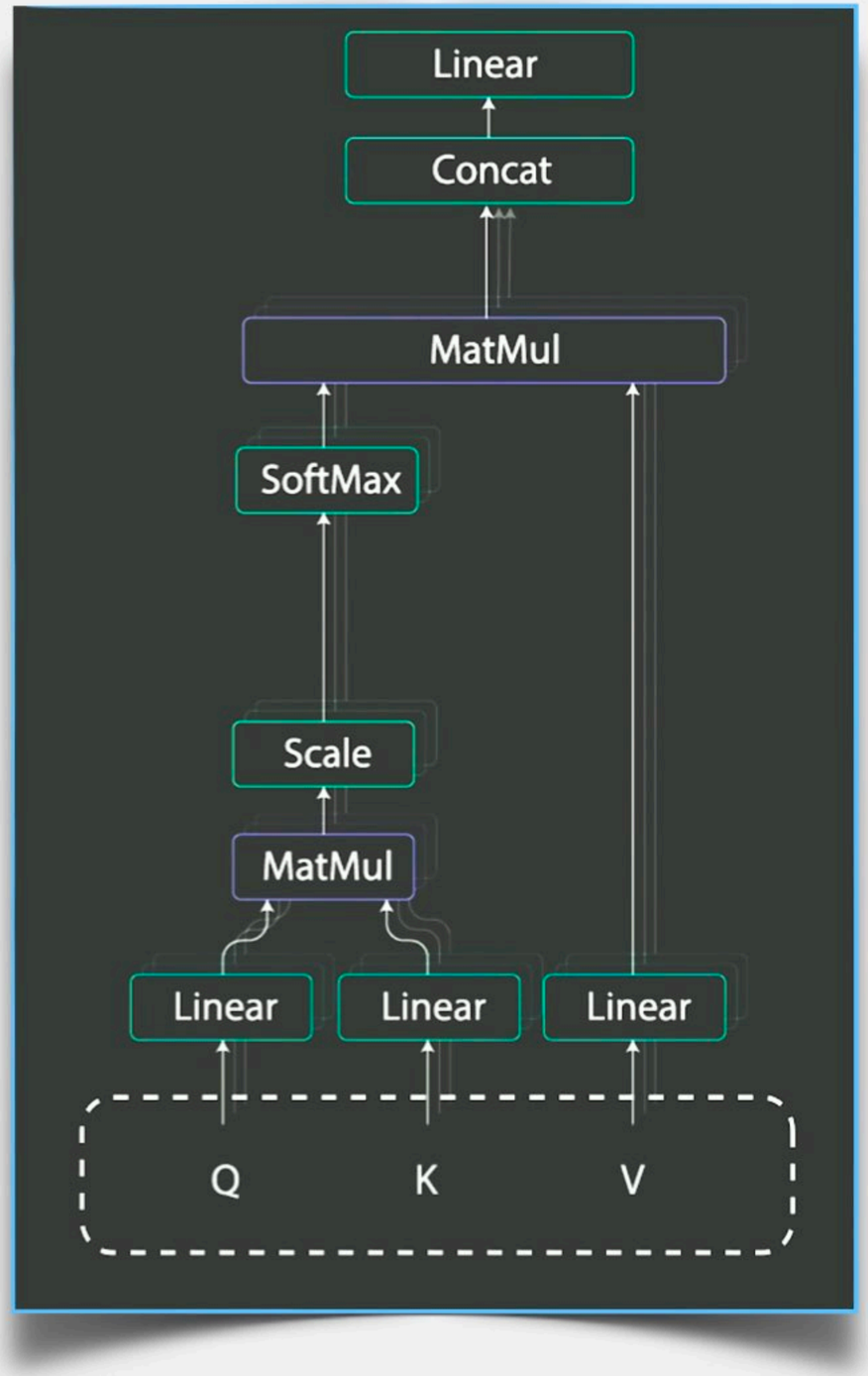


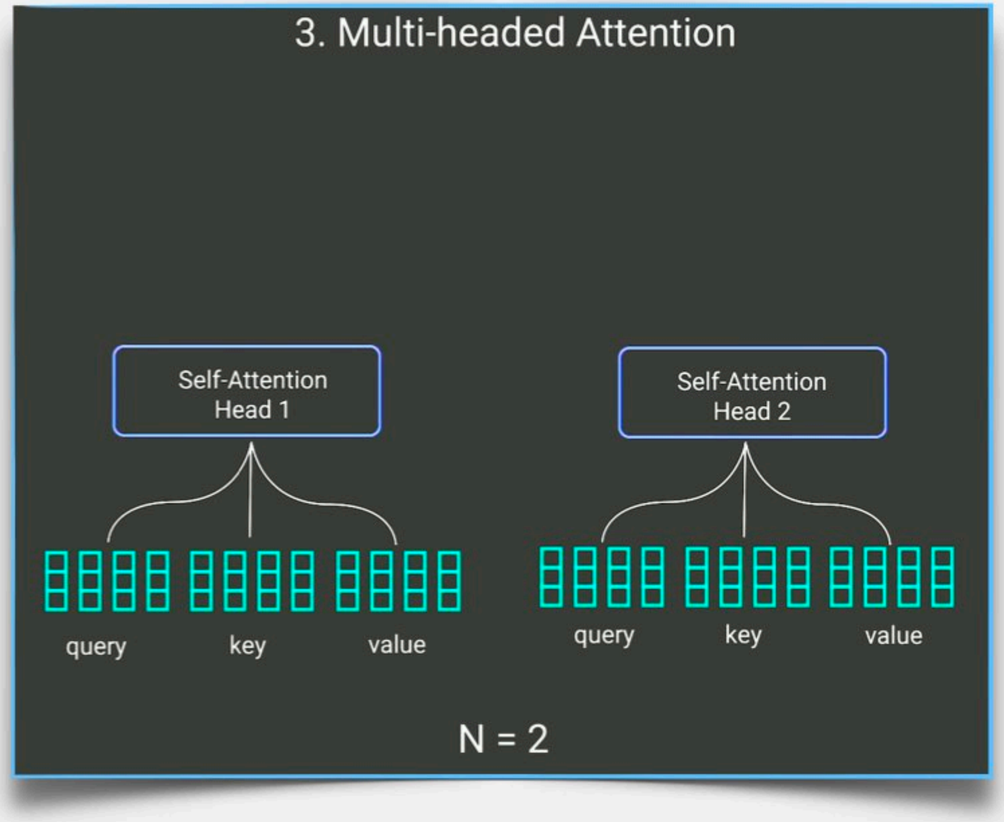
3. Multi-headed Attention

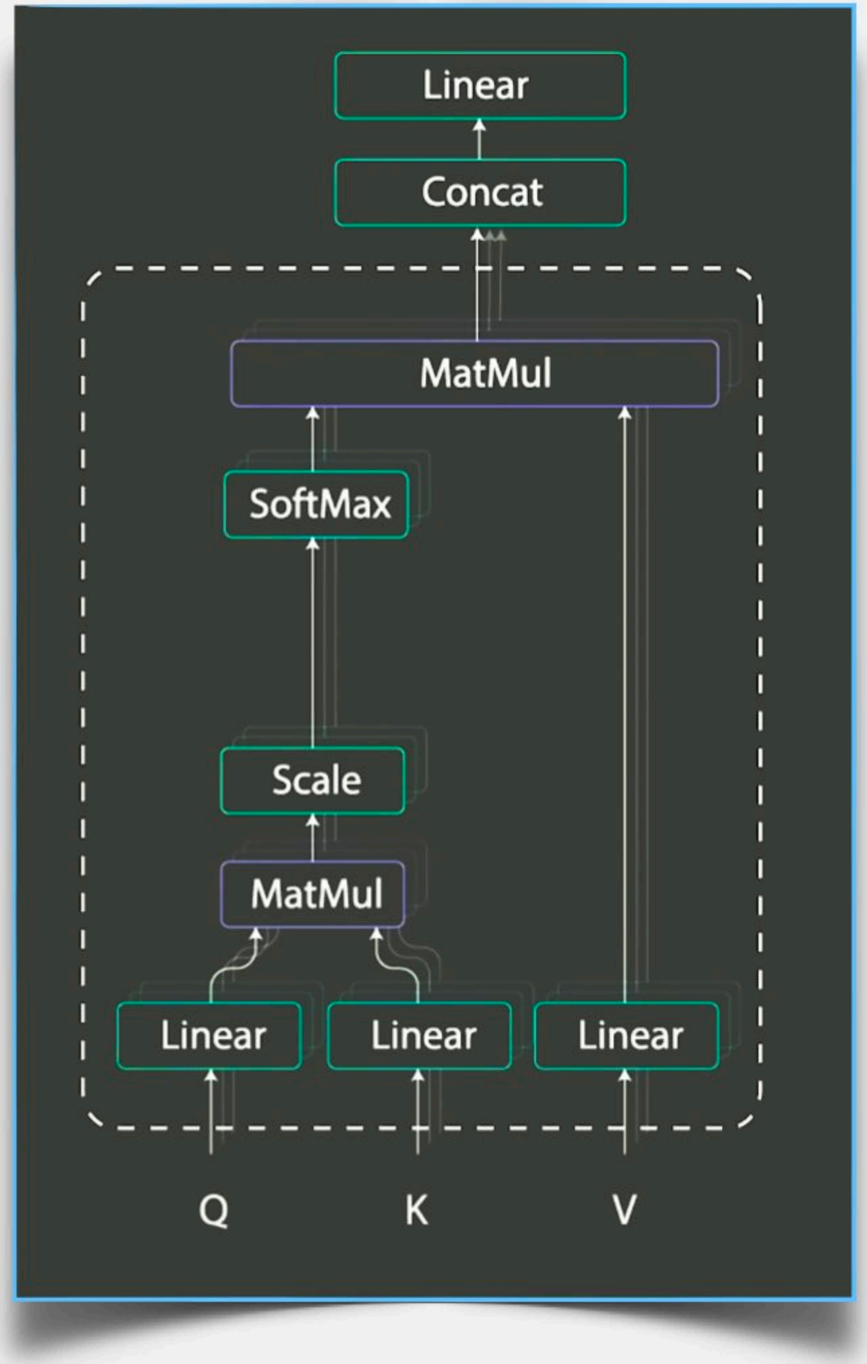


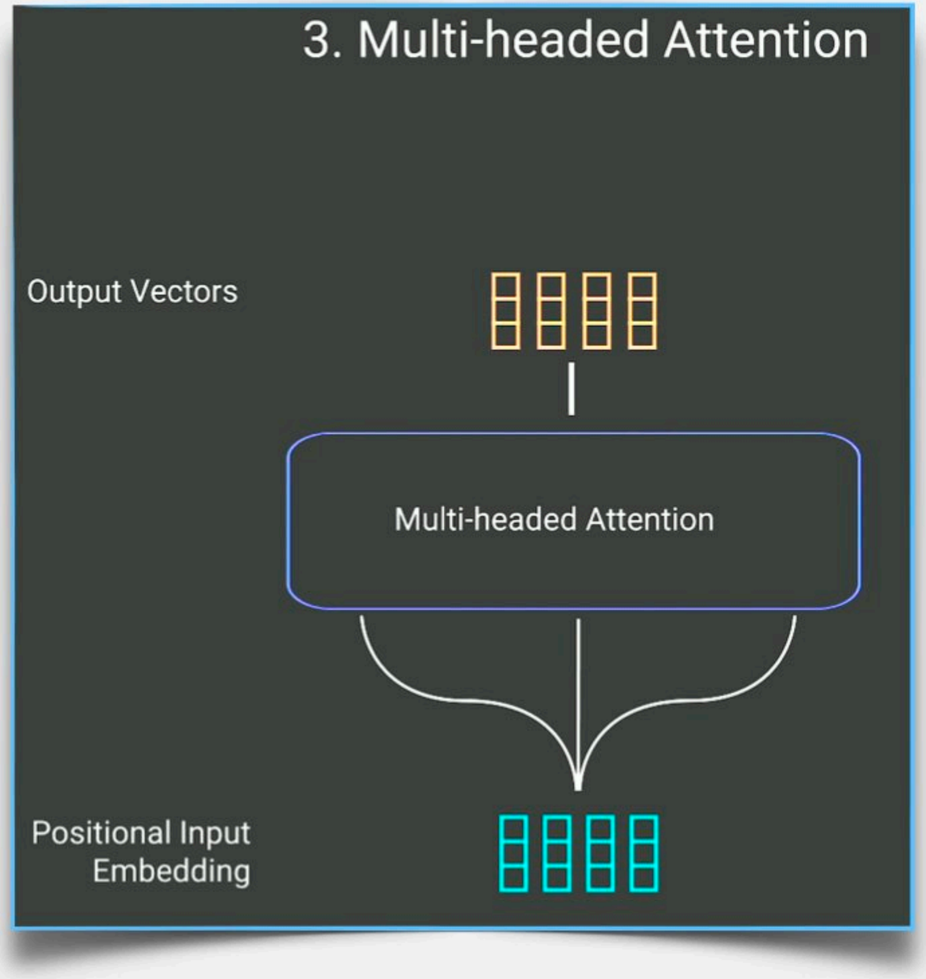
Attention Head-Who, What, and Whom

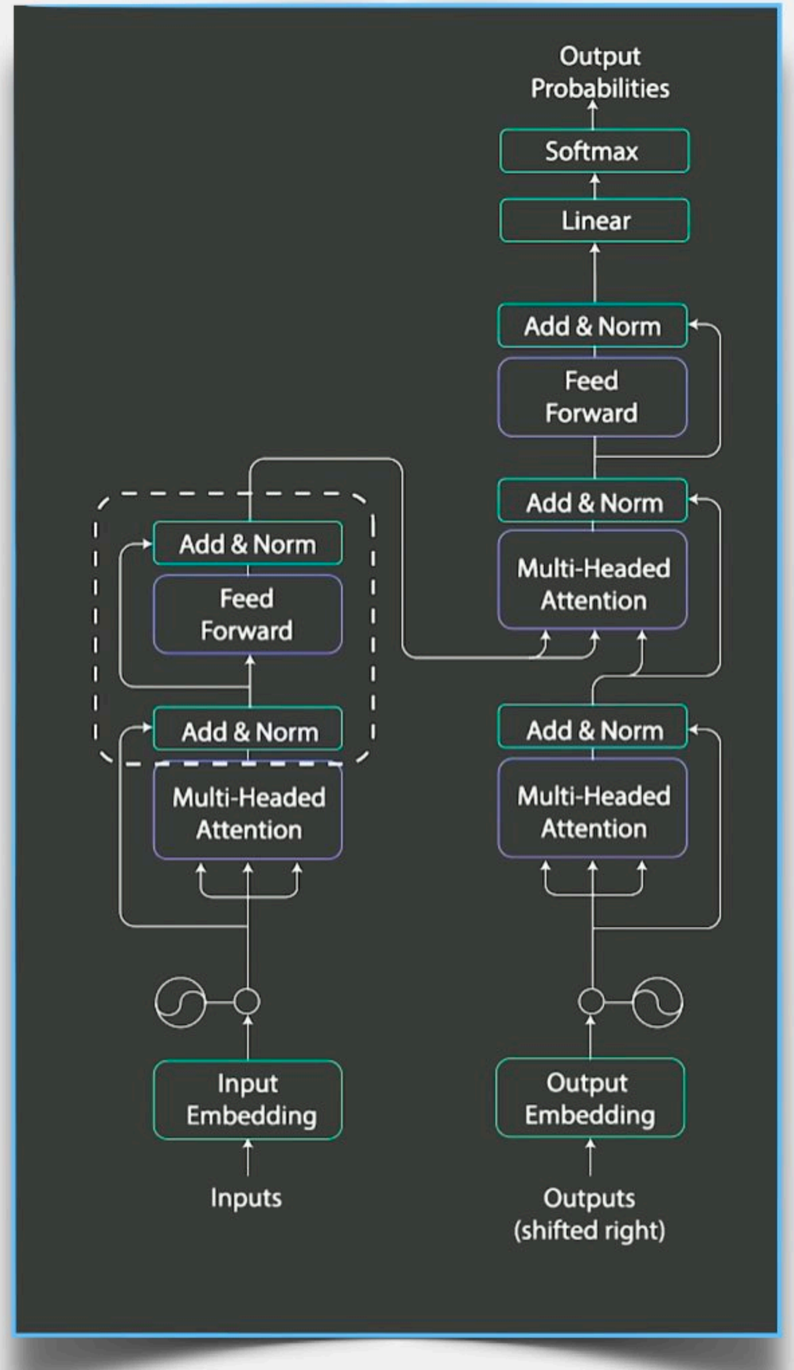


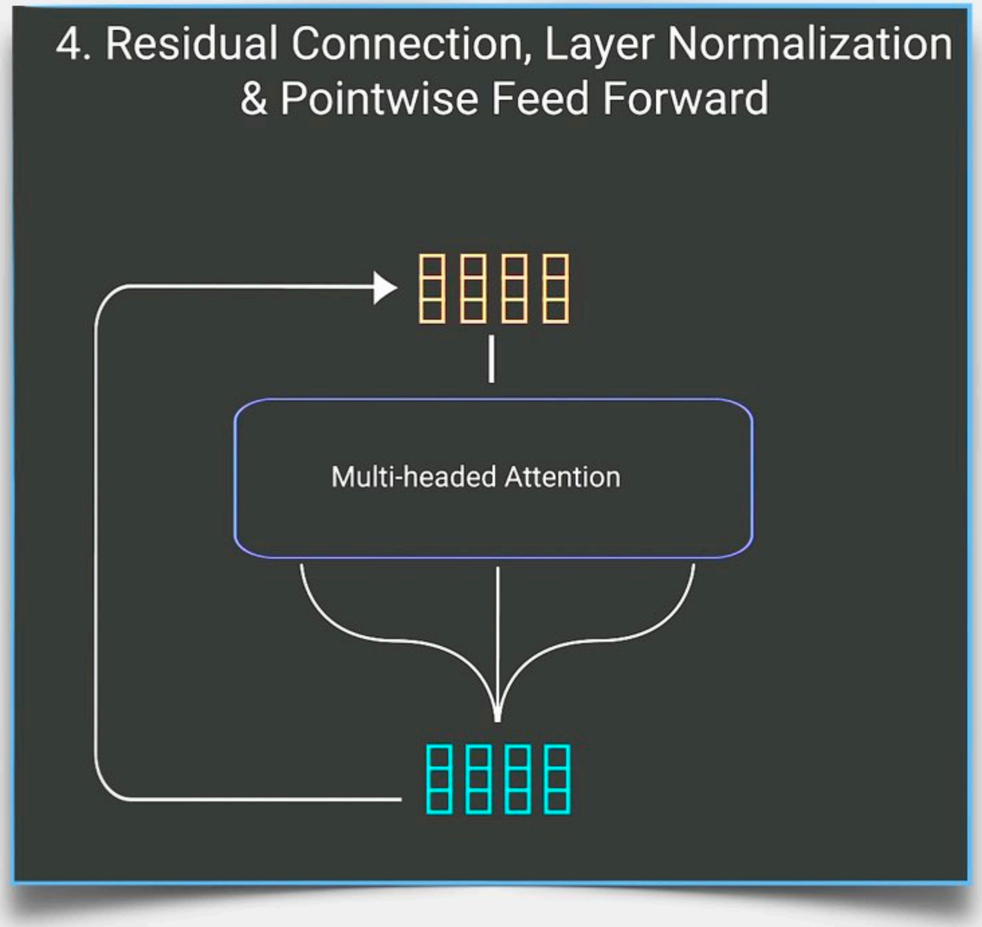

Linear
Concat
MatMul
SoftMax
Scale
MatMul
Linear
Linear
Linear
Q
K
V

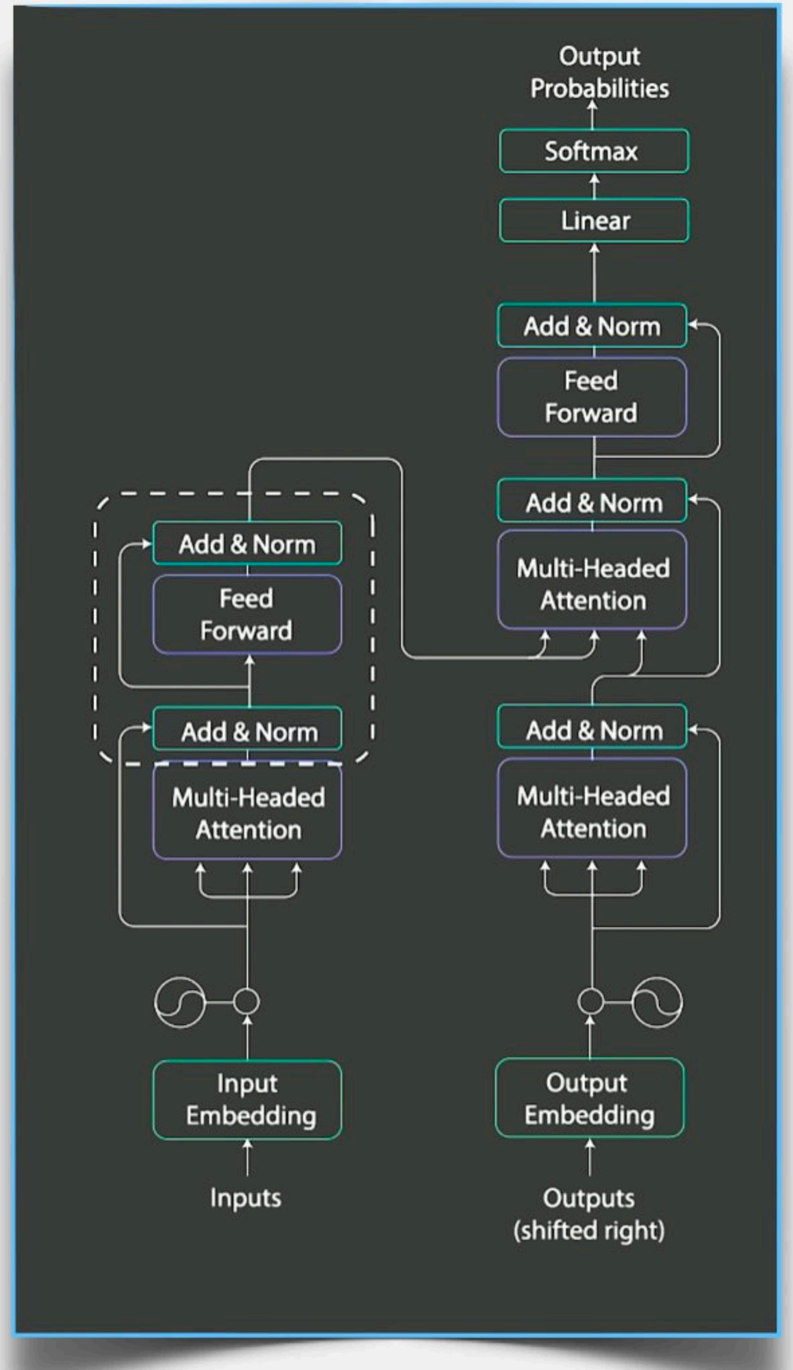

3. Multi-headed Attention
Self-Attention Head 1
