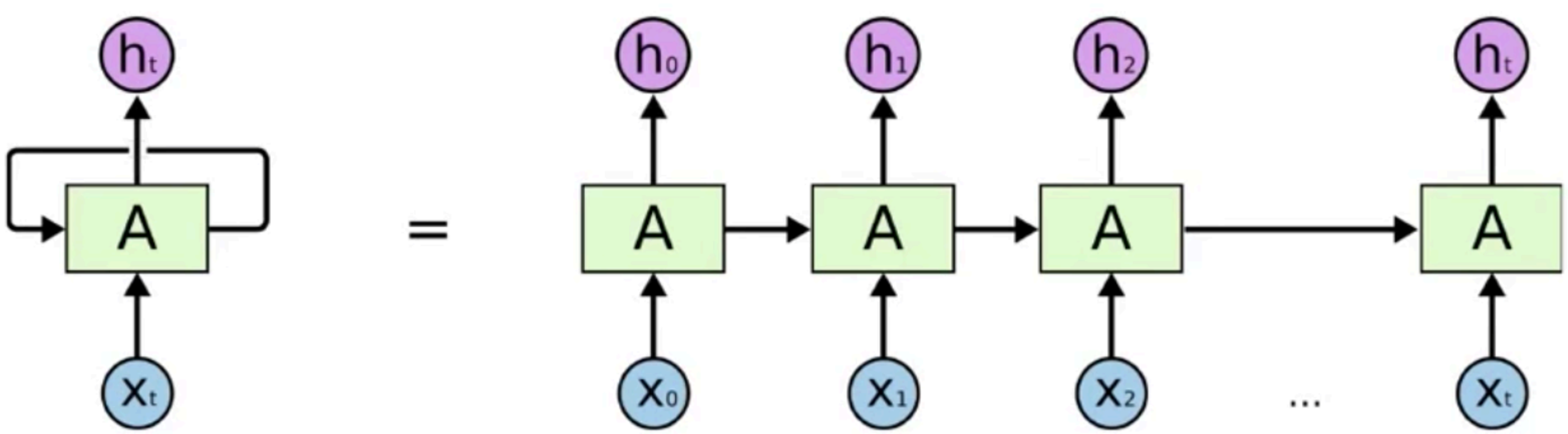
Attention Is All You Need

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones,
Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin
From: Google brain Google research

Presented by: Hsuan-Yu Chen

RNN

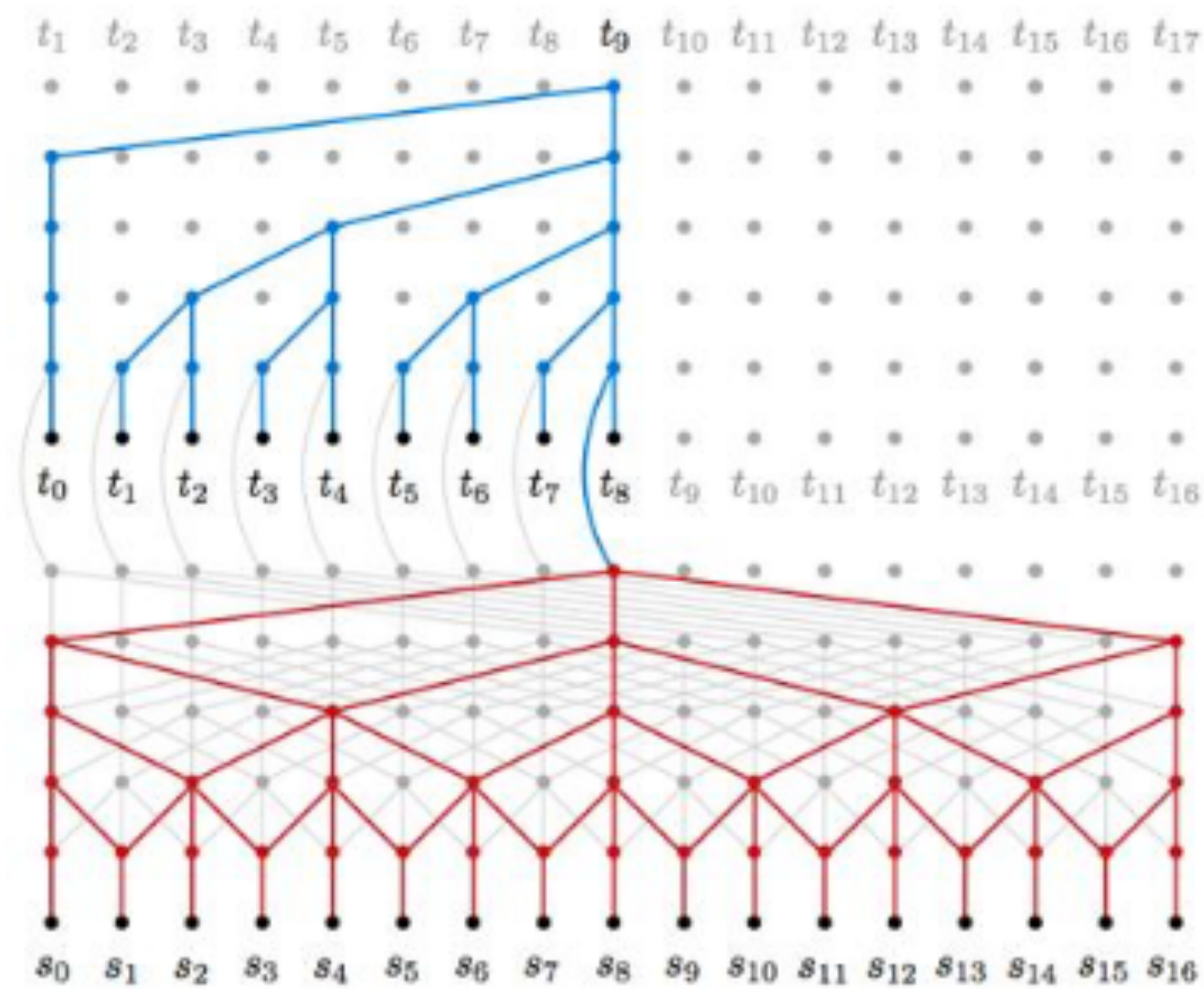
- Advantages:
  - State-of-the-art for variable-length representations such as sequences
  - RNN are considered core of Seq2Seq (with attention)
- Problems:
  - Sequential process prohibits parallelization. Long range dependencies
  - Sequences-aligned states: hard to model hierarchical-alike domains ex. languages



