Two Novel Sequence-to-Sequence Architectures

CS6604 Paper Presentation

Yufeng Ma

Department of Computer Science
Virginia Tech

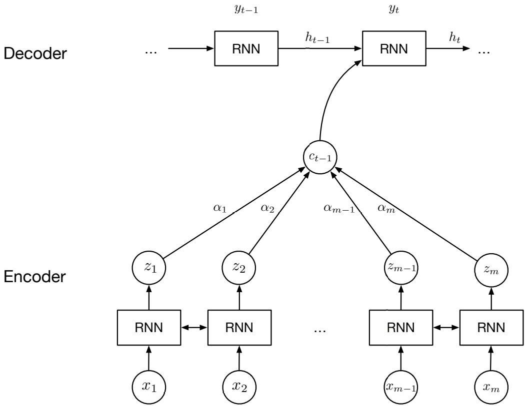
February 26, 2018

Overview

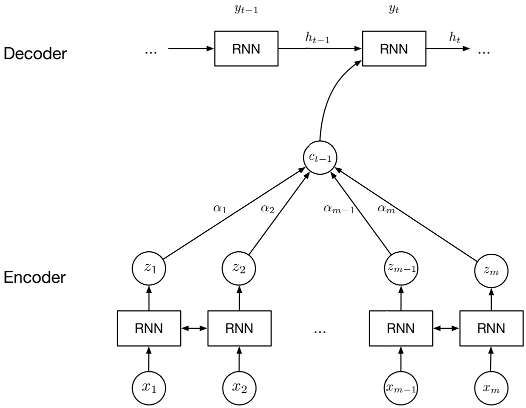
1. Sequence-to-Sequence Background
   - Sequence-to-Sequence Architecture
   - Limitations and Problems
2. Convolutional Sequence to Sequence Learning - ConvS2S
   - Convolution Architecture
   - Experiments
   - Results
3. Attention Is All You Need - Transformer
   - Transformer Architecture
   - Training Setup
   - Results

Sequence-to-Sequence Architecture

Sequence-to-Sequence Architecture



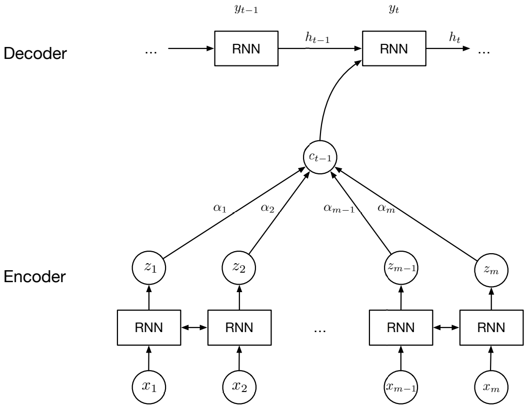
Sequence-to-Sequence Architecture



Notation

- Encoder input: $x = (x_1, x_2, \ldots, x_m)$;
- Encoder hidden: $z = (z_1, z_2, \ldots, z_m)$;
- Decoder output: $y = (y_1, y_2, \ldots, y_n)$;
- Decoder hidden: $h = (h_1, h_2, \ldots, h_n)$;

Sequence-to-Sequence Architecture



Notation

- Encoder input: $x = (x_1, x_2, \ldots, x_m)$;
- Encoder hidden: $z = (z_1, z_2, \ldots, z_m)$;
- Decoder output: $y = (y_1, y_2, \ldots, y_n)$;
- Decoder hidden: $h = (h_1, h_2, \ldots, h_n)$;

Attention

$$e_{ij} = a(h_i, z_j)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{m} \exp(e_{ik})}$$

$$c_i = \sum_{k=1}^{m} \alpha_{ij} z_j$$

Limitations and Problems

Limitations and Problems

Issues with RNNs

- Hard to parallelize efficiently;
- Later inputs are back-propagated less;
- Lengthy path length of long-term dependencies;
- Transmitting local and global information through one bottleneck;

Limitations and Problems

Issues with RNNs

- Hard to parallelize efficiently;
- Later inputs are back-propagated less;
- Lengthy path length of long-term dependencies;
- Transmitting local and global information through one bottleneck;