Self-Attention Head 2
query
key
value
query
key
value
N = 2

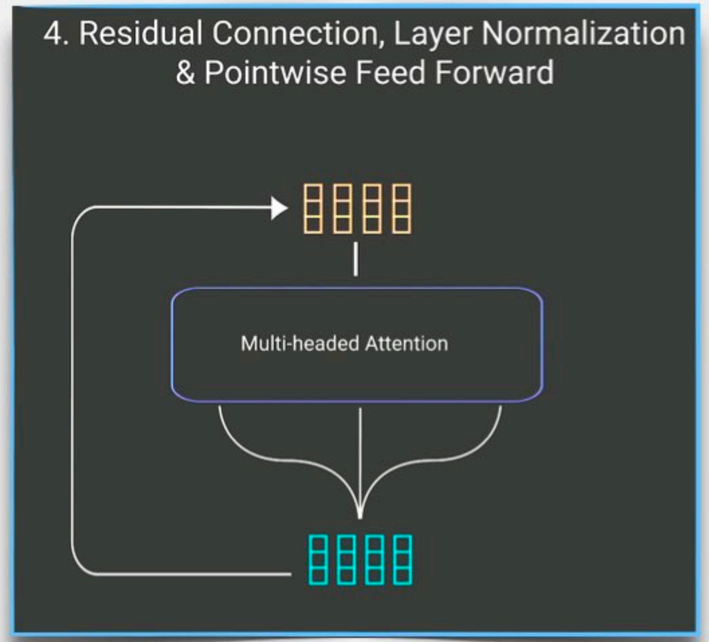

Linear
Concat
MatMul
SoftMax
Scale
MatMul
Linear
Linear
Linear
Q
K
V

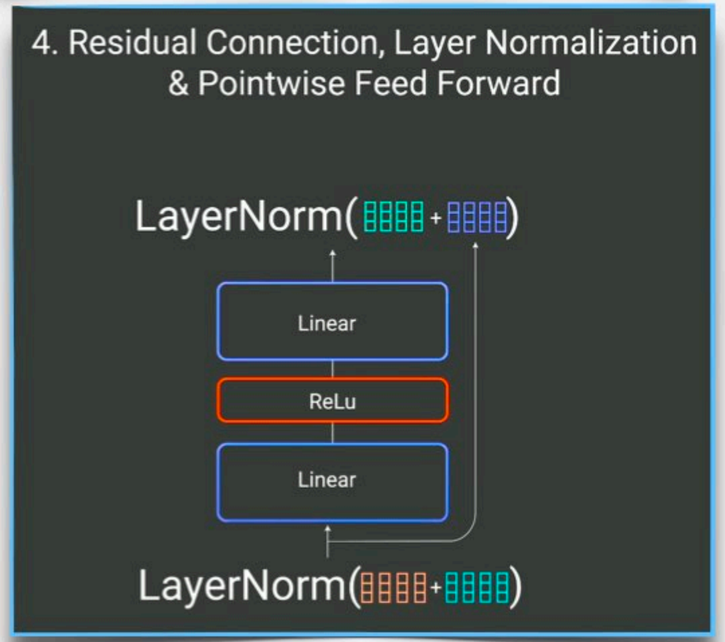

3. Multi-headed Attention
Output Vectors
Multi-headed Attention
Positional Input Embedding

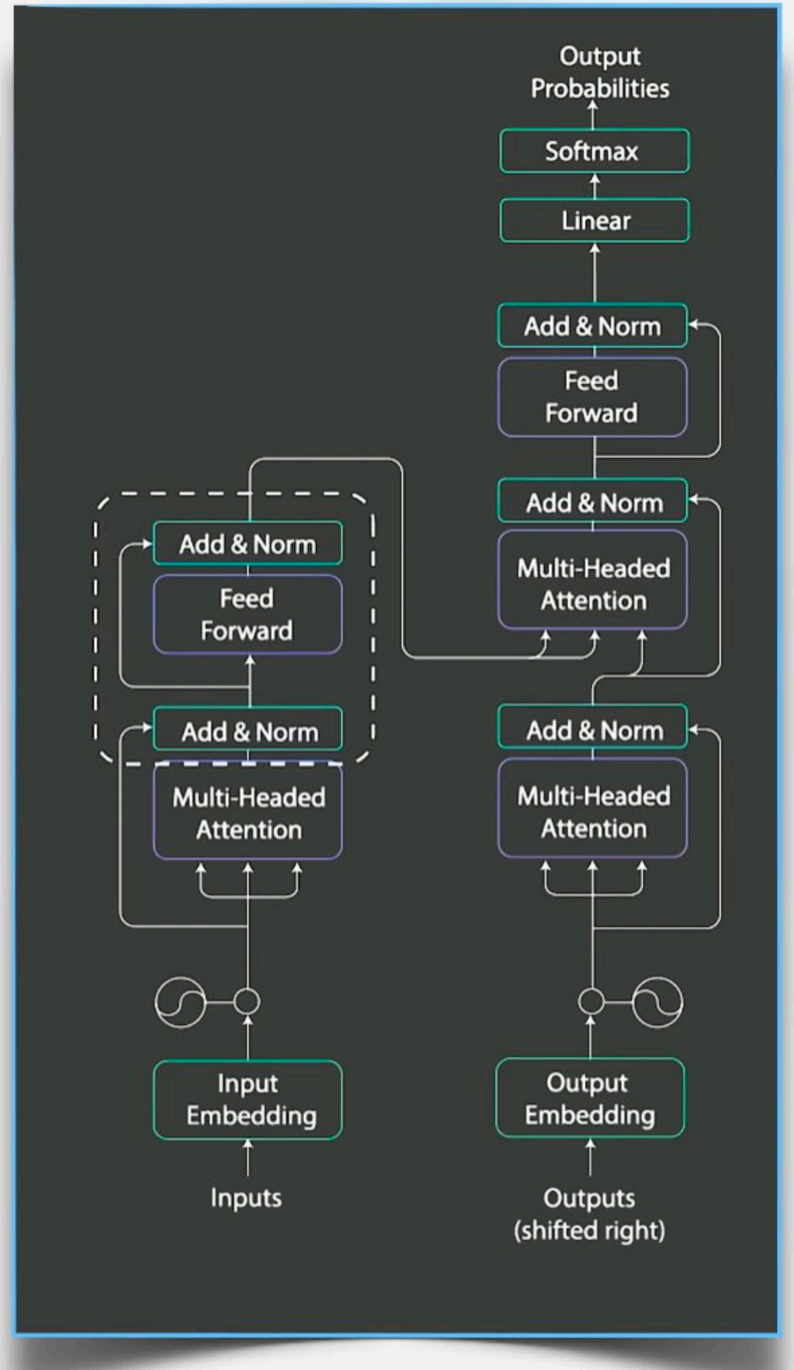

Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Headed Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Headed Attention