CNN

- Better than RNN (Linear): path length between positions can be logarithmic when using dilated convolutions

- Drawback: require a lot of layers to catch long-term dependencies



Attention and Self-Attention

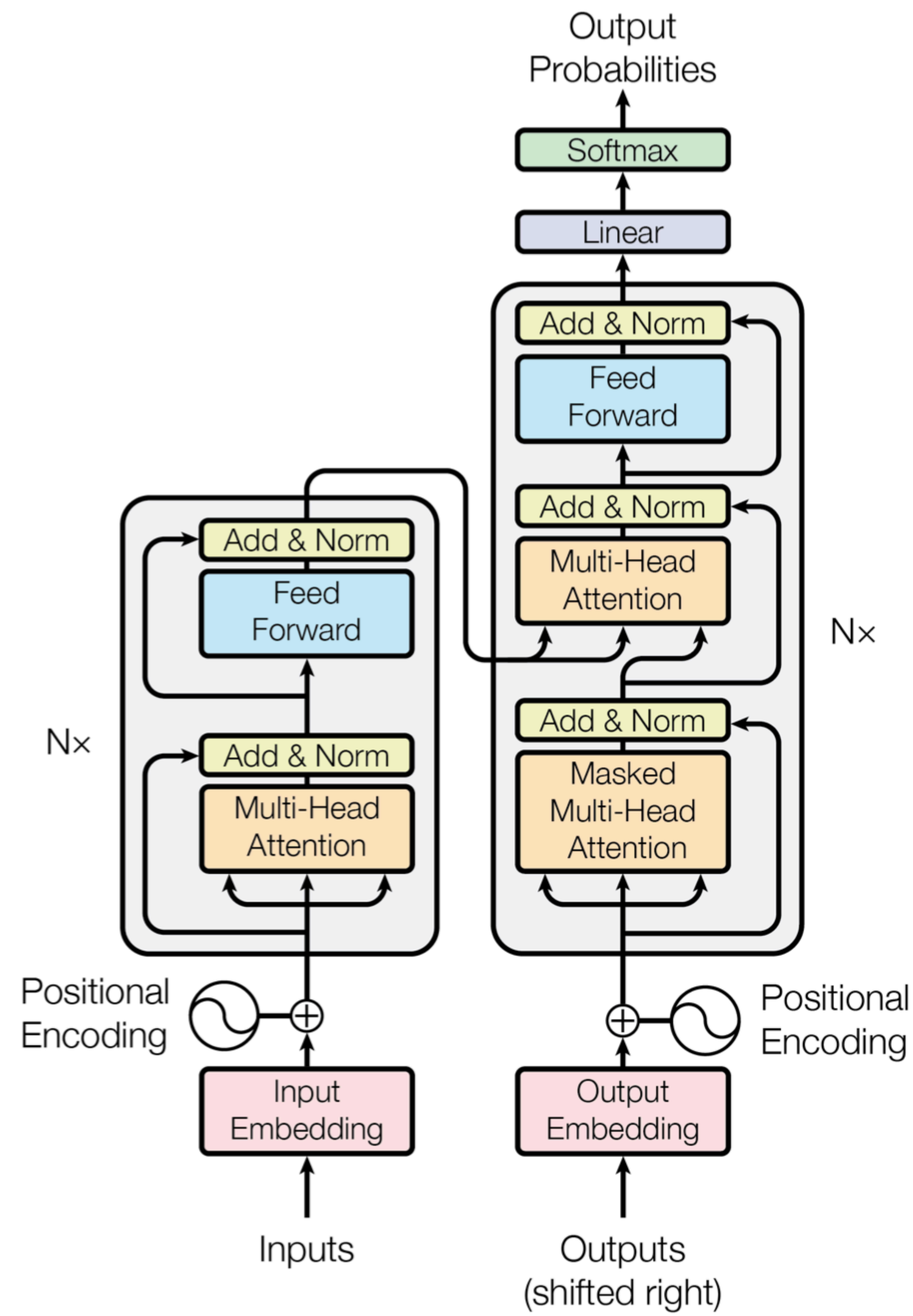
- Attention: $\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$
  - Removes bottleneck of Encoder-Decoder model
  - Focus on important parts
- Self-Attention:
  - all the variables (queries, keys and values) come from the same sequence

