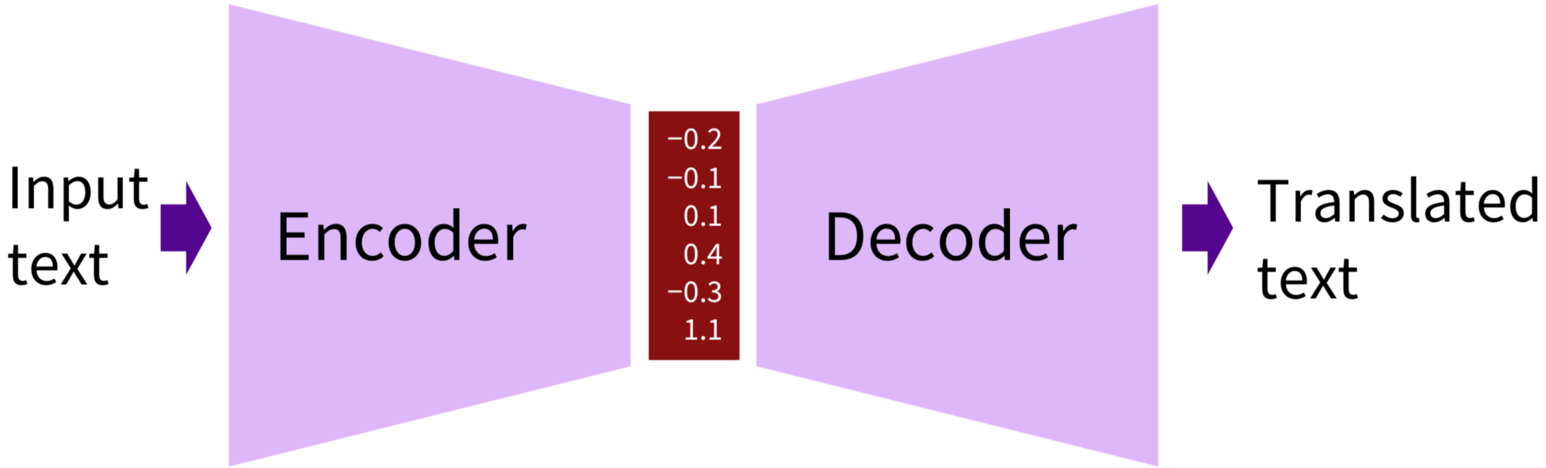
Attention Is All You Need ?