Add & Norm
Multi-Headed Attention
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

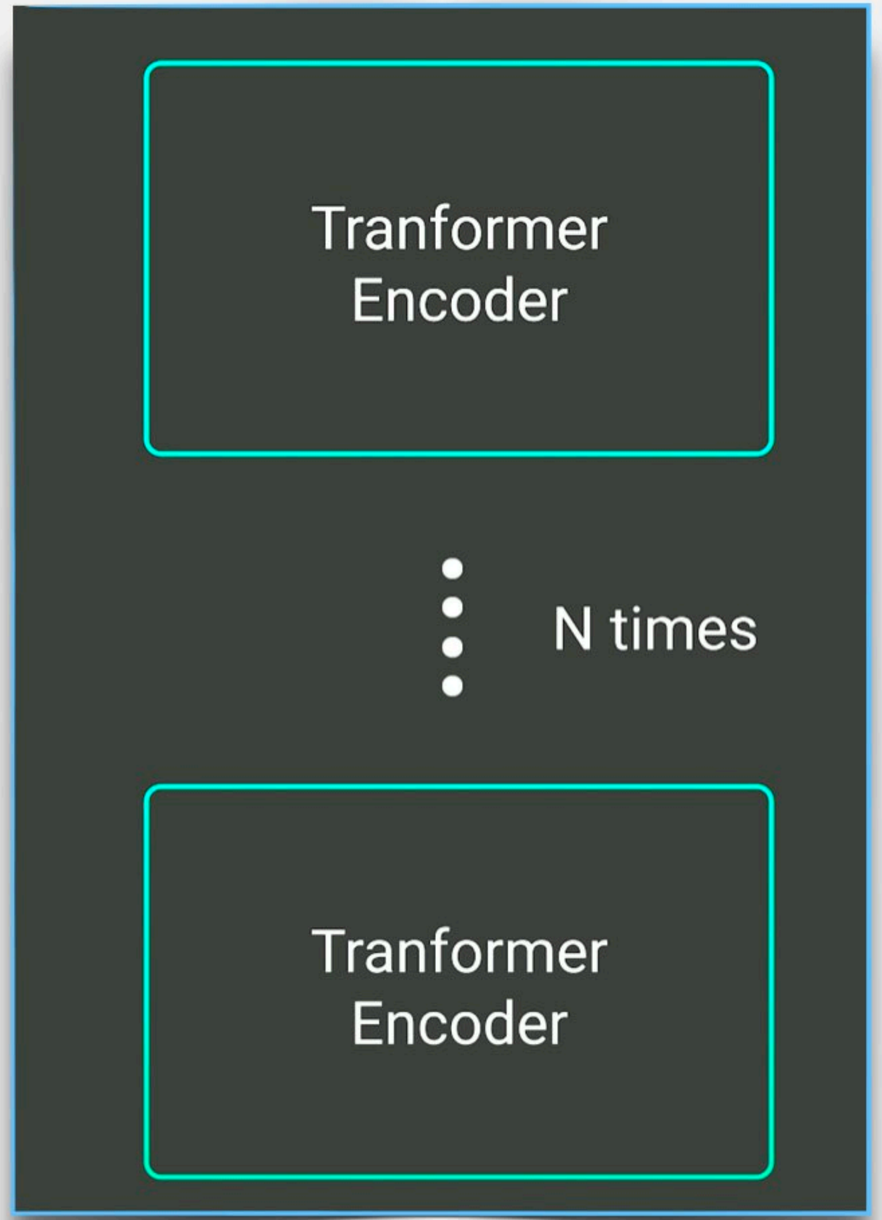

4. Residual Connection, Layer Normalization & Pointwise Feed Forward
Multi-headed Attention

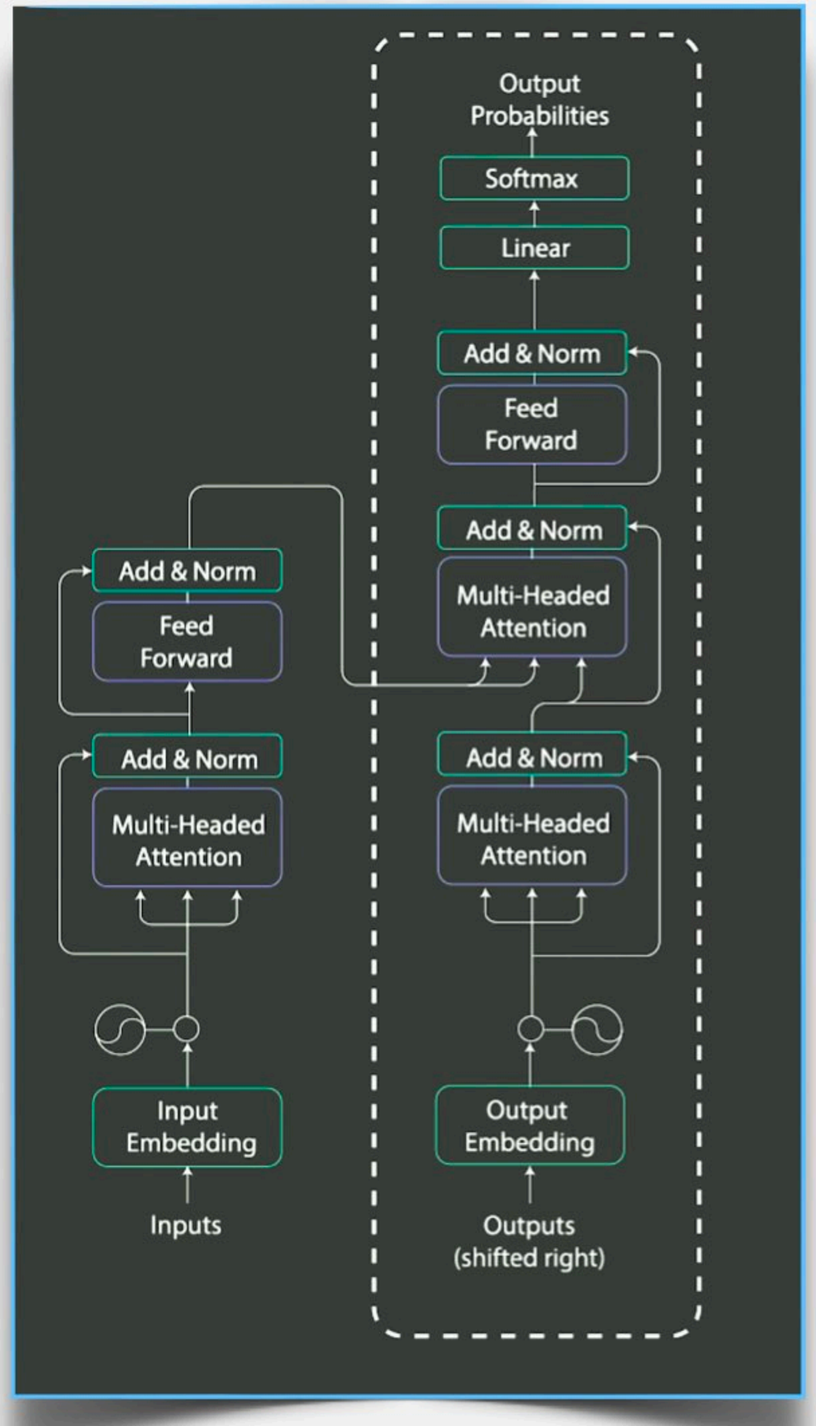

Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Headed Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Headed Attention
Add & Norm
Multi-Headed Attention
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