Why Self Attention

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

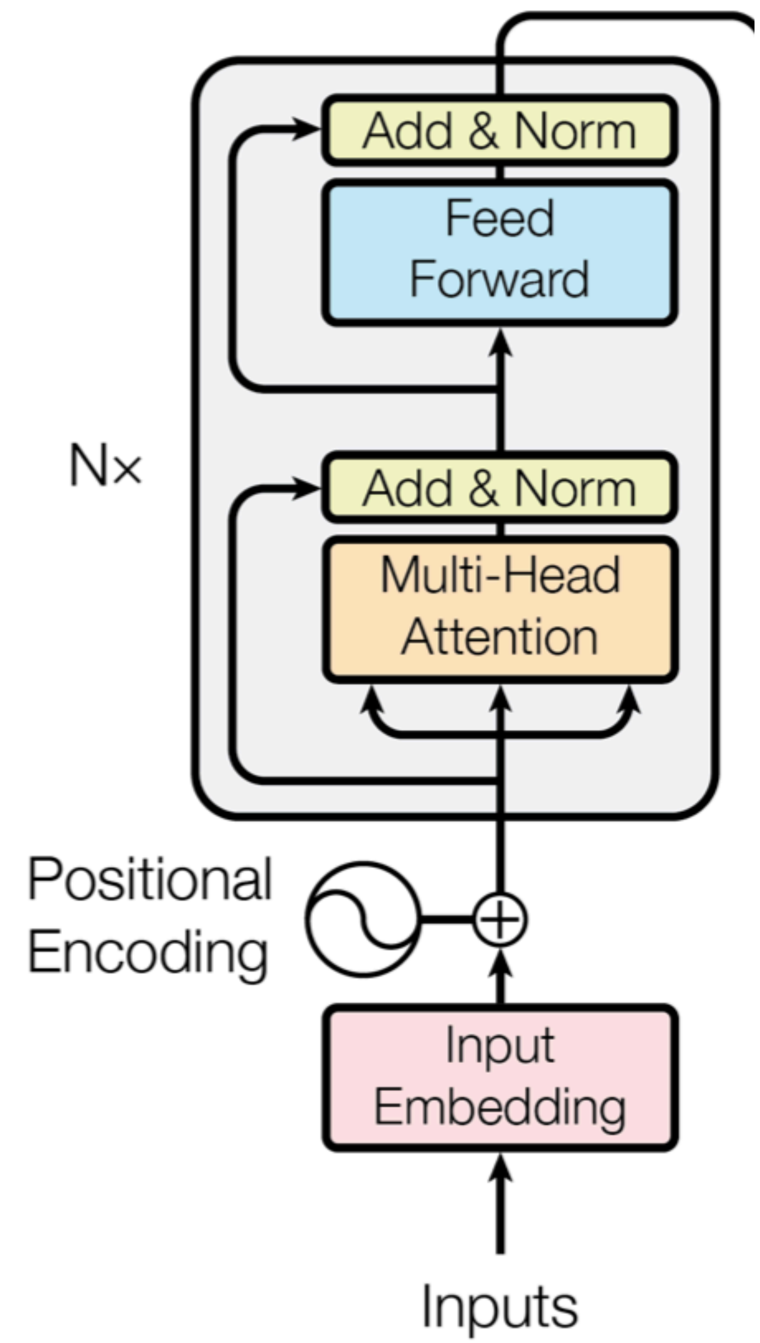
Transformer Architecture

- Encoder: 6 layers of self-attention + feed-forward network
- Decoder: 6 layers of masked self-attention and output of encoder + feed-forward



Encoder

- N = 6
- All layer output size 512
- Embedding
- Positional Encoding
- Multi-head Attention
- Residual Connection
- Position wise feed forward