speaker： 徐菁
18/9/14

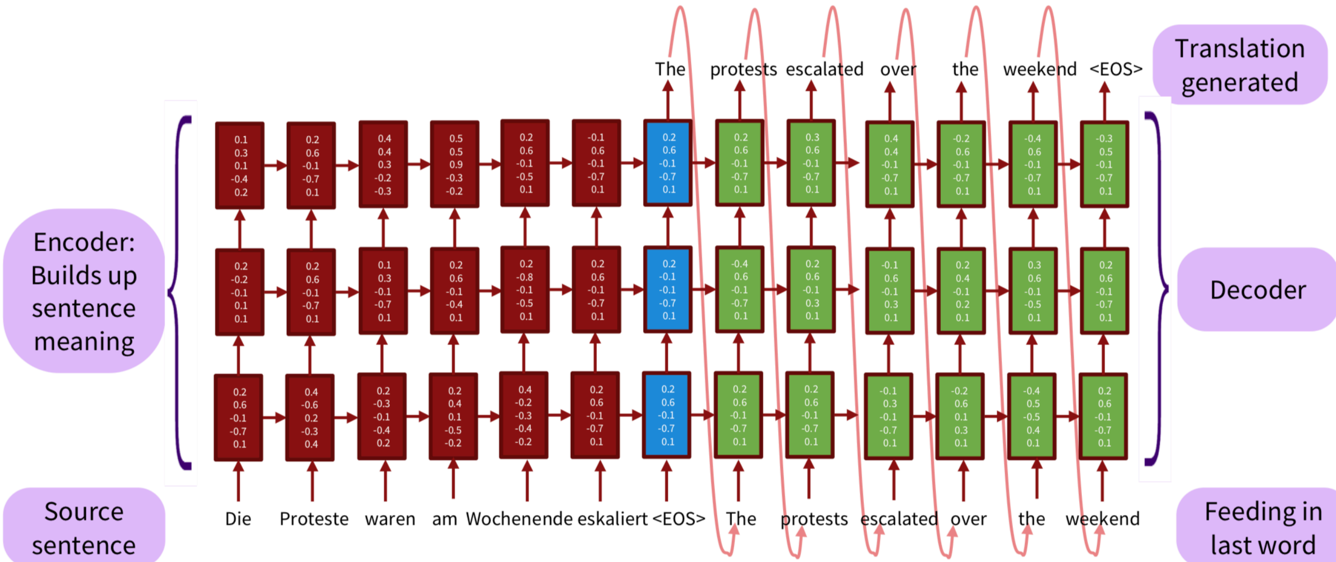
BACKGROUND

Neural Machine Translation Encoder-Decoder



BACKGROUND

RNNs Encoder-Decoder

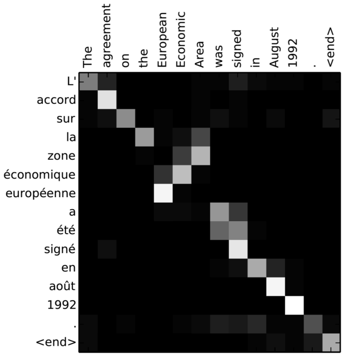


BACKGROUND

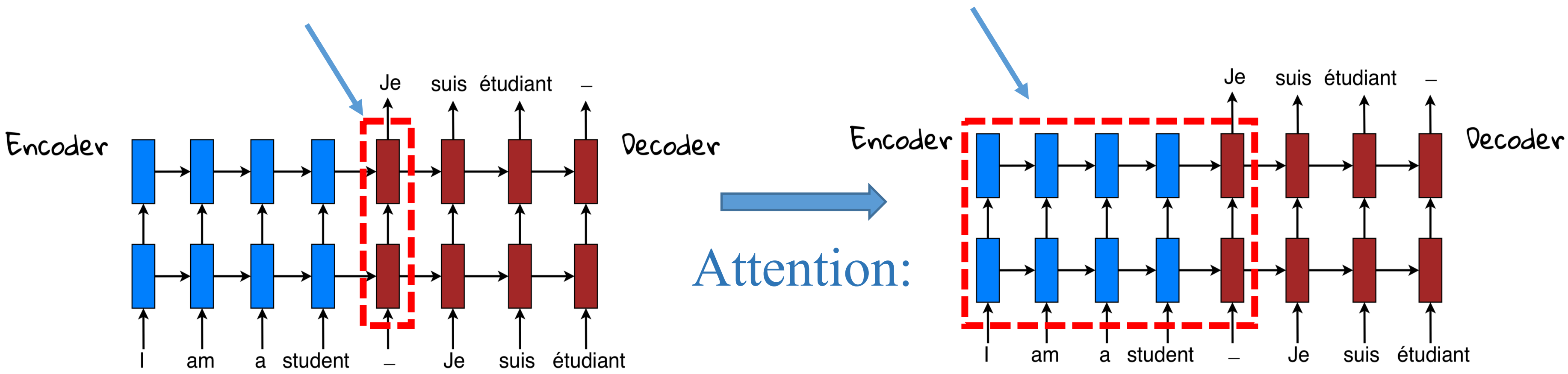
Neural Machine Translation Encoder-Decoder

Disadvantages:

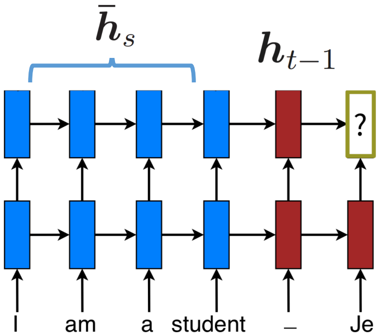
- Bottleneck: Source sentences compressed into a fixed-length vector.
- Long Sentences: Rapid decline in performance.
- Alignment: between source and translated sentences.