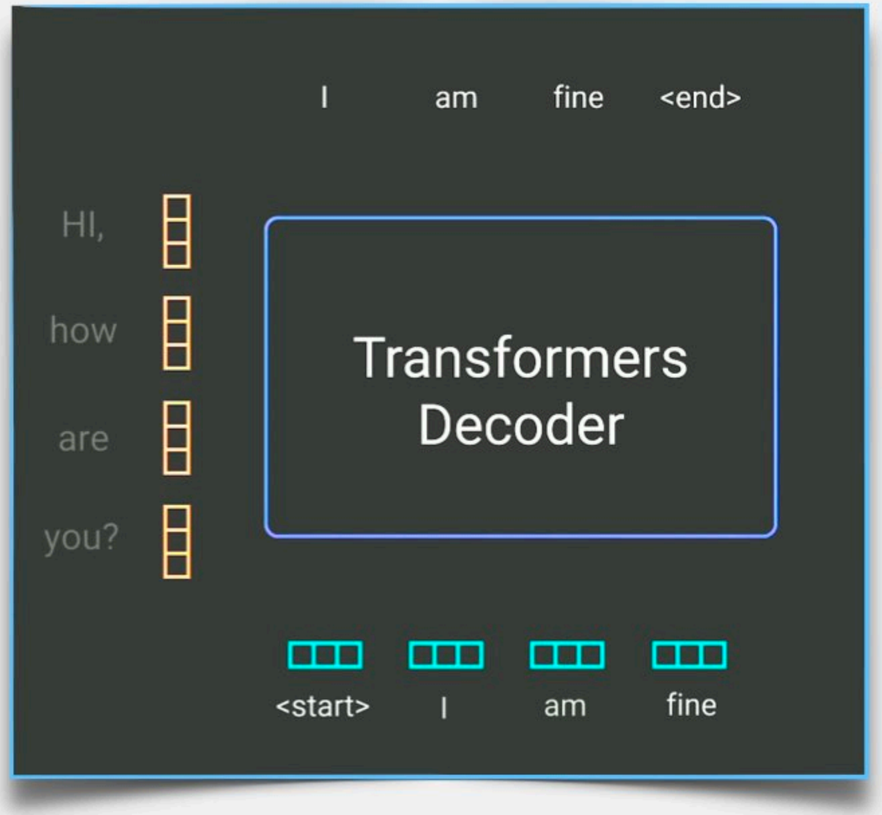

4. Residual Connection, Layer Normalization & Pointwise Feed Forward
Multi-headed Attention


4. Residual Connection, Layer Normalization & Pointwise Feed Forward
LayerNorm(+)
Linear
ReLu
Linear
LayerNorm(+)


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Headed Attention
Add & Norm
Multi-Headed Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Headed Attention
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

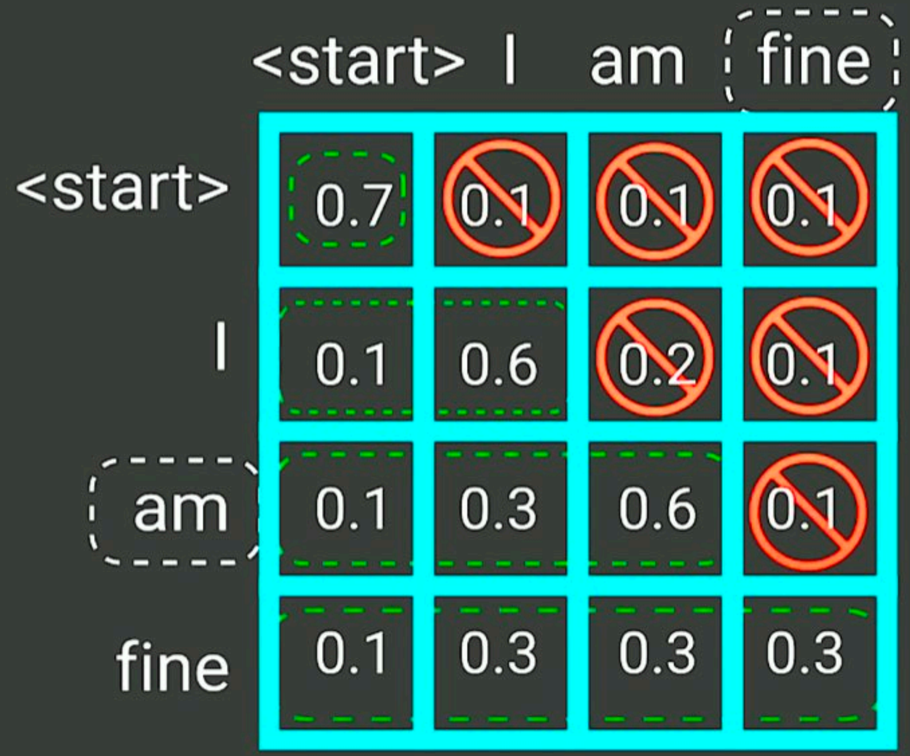

Tranformer Encoder
N times
Tranformer Encoder


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Headed Attention
Add & Norm
Multi-Headed Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Headed Attention
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)