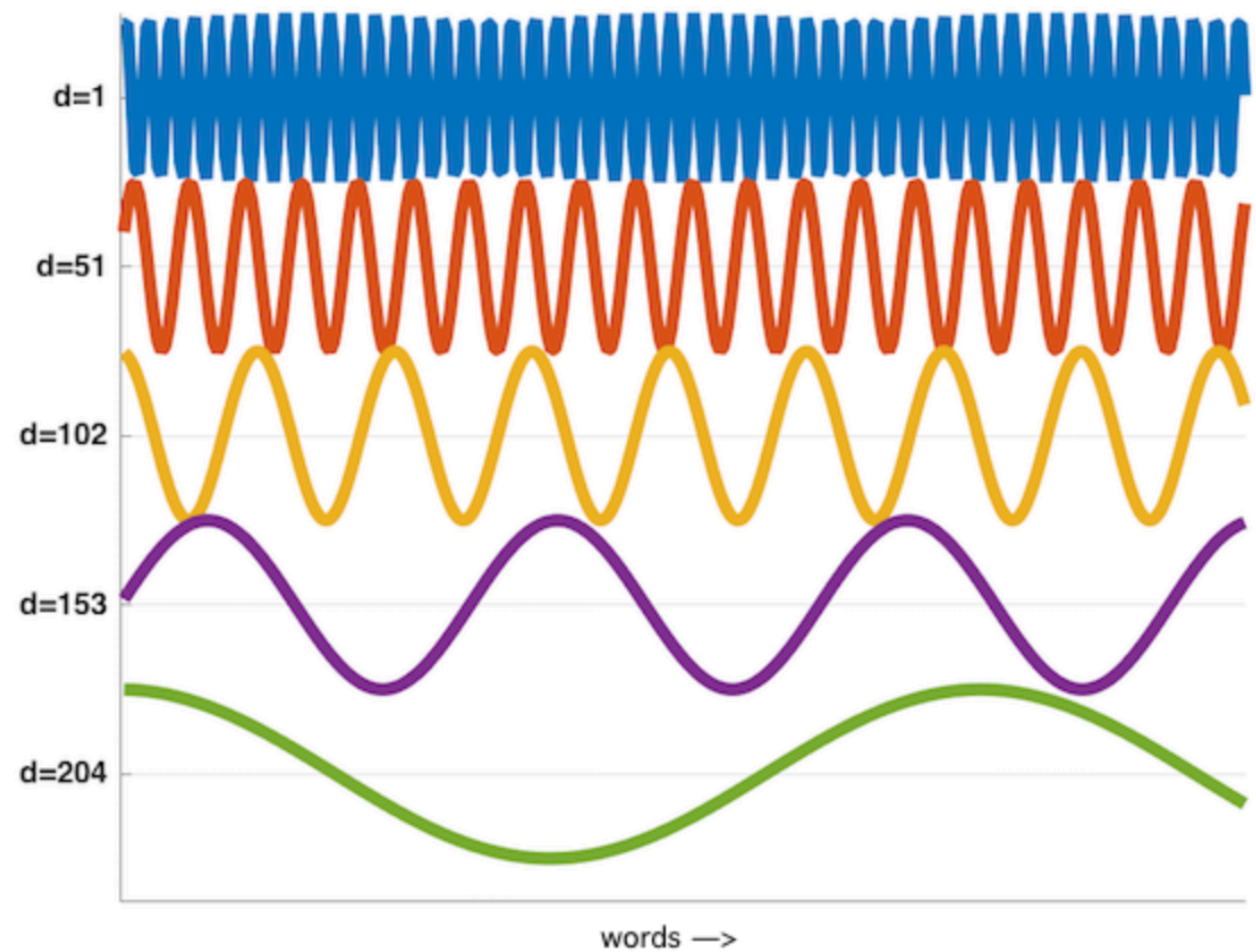


Positional Encoding

- Positional encoding provides relative or absolute position of given token

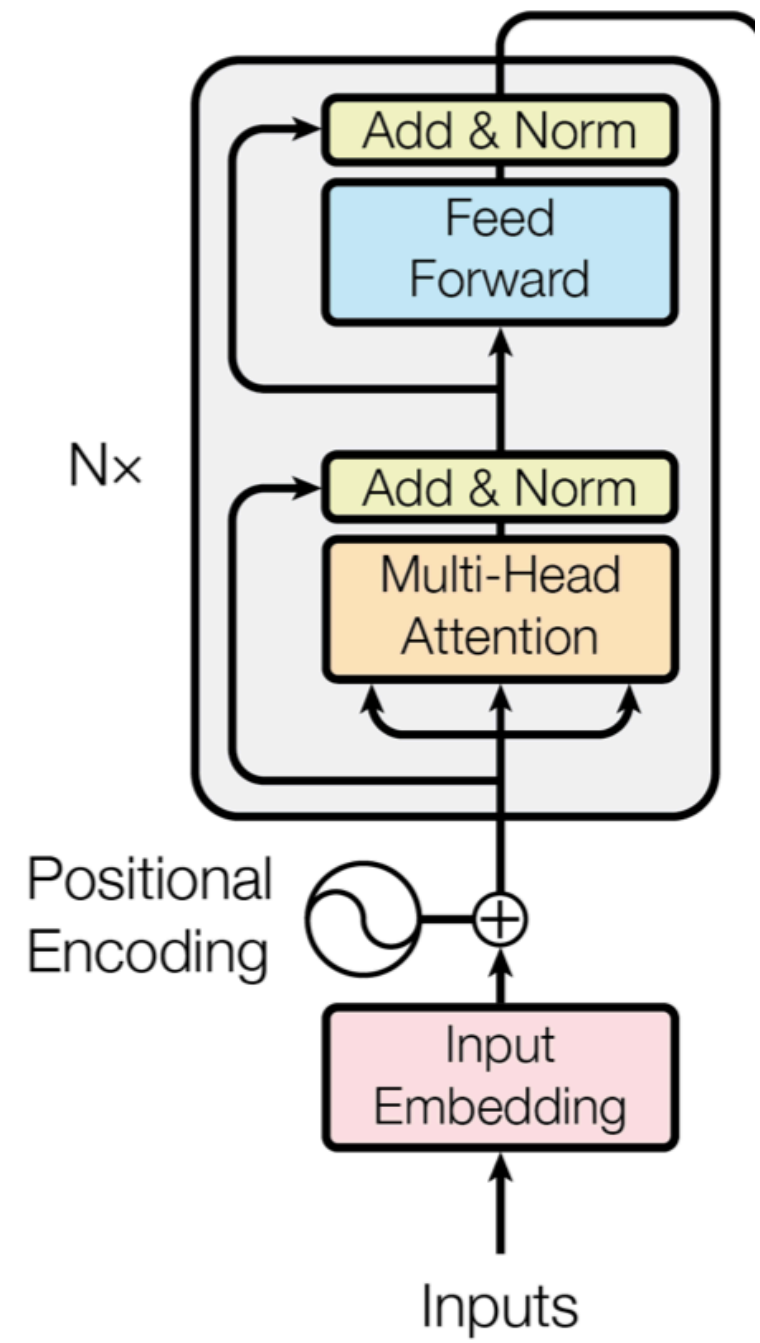
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

- where pos is the position and i is the dimension



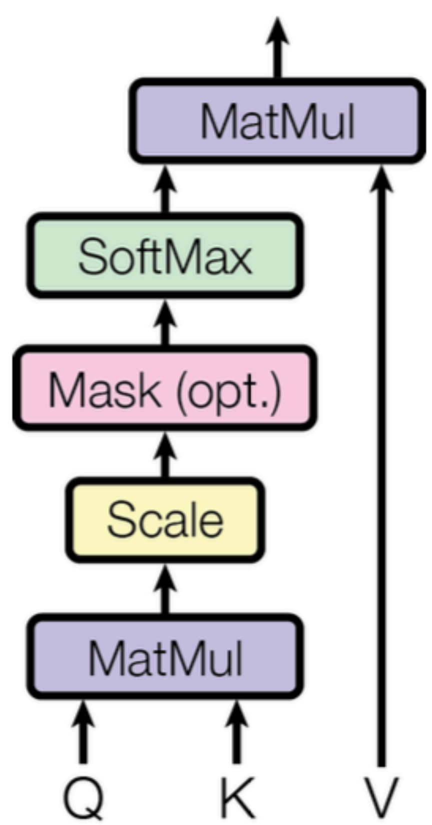
Encoder

- N = 6
- All layer output size 512
- Embedding
- Positional Encoding
- Multi-head Attention
- Residual Connection
- Position wise feed forward