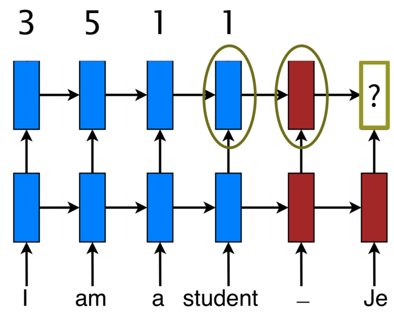
ATTENTION + RNNs:



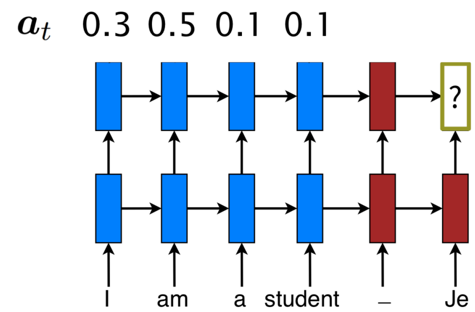
ATTENTION + RNNs:



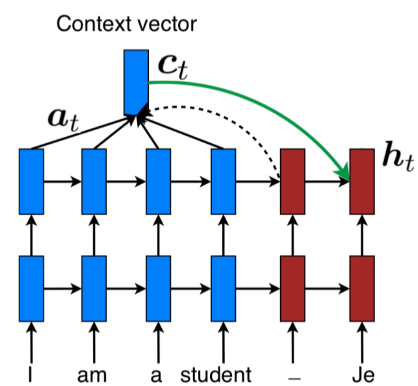
$\mathrm{score}(\boldsymbol{h}_{t-1}, \bar{\boldsymbol{h}}_s)$



$\boldsymbol{a}_t(s) = \frac{e^{\mathrm{score}(s)}}{\sum_{s'} e^{\mathrm{score}(s')}}$



$\boldsymbol{c}_t = \sum_s \boldsymbol{a}_t(s)\bar{\boldsymbol{h}}_s$



$$\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s \\ \boldsymbol{h}_t^\top \boldsymbol{W}_{\boldsymbol{a}} \bar{\boldsymbol{h}}_s \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W}_{\boldsymbol{a}}[\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s]\right) \end{cases}$$

Dzmitry Bahdanau, KyungHuyn Cho, and Yoshua Bengio. ***Neural Machine Translation by Jointly Learning to Translate and Align****. ICLR'15.*

ATTENTION + RNNs:

Performance(Sent Lengths):

- Vanilla RNN : 7-10 Vanilla LSTM: 30 ~ Attention+LSTM: 70 ~

Attention is great !

- Improves NMT performance
- Helps with vanishing gradient problem
- Solves the bottleneck problem
- Gets alignment for free

Attention is all you need

SMILE LAB
Statistical Machine Intelligence & Learning

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

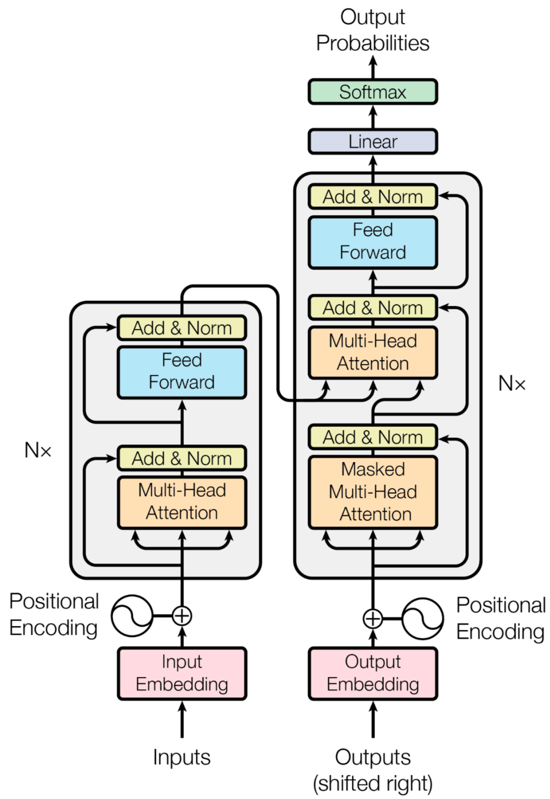
Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Attention is all you need