Paper Emphasis

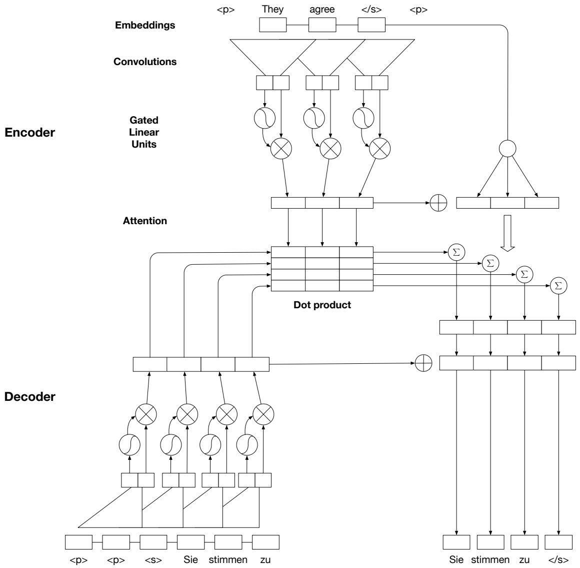
- ConvS2S:
    - Parallelization to some extent;
    - Still limited by convolution size;
- Transformer:
    - Complete parallelization;
    - Constant dependency path length;
    - Multiple attention interaction;

Overview

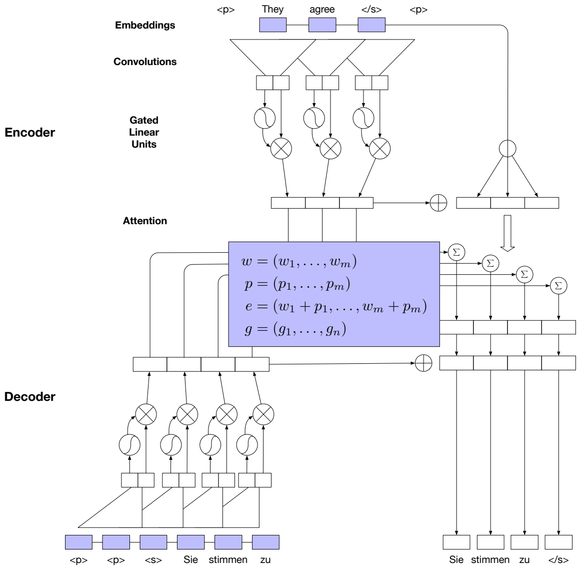
1. Sequence-to-Sequence Background
   - Sequence-to-Sequence Architecture
   - Limitations and Problems
2. Convolutional Sequence to Sequence Learning - ConvS2S
   - Convolution Architecture
   - Experiments
   - Results
3. Attention Is All You Need - Transformer
   - Transformer Architecture
   - Training Setup
   - Results