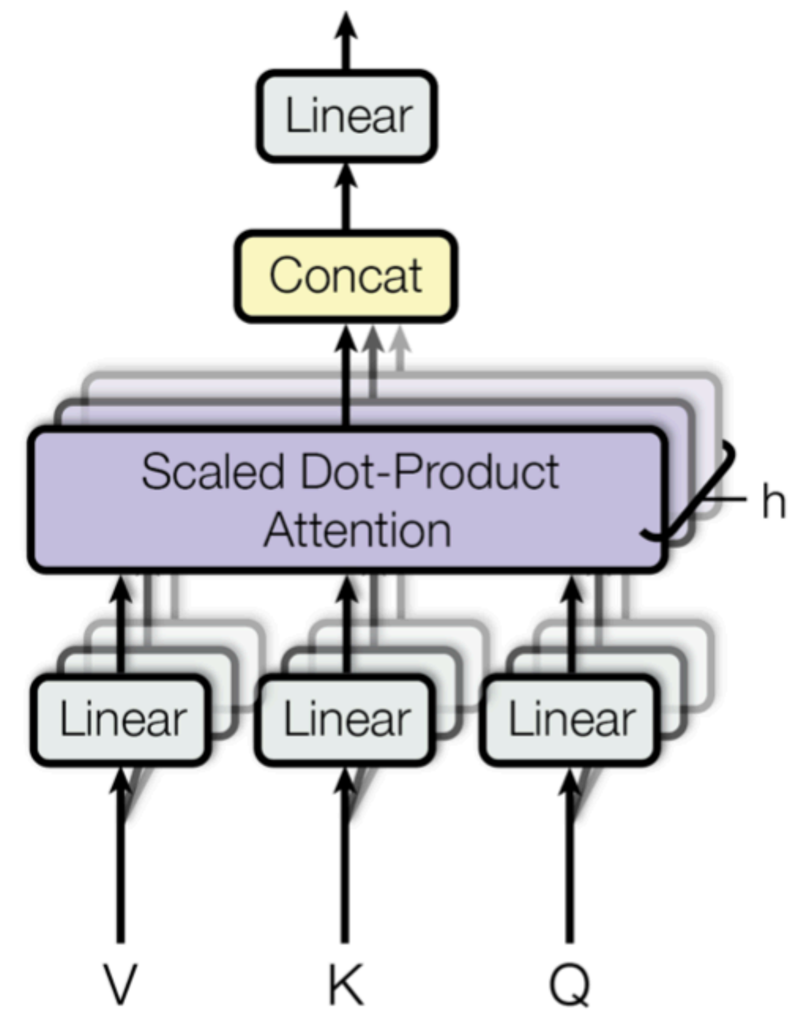


Scaled Dot Product and Multi-Head Attention

Scaled Dot-Product Attention



Multi-Head Attention

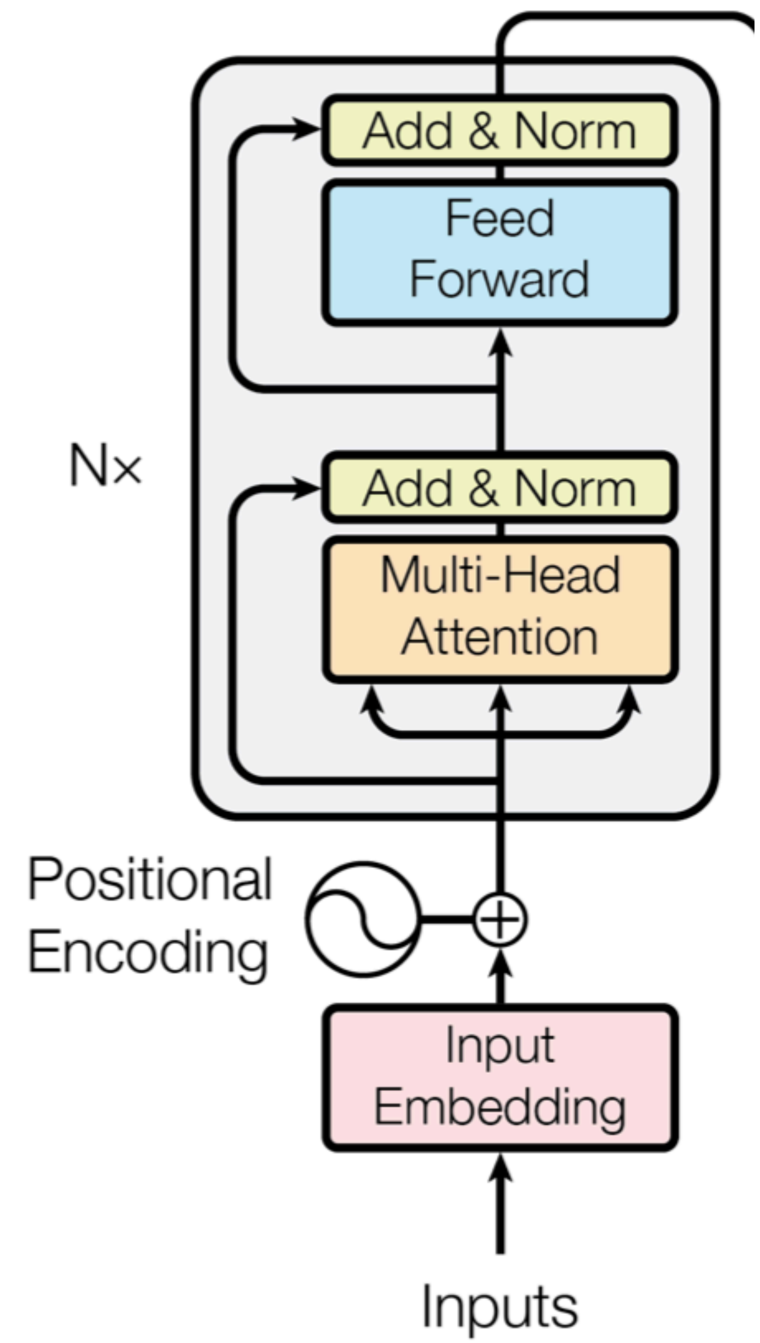


$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

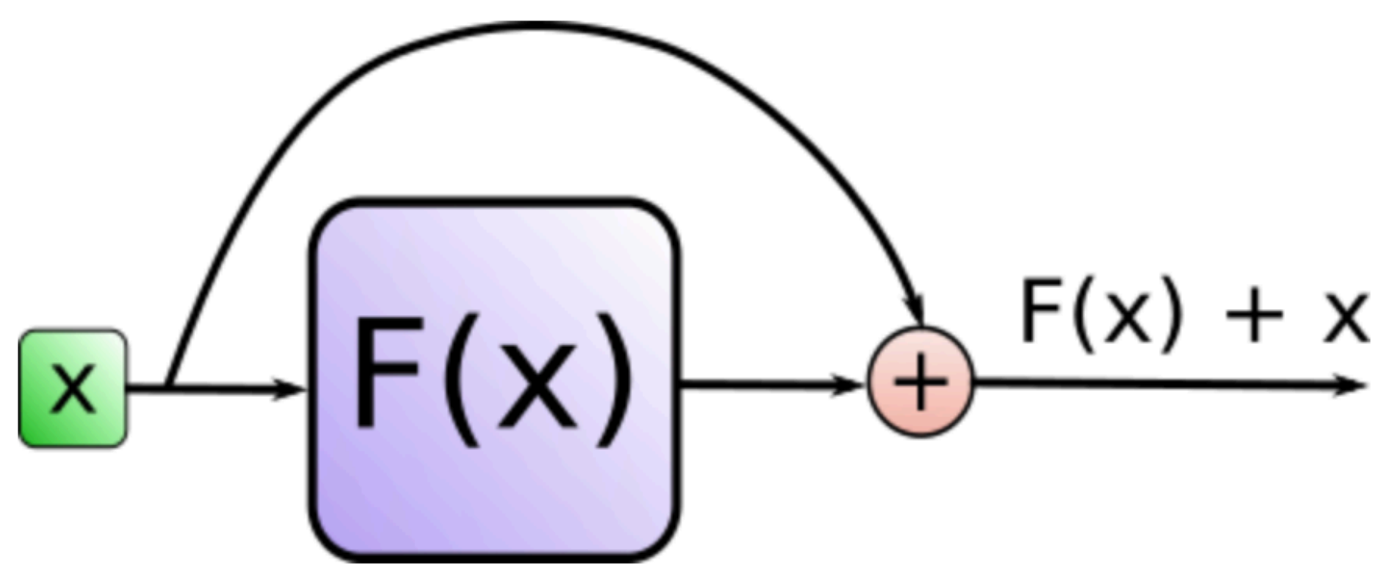