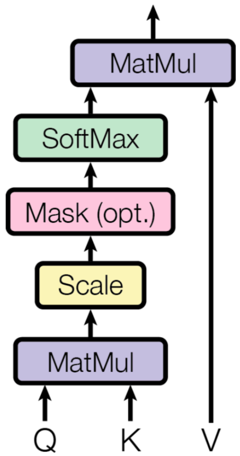
Transformer



Attention is all you need

Multi-Head Attention

Scaled Dot-Product Attention



Query Key - Values

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

$[|Q| \times d_k] \times [d_k \times |K|] \times [|K| \times d_v]$

softmax row-wise

$= [|Q| \times d_v]$

Q K^T V

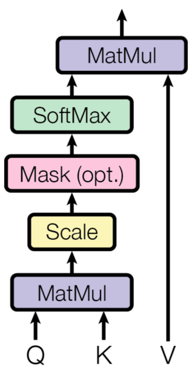
Output is computed as a weighted sum of the Values

Weight is computed by a compatibility function with Query and corresponding Key

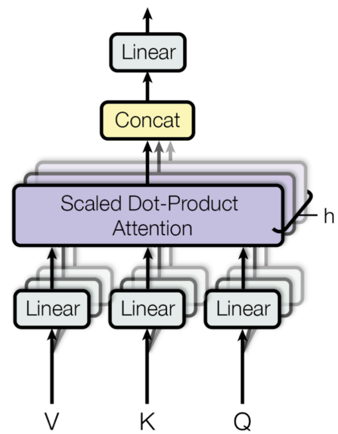
Attention is all you need

Multi-Head Attention

Scaled Dot-Product Attention



Multi-Head Attention



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

$$W_i^Q \in \mathbb{R}^{d_k \times \frac{d_k}{h}}$$

$$W_i^K \in \mathbb{R}^{d_k \times \frac{d_k}{h}}$$

$$W_i^V \in \mathbb{R}^{d_v \times \frac{d_v}{h}}$$

$$head_i \in \mathbb{R}^{|Q| \times \frac{d_v}{h}}$$

$[|Q| \times d_k] \times [d_k \times |K|] \times [|K| \times d_v]$

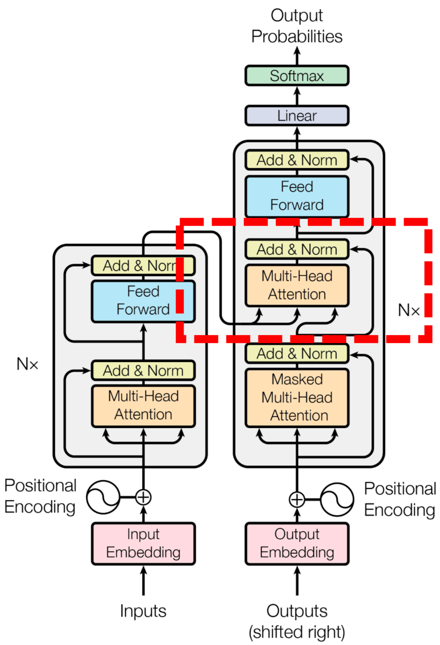
softmax
row-wise

$= [|Q| \times d_v]$

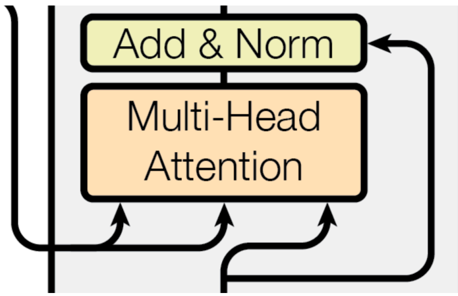
Jointly attend to information from different representation subspaces at different positions