ConvS2S Architecture

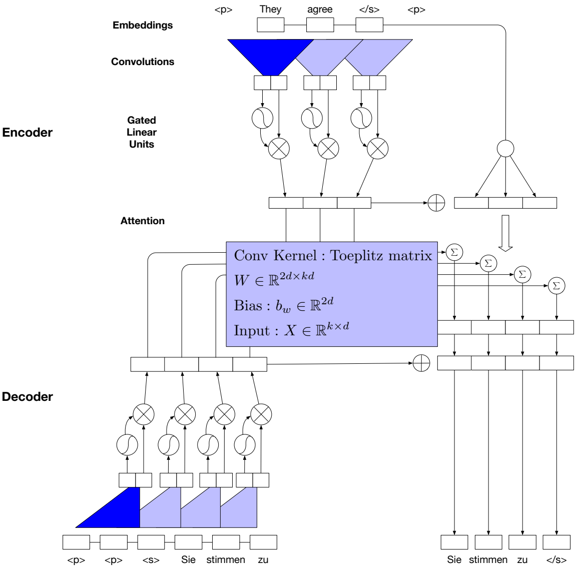
ConvS2S Architecture



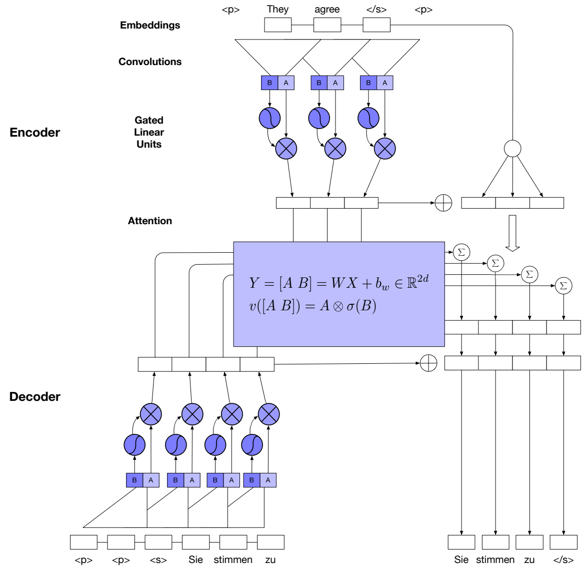
Position Embeddings



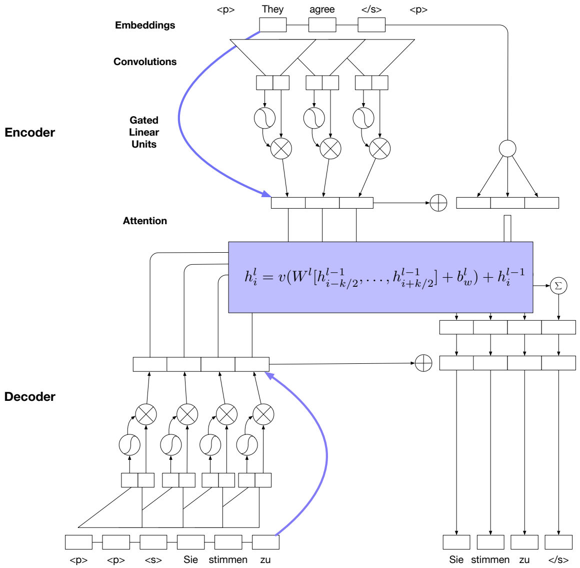
Convolution Block



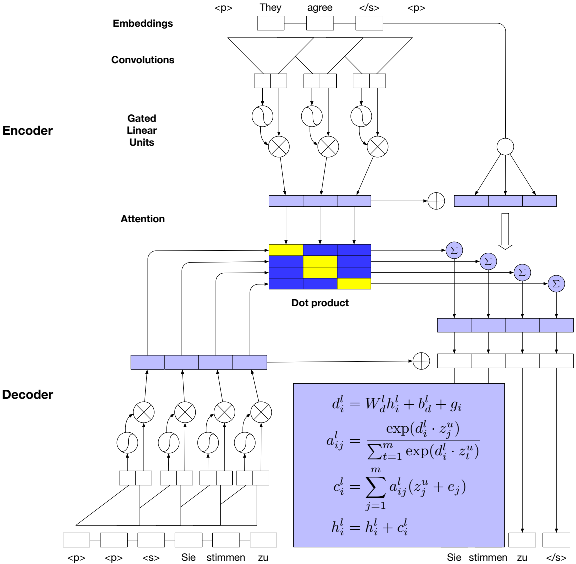
Gated Linear Units



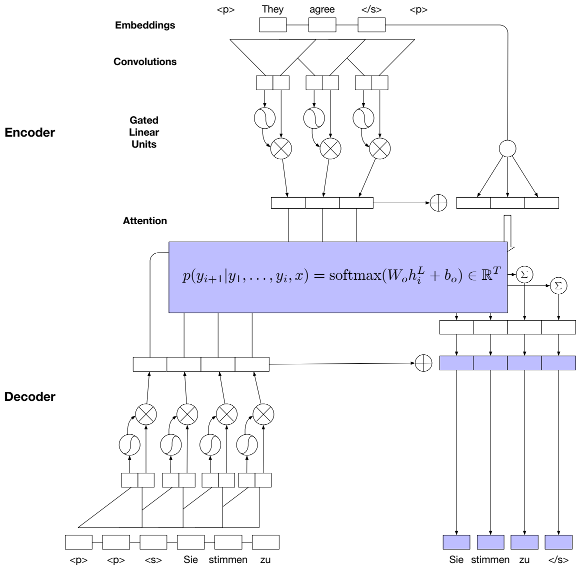
Residual Connections



Multi-step Attention



Word Prediction



Tricks to Stabilize Learning

Tricks to Stabilize Learning

Normalization Strategy

- Residual block & attention output: multiplied by $\sqrt{0.5}$;
- Conditional input c_i^l: scaled by $m\sqrt{m}$;
- Encoder gradients: scaled by the $\#$ of attention mechanisms;

Tricks to Stabilize Learning

Normalization Strategy

- Residual block & attention output: multiplied by $\sqrt{0.5}$;
- Conditional input c_i^l: scaled by $m\sqrt{m}$;
- Encoder gradients: scaled by the $\#$ of attention mechanisms;

Initialization

- Word embedding: $\mathcal{N}(0, 1)$;
- Layers not directly into GLU: $\mathcal{N}(0, \sqrt{1/n_l})$;
- Layers followed by GLU: $\mathcal{N}(0, \sqrt{4/n_l})$;
- Dropout: $\mathcal{N}(0, \sqrt{4p/n_l})$ for ones into GLU, $\mathcal{N}(0, \sqrt{p/n_l})$ otherwise;

Experimental Setup

Experimental Setup

NLP tasks

- WMT'16 English-Romanian;
- WMT'14 English-German;
- WMT'14 English-French;
- Abstractive Summarization: Gigaword, DUC-2004;

Experimental Setup

NLP tasks

- WMT'16 English-Romanian;
- WMT'14 English-German;
- WMT'14 English-French;
- Abstractive Summarization: Gigaword, DUC-2004;