I
am
fine
<end>
HI,
how
are
you?
Transformers Decoder
<start>
I
am
fine

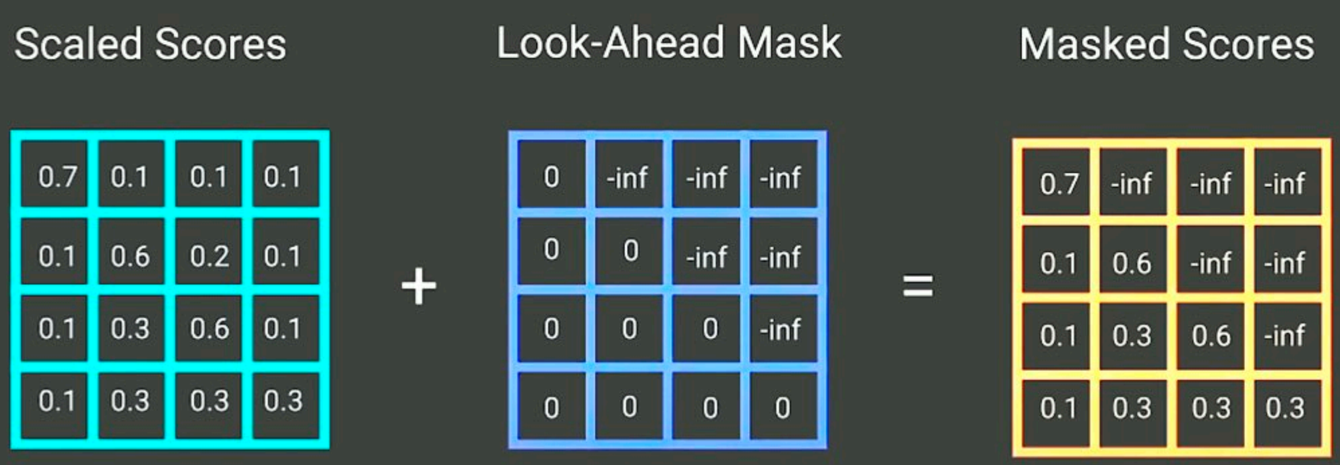
6. Decoder Multi-Headed Attention 1

6. Decoder Multi-Headed Attention 1



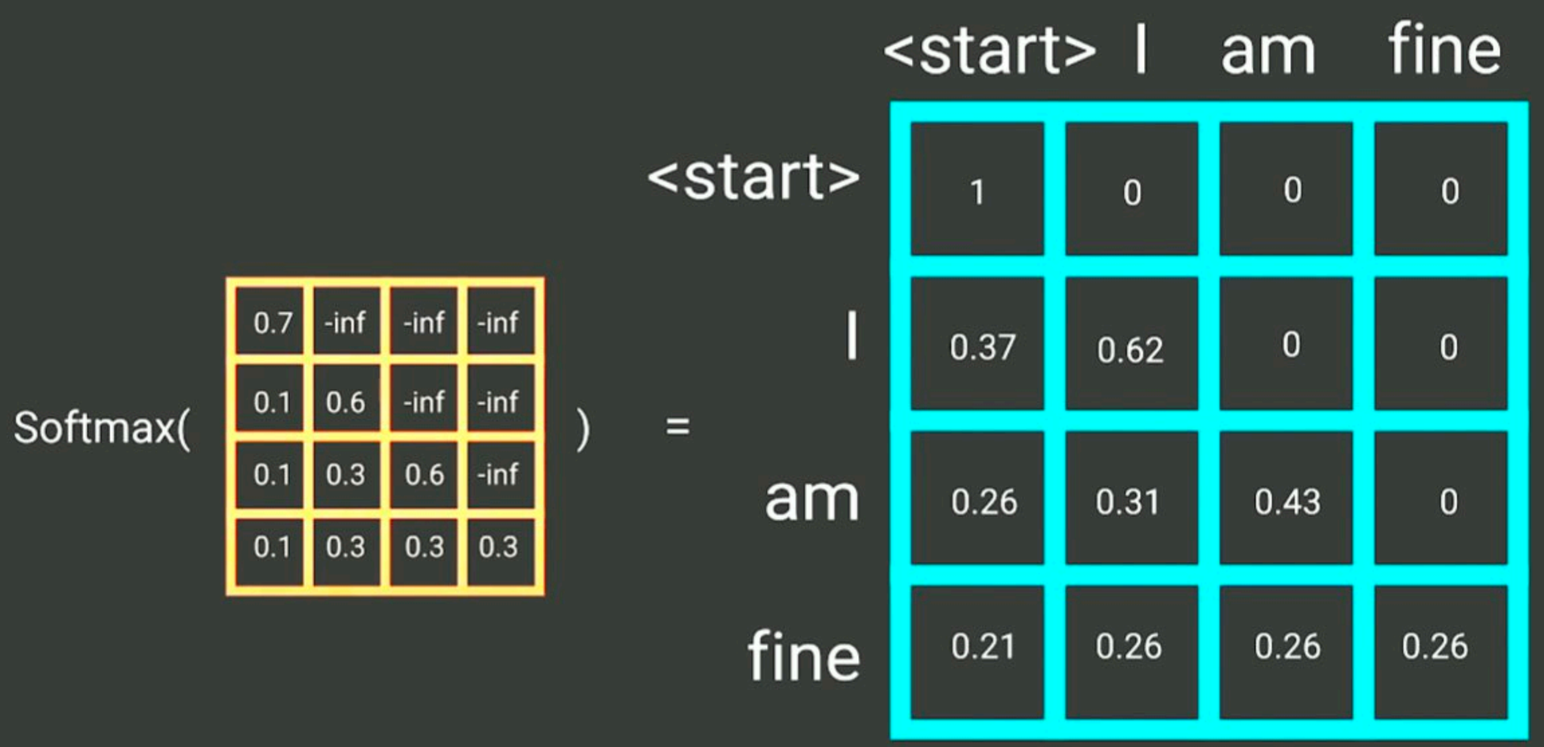
6. Decoder Multi-Headed Attention 1

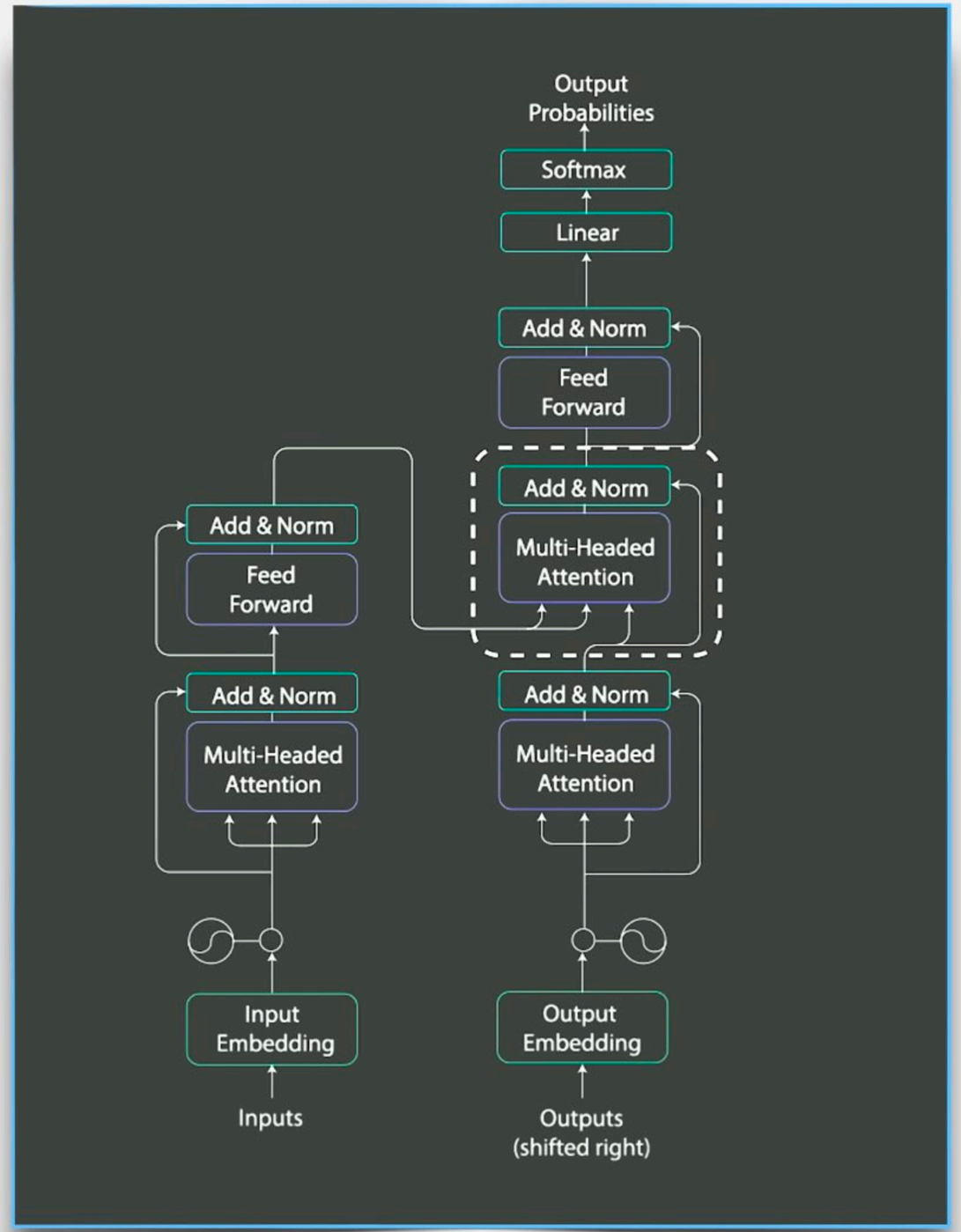
6.1. Look-Ahead Mask

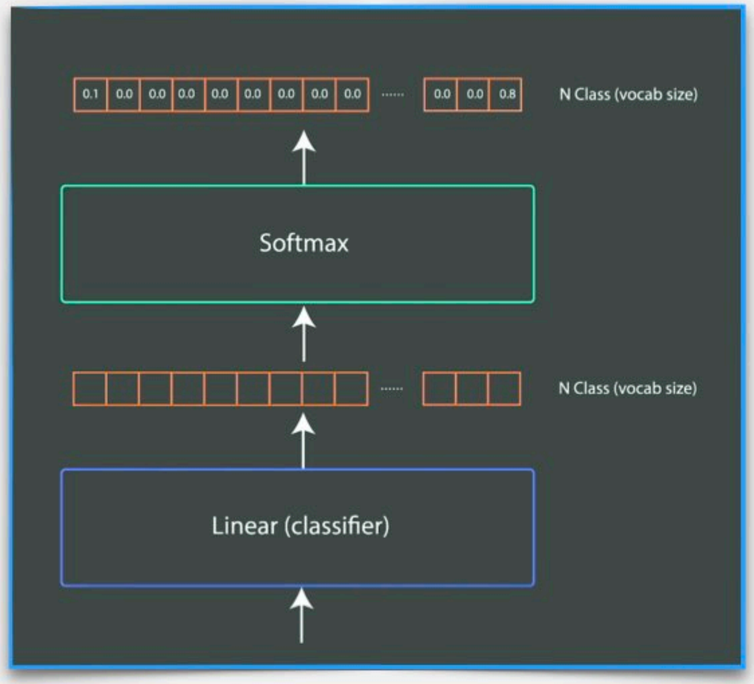


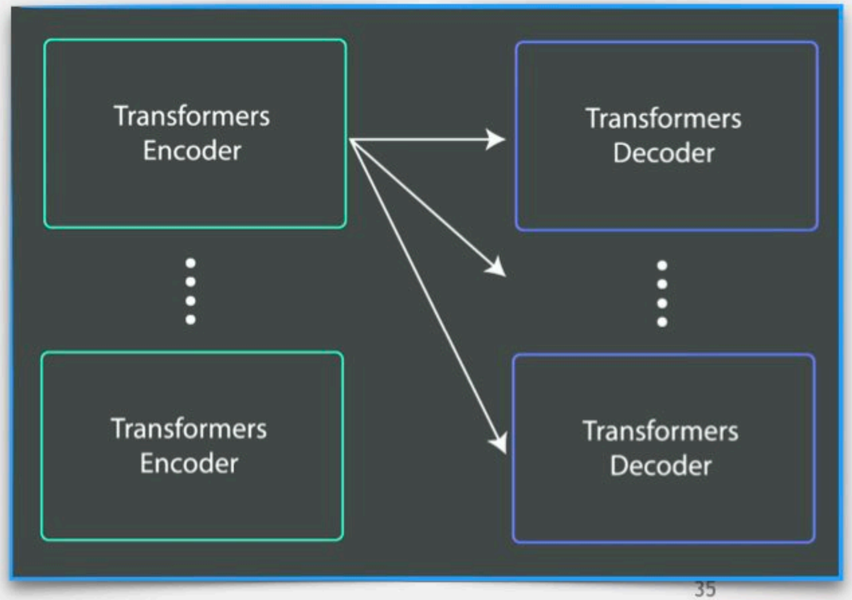
6. Decoder Multi-Headed Attention 1

6.1. Look-Ahead Mask




Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Headed Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Headed Attention
Add & Norm
Multi-Headed Attention
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)


0.1 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ..... 0.0 0.0 0.8
N Class (vocab size)
Softmax
N Class (vocab size)
Linear (classifier)


Transformers Encoder
Transformers Decoder
Transformers Encoder
Transformers Decoder

Positional Encoding

what we see:

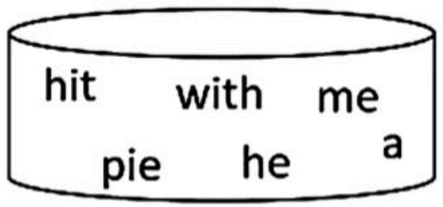
he hit me with a pie

what naïve self-attention sees:

a pie hit me with he

a hit with me he pie

he pie me with a hit



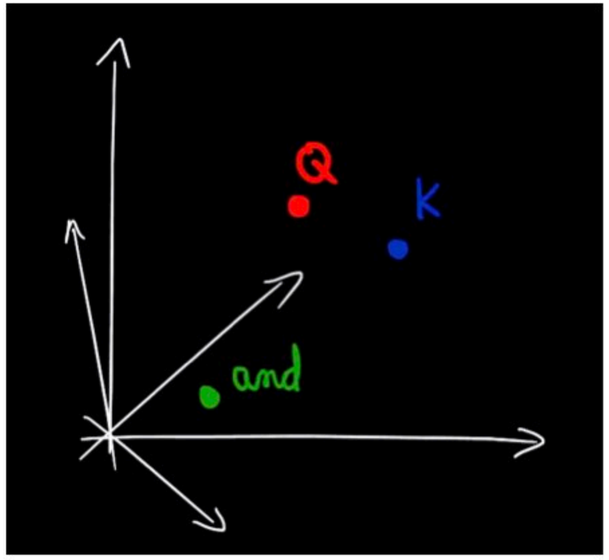
most alternative orderings are nonsense, but some change the meaning

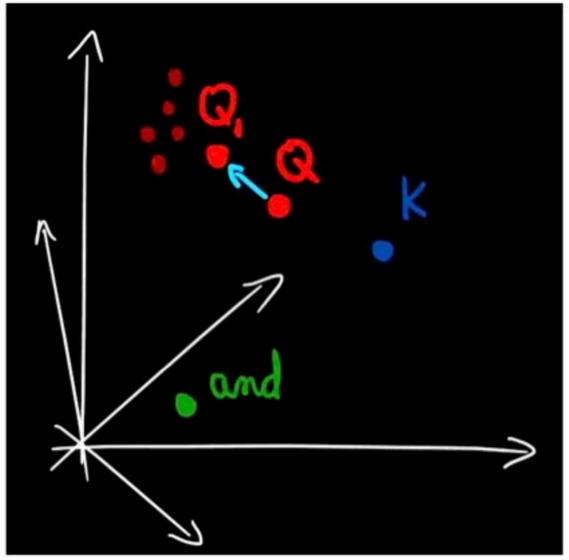
in general the position of words in a sentence carries information!

Idea: add some information to the representation at the beginning that indicates where it is in the sequence!

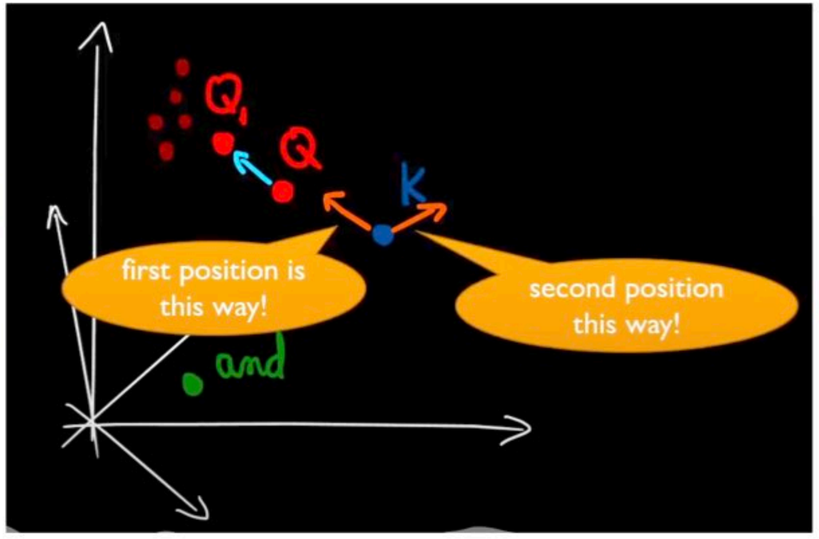
Positional Encoding

Naïve positional encoding:





0.33	0.71	0.91	0.23	0.15
0.01	0.01	0.01	0.01	0.01



I walk my dog every day

every single day I walk my dog

The fact that "my dog" is right after "I walk" is the important part, not its absolute position

Positional Encoding

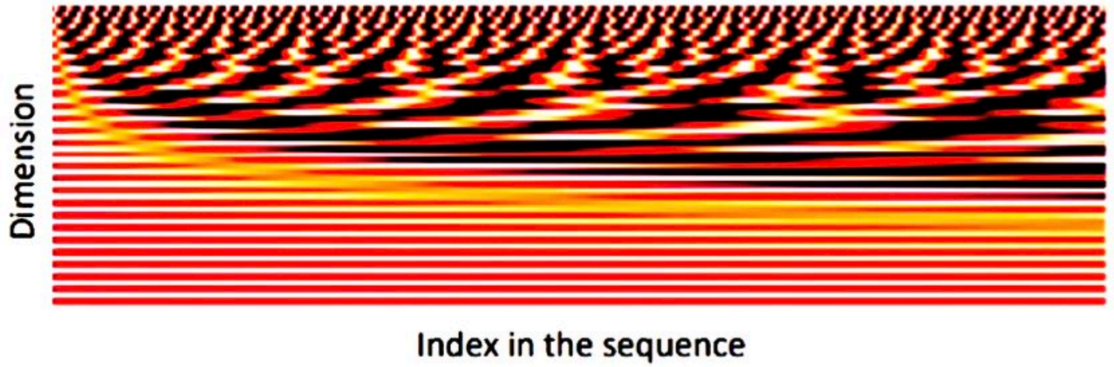
- Positional embeddings should be same irrespective of sequence length or content
- Shift in the embeddings space should not be large
- Ability to estimate distance between two tokens
- Deterministic based on some underlying rule