Attention is all you need

Multi-Head Attention

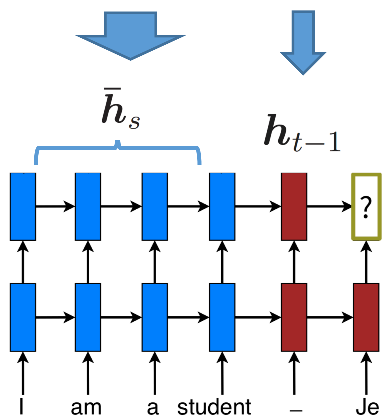


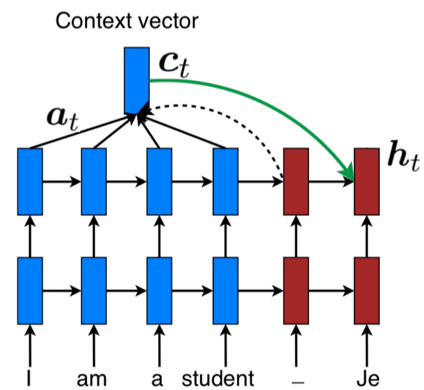
Encoder



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

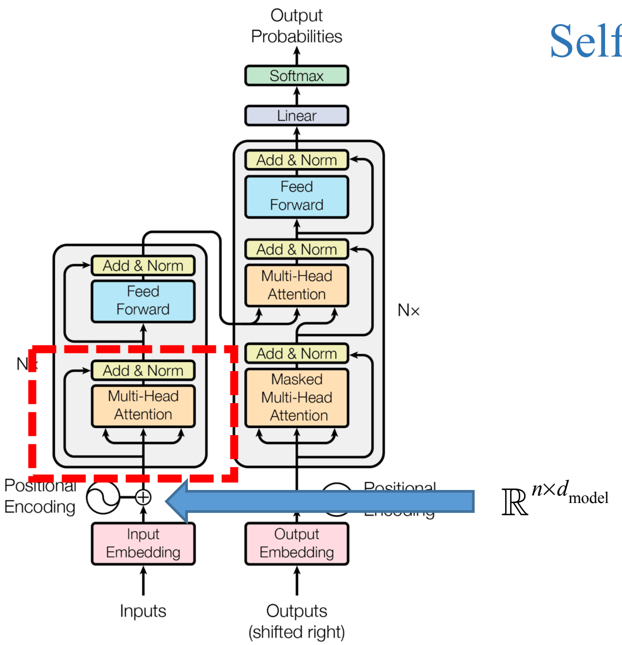
Values - Key Query





Attention is all you need

Multi-Head Attention



Self-Attention:

$$Attention(X, X, X)$$

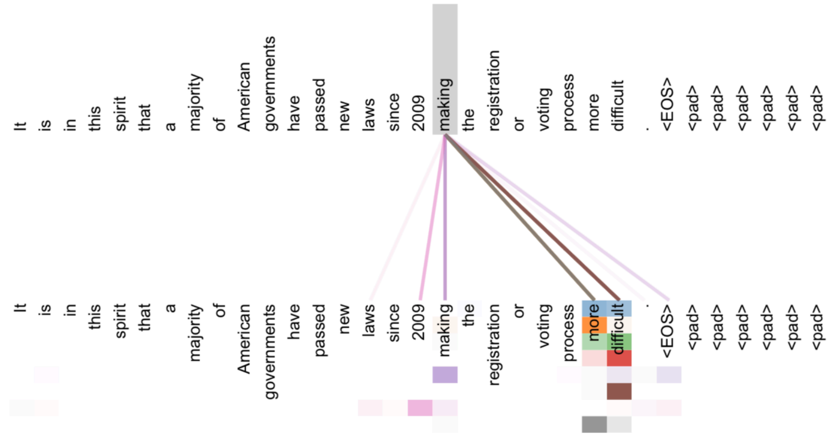
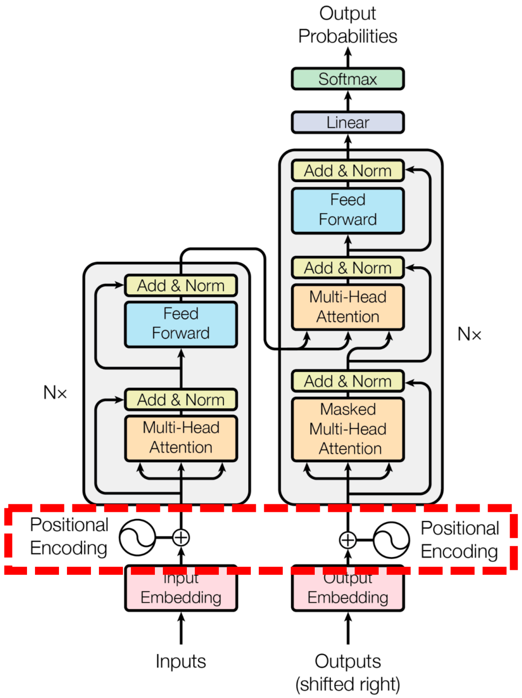