Encoder

- N = 6
- All layer output size 512
- Embedding
- Positional Encoding
- Multi-head Attention
- Residual Connection
- Position wise feed forward



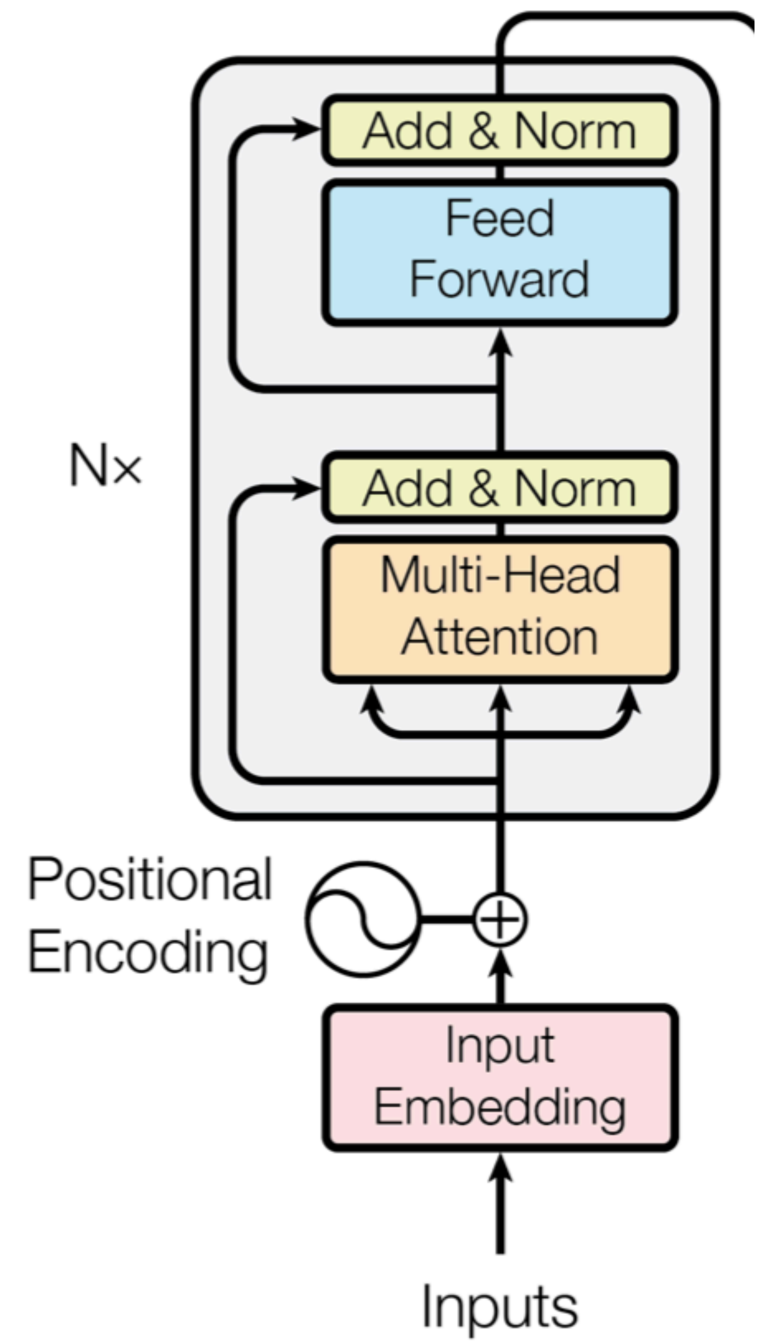
Residual Connection

- LayerNorm(x + Sublayer(x))