Positional Encoding

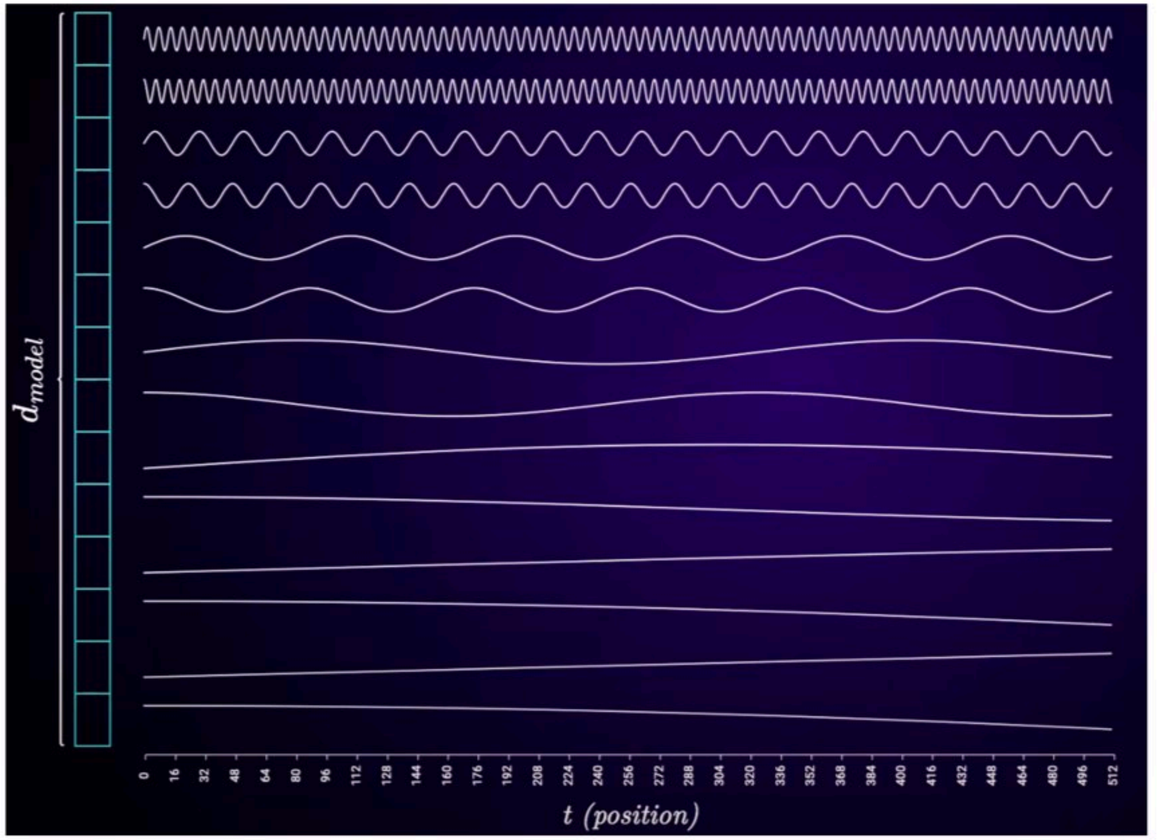
Idea: what if we use **frequency-based** representations?

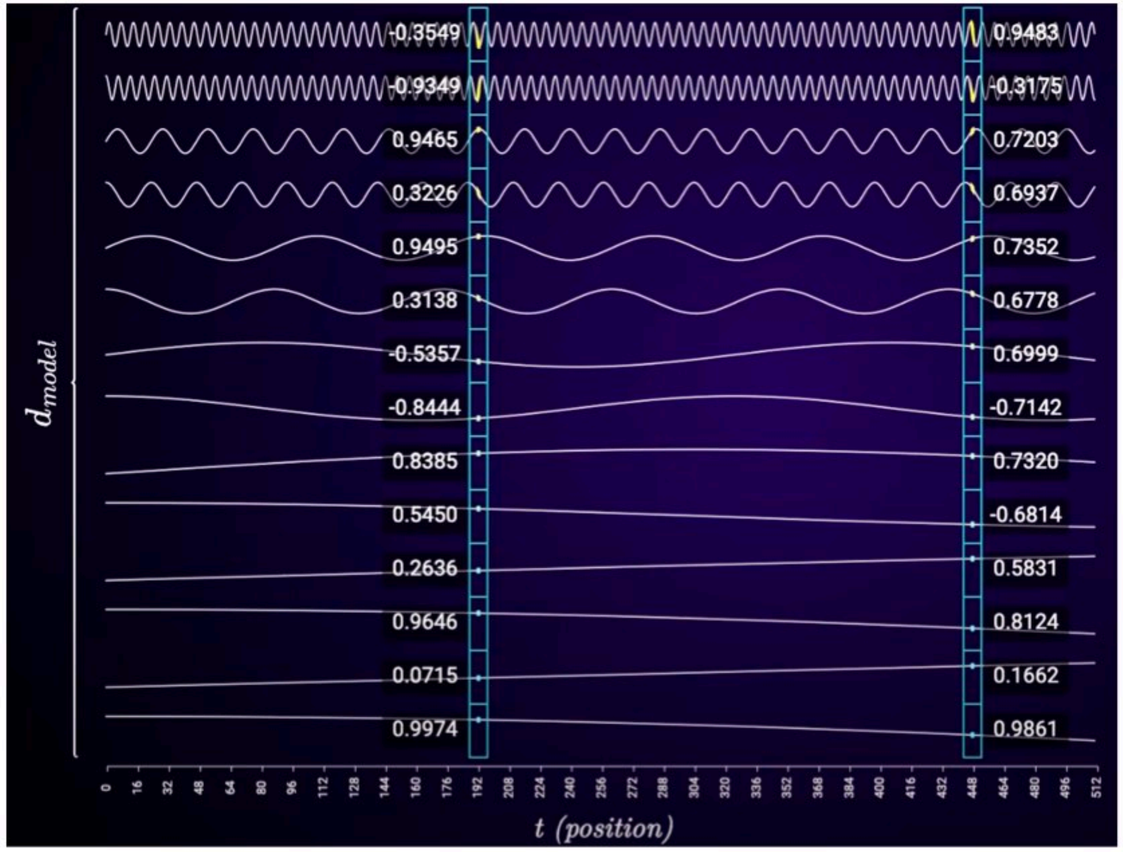
$$p_t = \begin{bmatrix} \sin(t/10000^{2*1/d}) \\ \cos(t/10000^{2*1/d}) \\ \sin(t/10000^{2*2/d}) \\ \cos(t/10000^{2*2/d}) \\ \cdots \\ \sin(t/10000^{2*\frac{d}{2}/d}) \\ \cos(t/10000^{2*\frac{d}{2}/d}) \end{bmatrix}$$