Figure 3: An example of the attention mechanism following long-distance dependencies in the encoder self-attention in layer 5 of 6. Many of the attention heads attend to a distant dependency of the verb 'making', completing the phrase 'making...more difficult'. Attentions here shown only for the word 'making'. Different colors represent different heads. Best viewed in color.

Attention Is All You Need

Positional Encoding



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

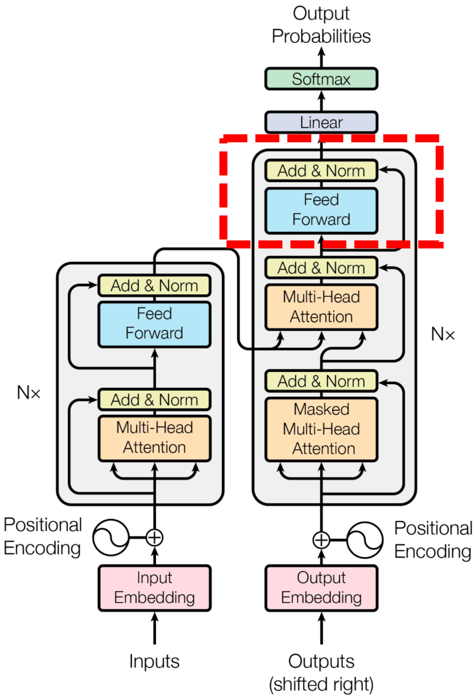
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

Make use of the order of the sequence

Inject some information about the relative or absolute position

Attention Is All You Need

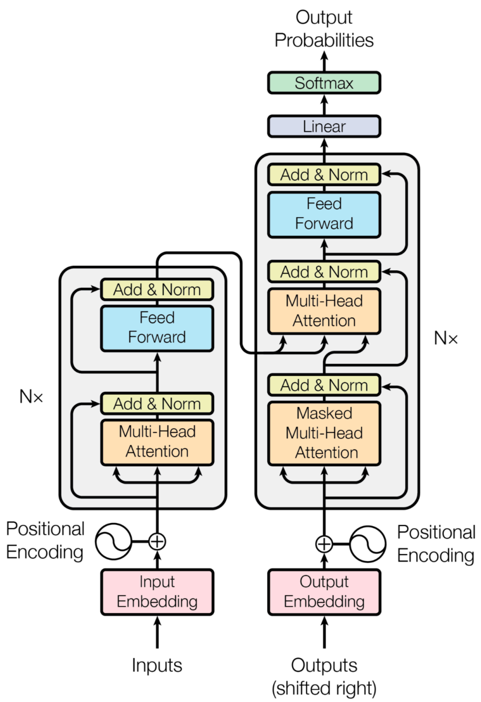
Feed-Forward :



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

Attention Is All You Need

Model Variations:



	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

Attention Is All You Need

Result:

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	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**	**41.8**	$2.3 \cdot 10^{19}$	

Attention is all we need ?

- Attention has enough ability to capture enough information.
- Multi-head attention mechanism
- Self-Attention mechanism

Combine Attention with other architectures.

Thank you