Encoder

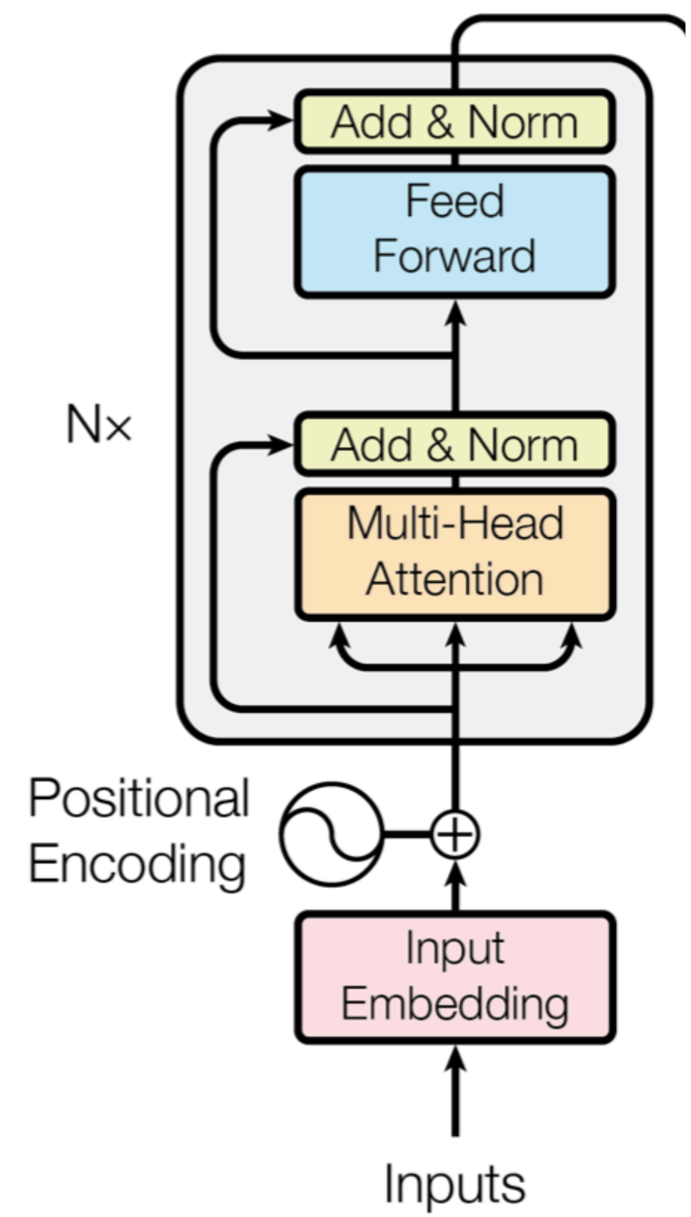
- N = 6
- All layer output size 512
- Embedding
- Positional Encoding
- Multi-head Attention
- Residual Connection
- Position wise feed forward



Position Wise Feed Forward

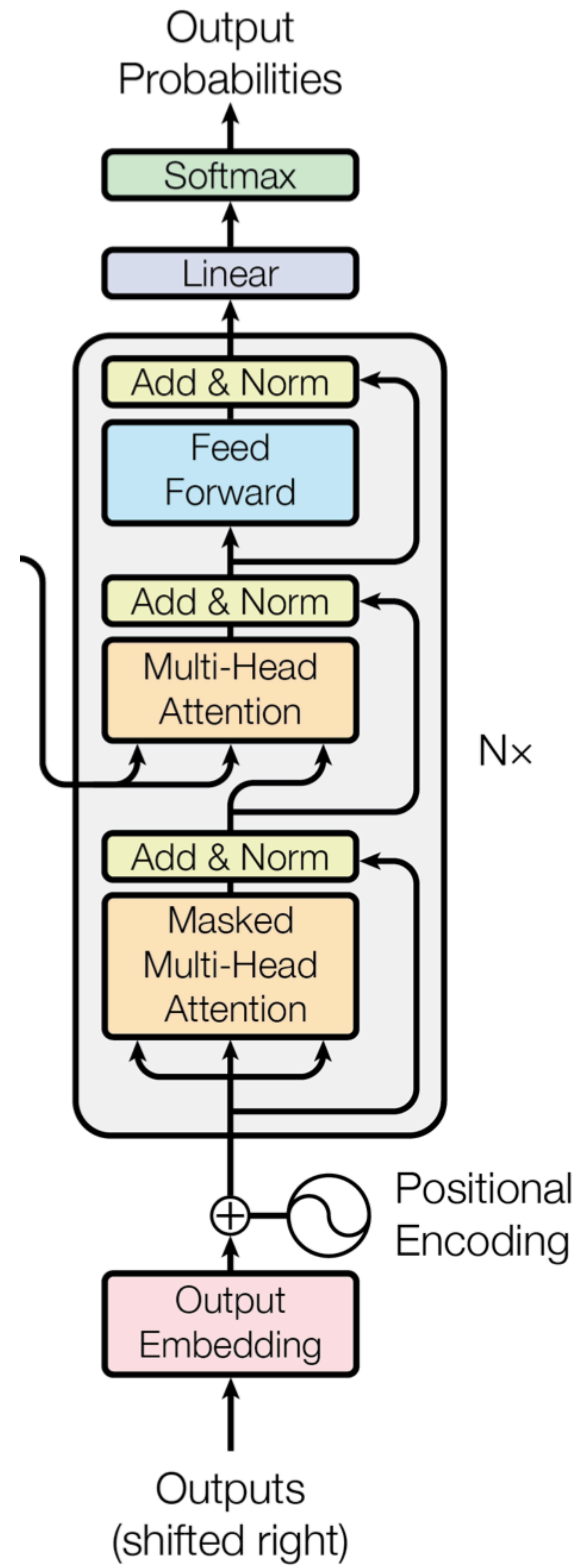
- two linear transformation with a ReLU activation in between