dimensionality of positional encoding

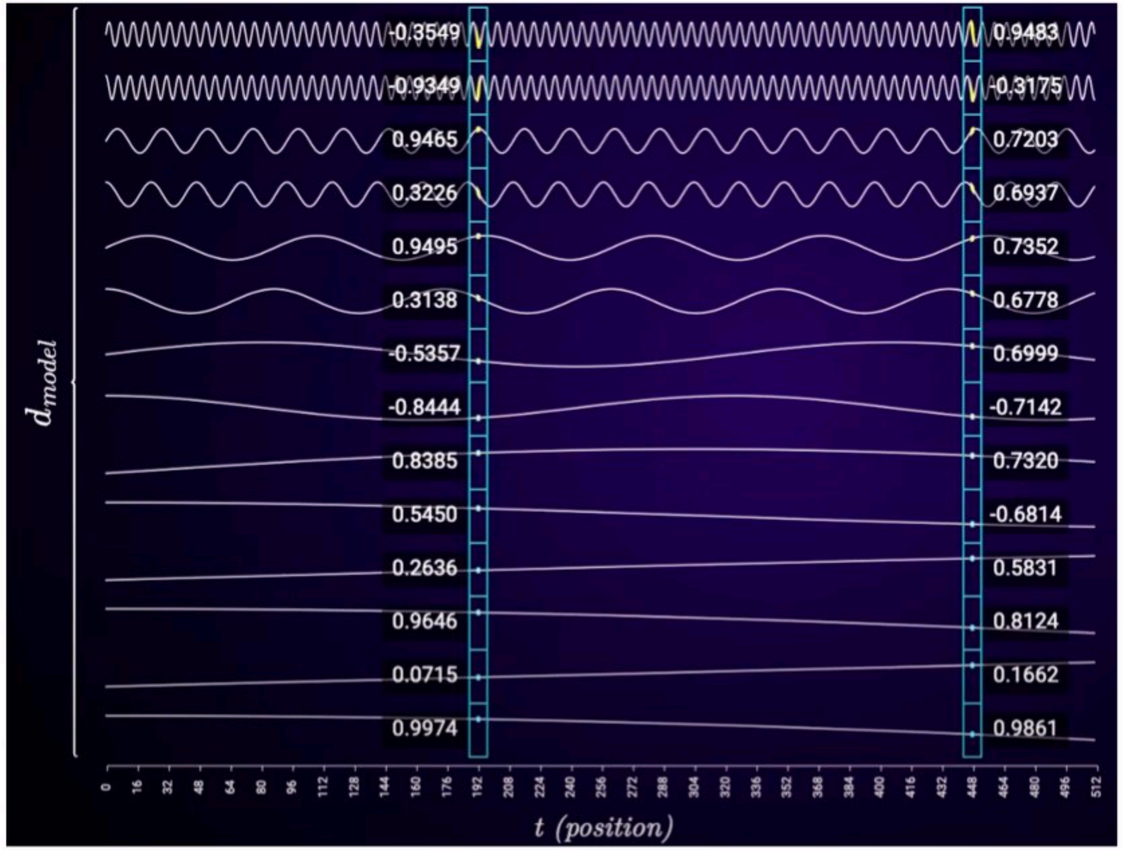


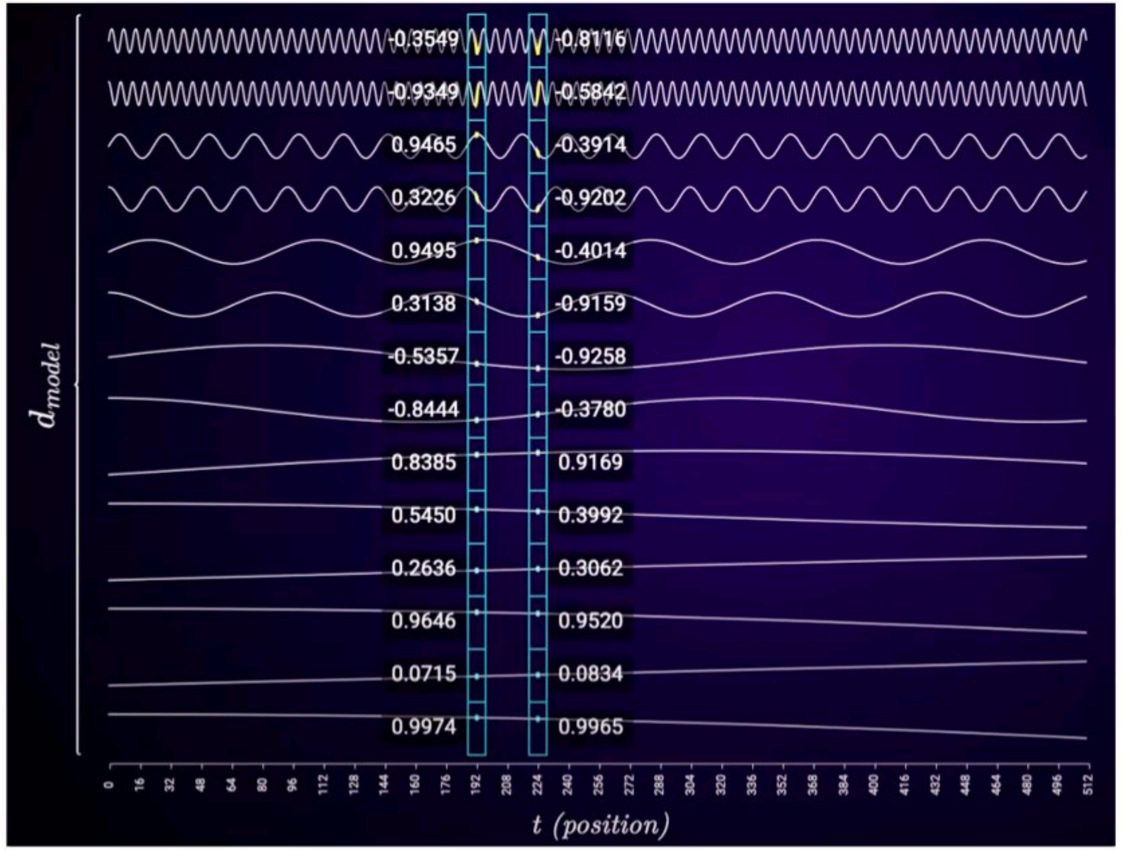
Positional Encoding



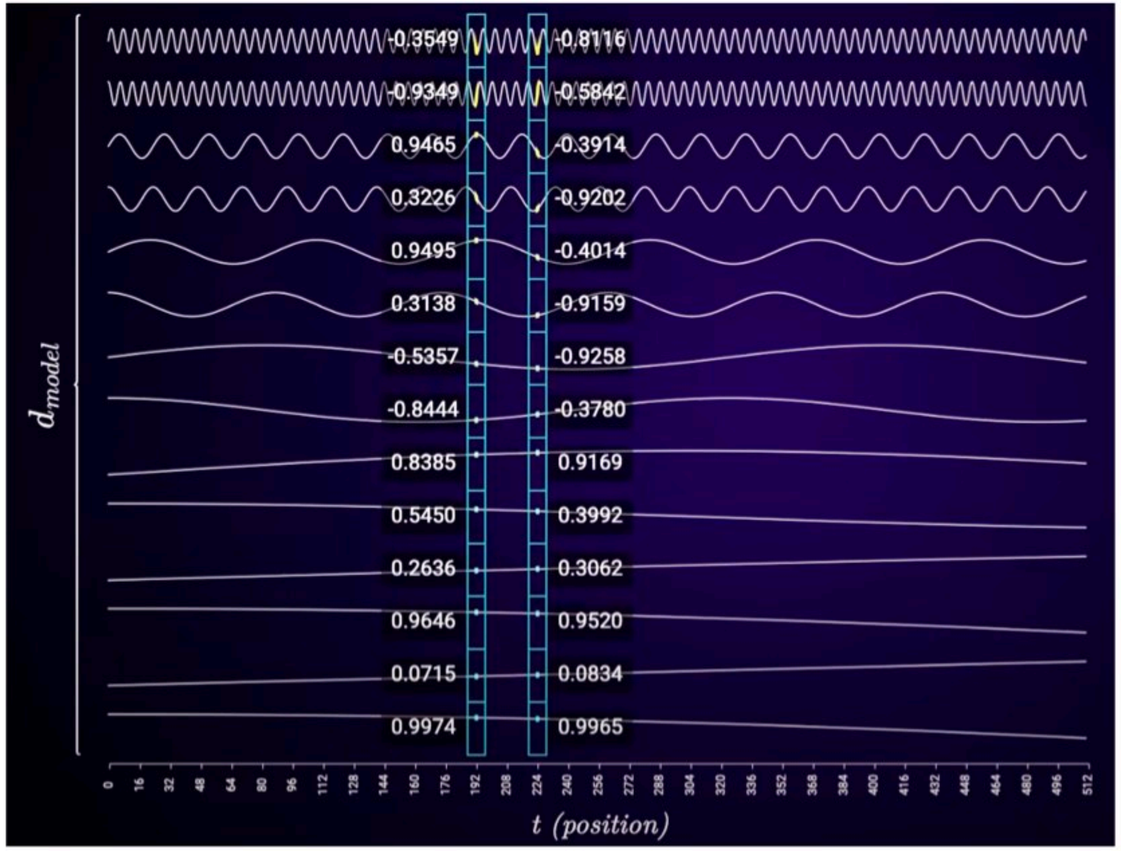


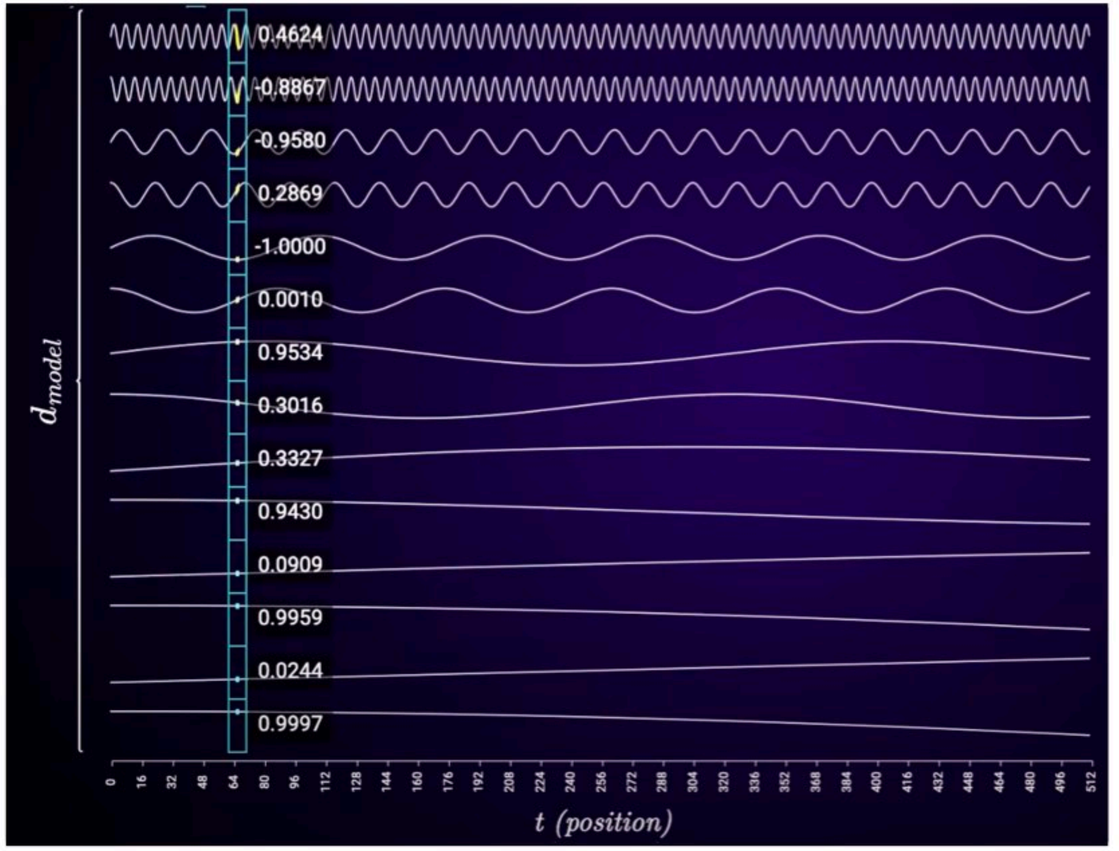
Positional Encoding



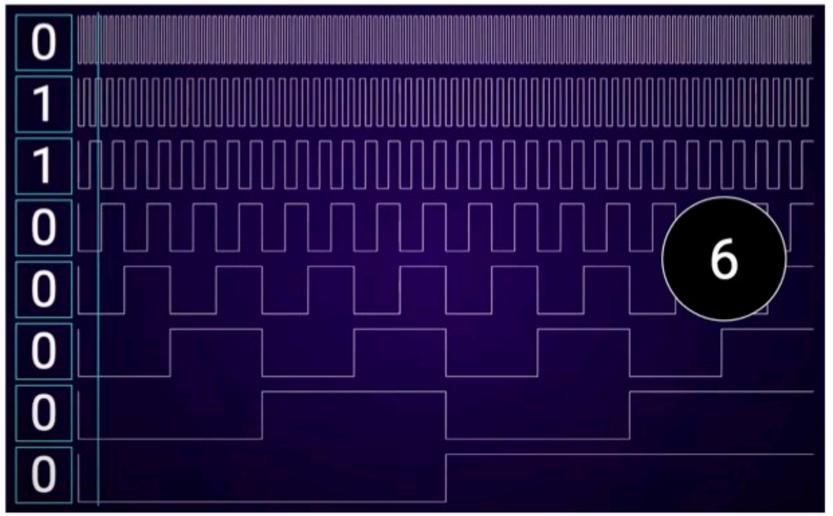


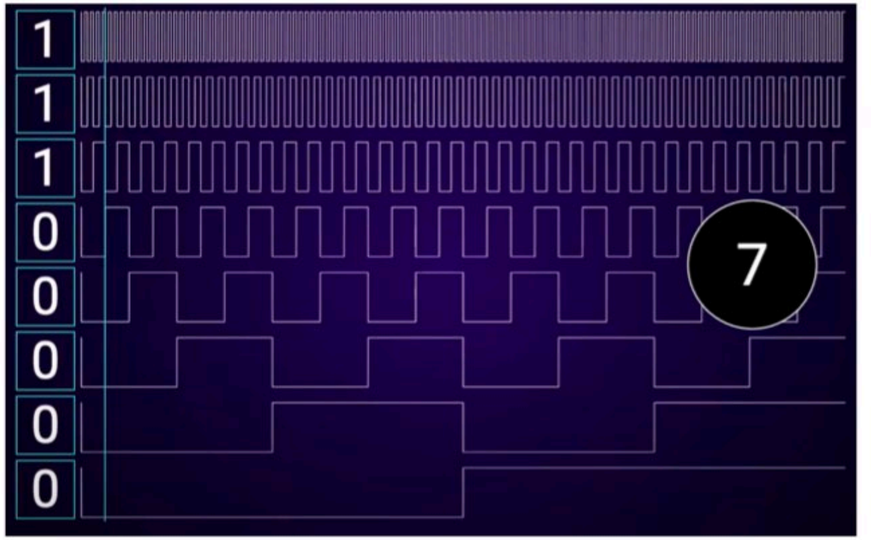
Positional Encoding

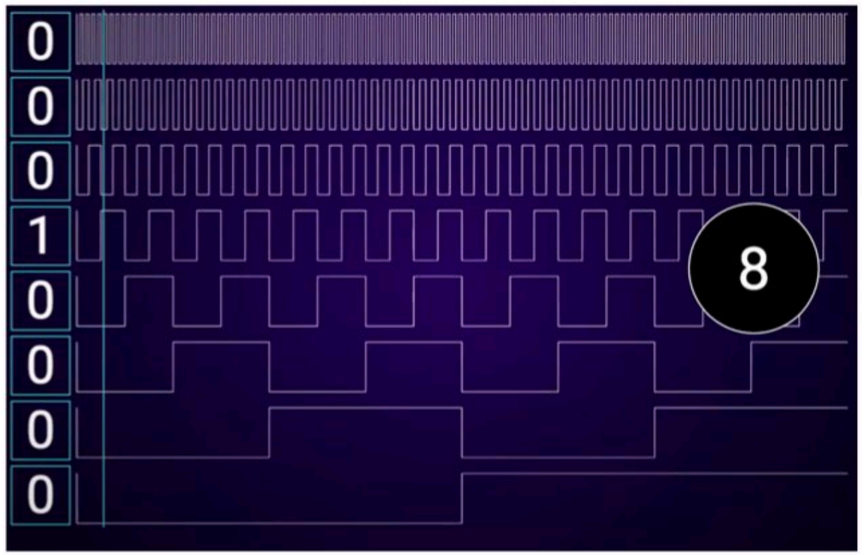




Positional Encoding

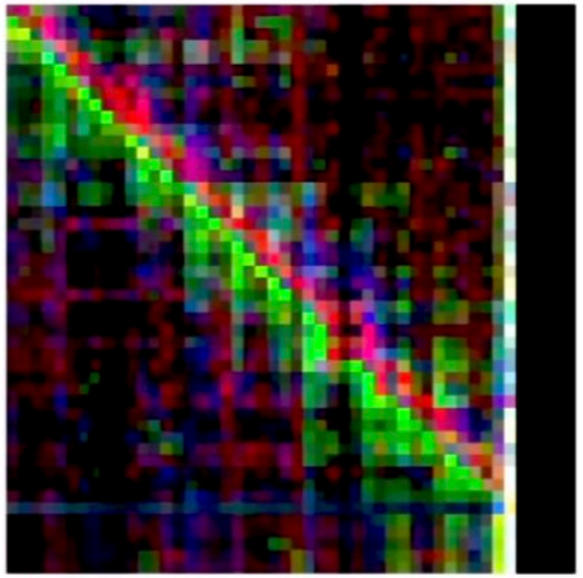




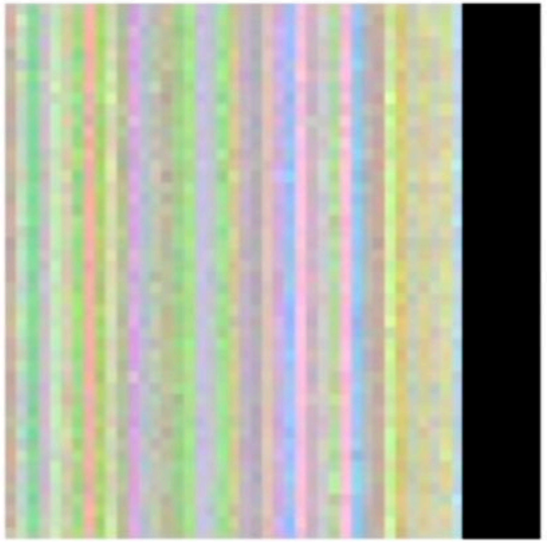


Importance of Residual Connection

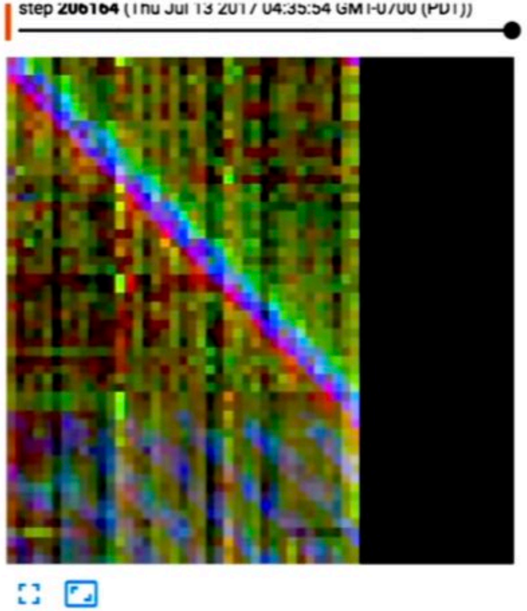
Residuals carry positional information to higher layers, among other information.



With residuals

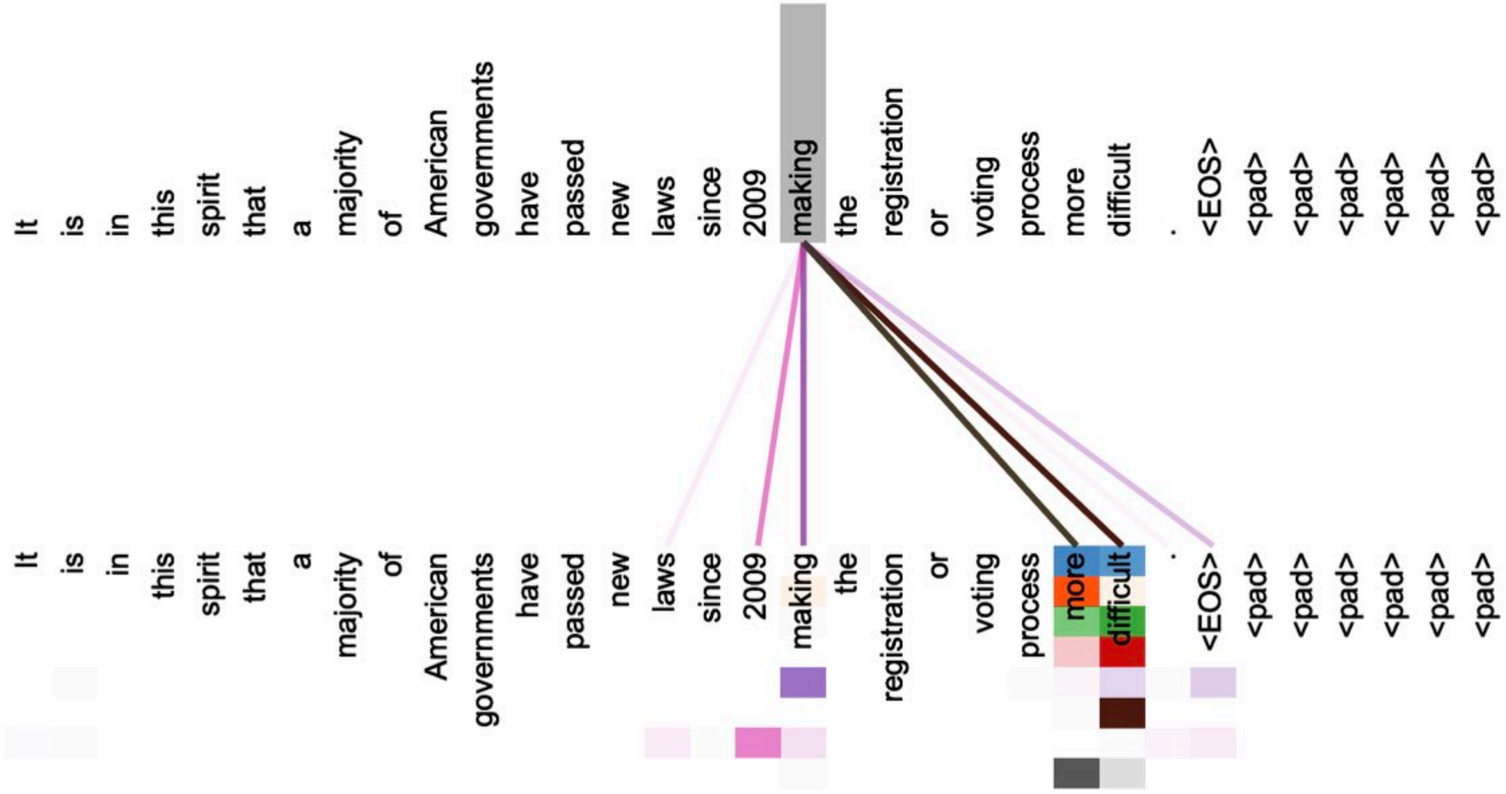


Without residuals

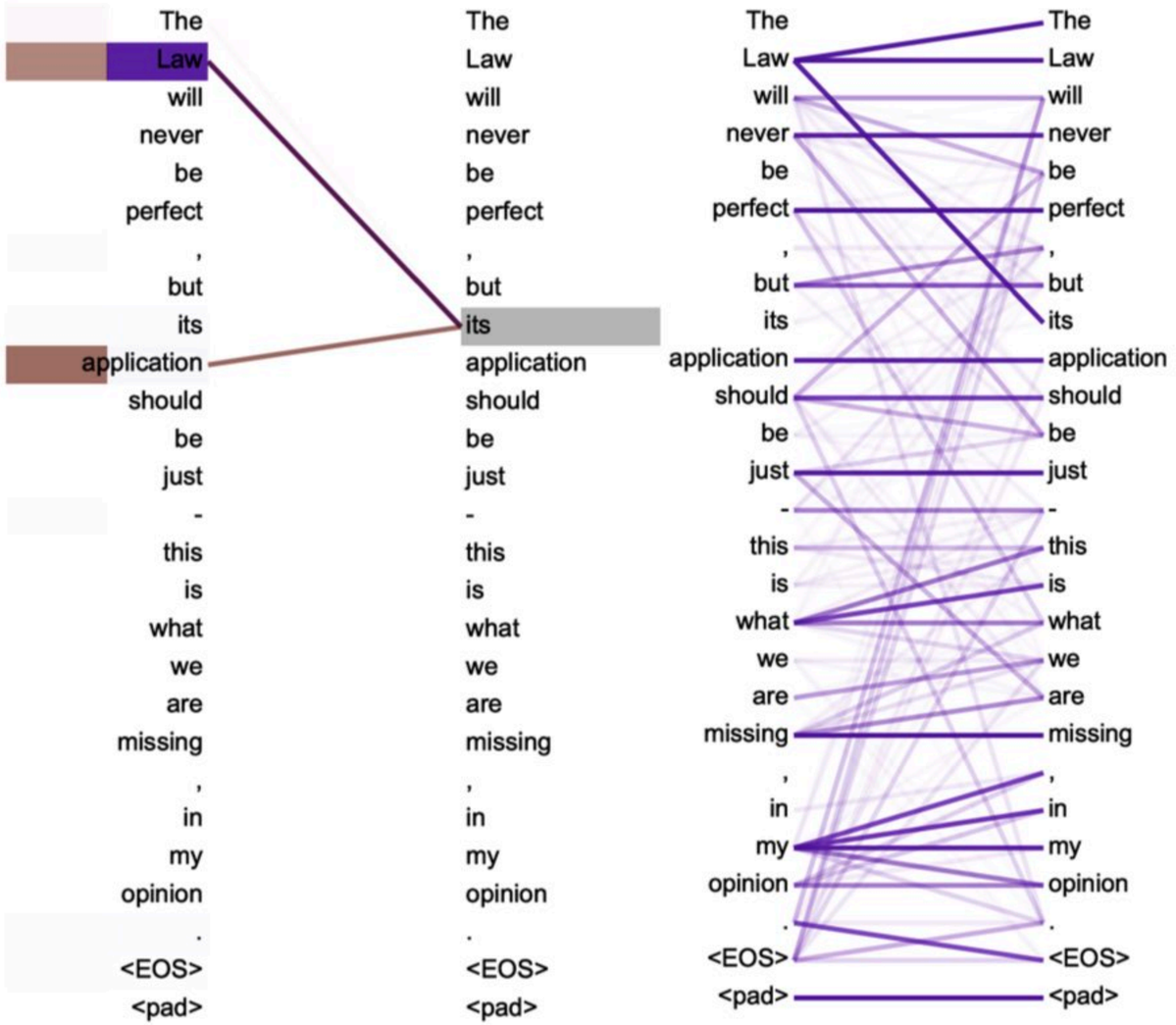


Without residuals,
with timing signals

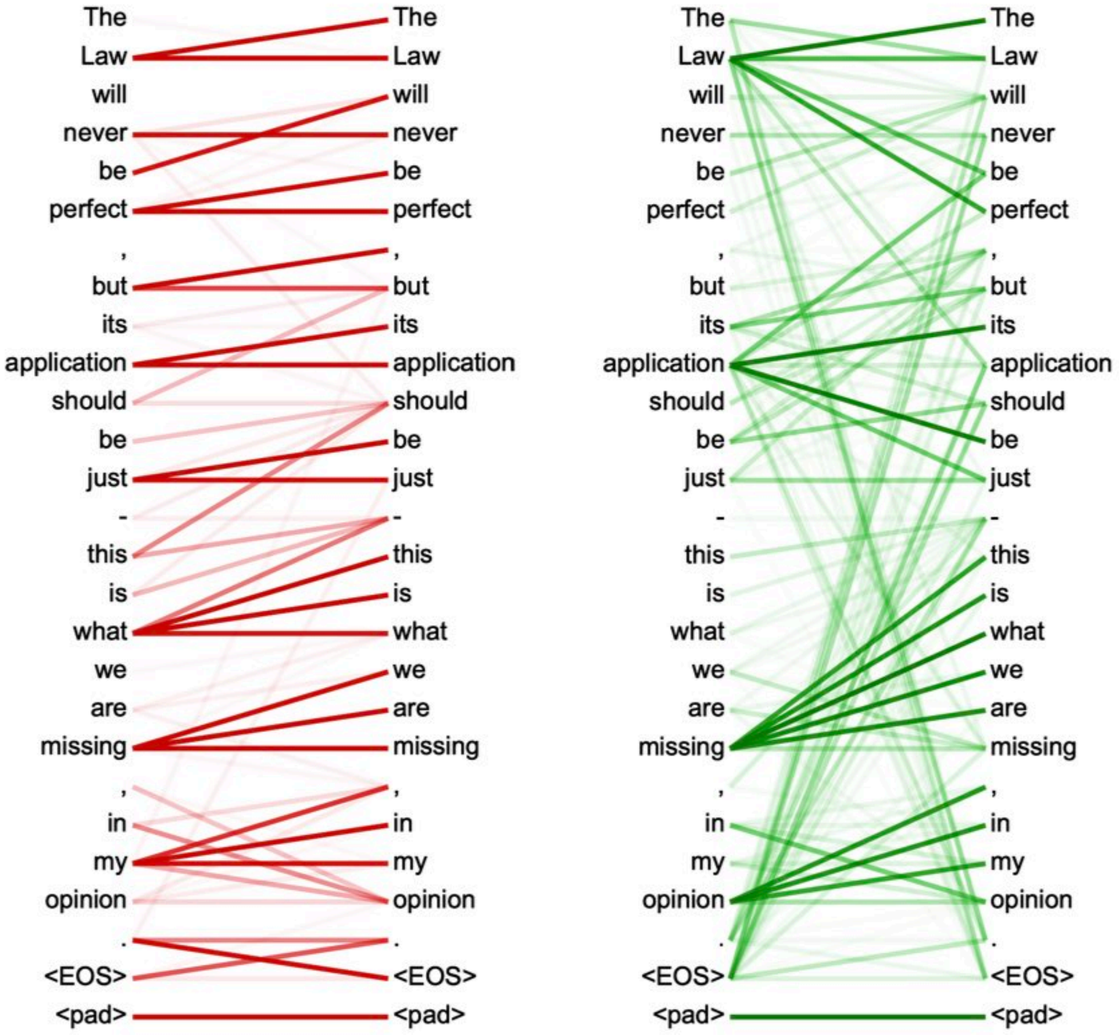
Multi-head Attention intuition



Multi-head Attention intuition



Multi-head Attention intuition



Limitations of Transformers

- Computationally Expensive
- Lack of interpretability
- Limited sequence length