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$



Decoder

- N = 6
- All layer output size 512
- Embedding
- Positional Encoding
- Residual Connection: LayerNorm(x + Sublayer(x))
- Multi-head Attention
- Position wise feed forward
- softmax: $\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$



Q, V, K

- Queries (Q) come from previous decoder layer, and the memory keys (K) and values (V) come from the output of the encoder

- all three come from previous layer (Hidden State)

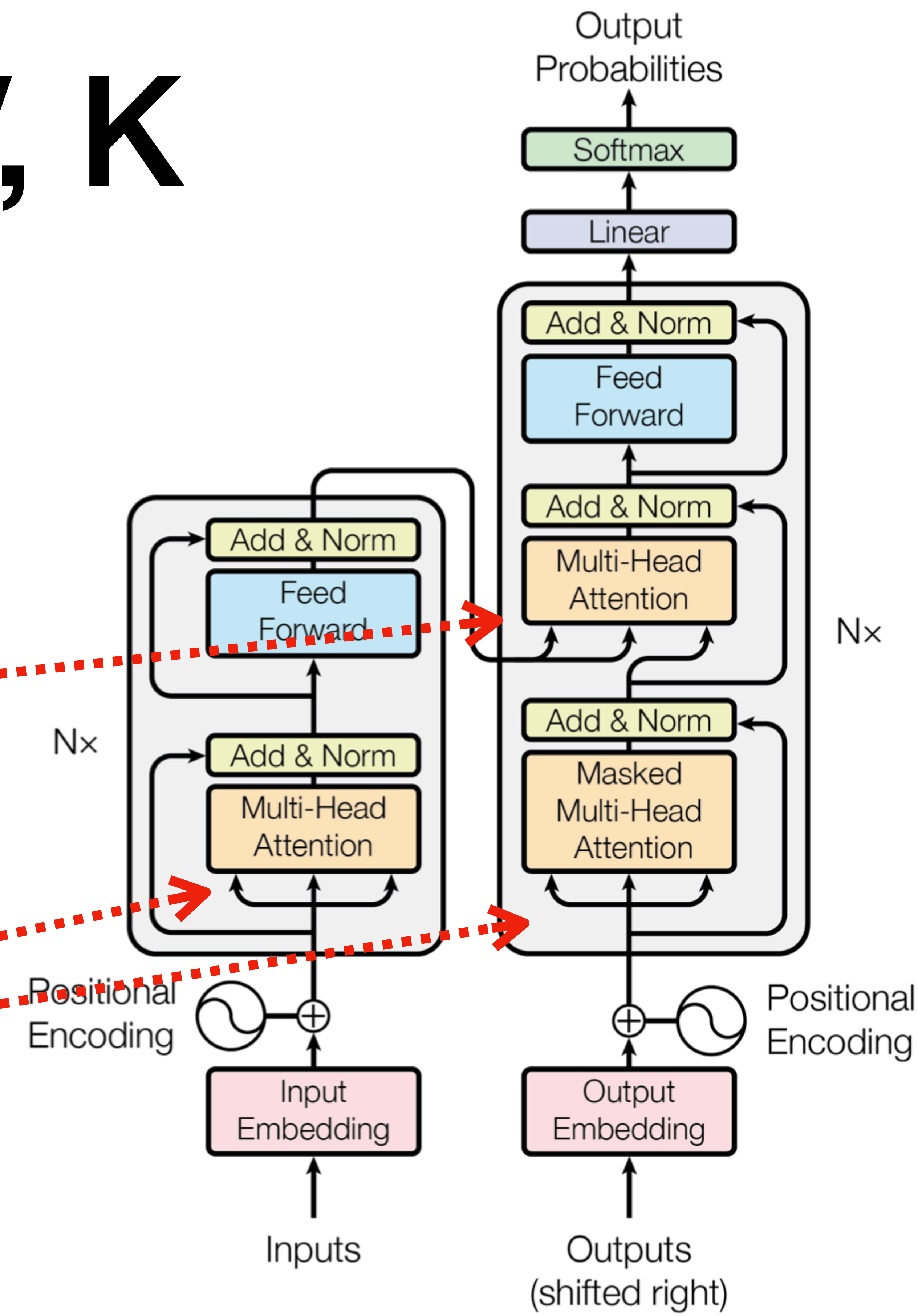


Figure 1: The Transformer - model architecture.

Training

- Data sets:
  - WMT 2014 English-German:
    - 4.5 million sentences pairs with 37K tokens.
  - WMT 2014 English-French:
    - 36M sentences, 32K tokens.
- Hardware:
  - 8 Nvidia P100 GPus (Base model 12 hours, big model 3.5 days)

Results

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.0**	$2.3 \cdot 10^{19}$	

More Results

	N	d_{model}	d_{ff}	h	d_k	d_v	P_{drop}	ϵ_{ls}	train steps	PPL (dev)	BLEU (dev)	params $\times 10^6$
base	6	512	2048	8	64	64	0.1	0.1	100K	4.92	25.8	65
(A)				1	512	512				5.29	24.9	
				4	128	128				5.00	25.5	
				16	32	32				4.91	25.8	
				32	16	16				5.01	25.4	
(B)					16					5.16	25.1	58
					32					5.01	25.4	60
(C)	2									6.11	23.7	36
	4									5.19	25.3	50
	8									4.88	25.5	80
		256			32	32				5.75	24.5	28
		1024			128	128				4.66	26.0	168
			1024							5.12	25.4	53
			4096							4.75	26.2	90
(D)							0.0			5.77	24.6	
							0.2			4.95	25.5	
								0.0		4.67	25.3	
								0.2		5.47	25.7	
(E)	positional embedding instead of sinusoids									4.92	25.7	
big	6	1024	4096	16			0.3		300K	**4.33**	**26.4**	213

Summary

- Introduces a new model, named **Transformer**
- In particular, introduces the concept of **multi-head attention mechanism**.
- It follows a **classical encoder** + **decoder structure**.
- It is an **autoregressive model**
- Achieves new state-of-the-art results in NMT