Evaluation

- Average of 3 runs from different random seeds;
- BLEU for translation;
- ROUGE for summarization;

Machine Translation Improvement

Machine Translation Improvement

WMT'16 English-Romanian	BLEU
Sennrich et al. (2016b) GRU (BPE 90K)	28.1
ConvS2S (Word 80K)	29.45
ConvS2S (BPE 40K)	30.02

WMT'14 English-German	BLEU
Luong et al. (2015) LSTM (Word 50K)	20.9
Kalchbrenner et al. (2016) ByteNet (Char)	23.75
Wu et al. (2016) GNMT (Word 80K)	23.12
Wu et al. (2016) GNMT (Word pieces)	24.61
ConvS2S (BPE 40K)	25.16

WMT'14 English-French	BLEU
Wu et al. (2016) GNMT (Word 80K)	37.90
Wu et al. (2016) GNMT (Word pieces)	38.95
Wu et al. (2016) GNMT (Word pieces) + RL	39.92
ConvS2S (BPE 40K)	40.51

Table 1. Accuracy on WMT tasks comapred to previous work. ConvS2S and GNMT results are averaged over several runs.

Generation Speed

Generation Speed

	BLEU	Time (s)
GNMT GPU (K80)	31.20	3,028
GNMT CPU 88 cores	31.20	1,322
GNMT TPU	31.21	384
ConvS2S GPU (K40) $b=1$	33.45	327
ConvS2S GPU (M40) $b=1$	33.45	221
ConvS2S GPU (GTX-1080ti) $b=1$	33.45	142
ConvS2S CPU 48 cores $b=1$	33.45	142
ConvS2S GPU (K40) $b=5$	34.10	587
ConvS2S CPU 48 cores $b=5$	34.10	482
ConvS2S GPU (M40) $b=5$	34.10	406
ConvS2S GPU (GTX-1080ti) $b=5$	34.10	256

Table 3. CPU and GPU generation speed in seconds on the development set of WMT'14 English-French. We show results for different beam sizes b. GNMT figures are taken from Wu et al. (2016). CPU speeds are not directly comparable because Wu et al. (2016) use a 88 core machine versus our 48 core setup.

Position Embeddings & Attention Layers

Position Embeddings & Attention Layers

	PPL	BLEU
ConvS2S	6.64	21.7
-source position	6.69	21.3
-target position	6.63	21.5
-source & target position	6.68	21.2

Table 4. Effect of removing position embeddings from our model in terms of validation perplexity (valid PPL) and BLEU.

Position Embeddings & Attention Layers

	PPL	BLEU
ConvS2S	6.64	21.7
-source position	6.69	21.3
-target position	6.63	21.5
-source & target position	6.68	21.2

Table 4. Effect of removing position embeddings from our model in terms of validation perplexity (valid PPL) and BLEU.

Attn Layers	PPL	BLEU
1,2,3,4,5	6.65	21.63
1,2,3,4	6.70	21.54
1,2,3	6.95	21.36
1,2	6.92	21.47
1,3,5	6.97	21.10
1	7.15	21.26
2	7.09	21.30
3	7.11	21.19
4	7.19	21.31
5	7.66	20.24

Table 5. Multi-step attention in all five decoder layers or fewer layers in terms of validation perplexity (PPL) and test BLEU.

Abstractive Summarization

Abstract Summarization

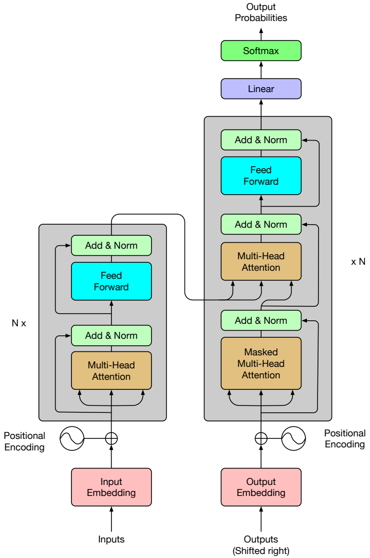
	DUC-2004			Gigaword		
	RG-1 (R)	**RG-2 (R)**	**RG-L (R)**	**RG-1 (F)**	**RG-2 (F)**	**RG-L (F)**
RNN MLE (Shen et al., 2016)	24.92	8.60	22.25	32.67	15.23	30.56
RNN MRT (Shen et al., 2016)	30.41	10.87	26.79	36.54	16.59	33.44
WFE (Suzuki & Nagata, 2017)	32.28	10.54	27.80	36.30	17.31	33.88
ConvS2S	30.44	10.84	26.90	35.88	17.48	33.29

Table 6. Accuracy on two summarization tasks in terms of Rouge-1 (RG-1), Rouge-2 (RG-2), and Rouge-L (RG-L).

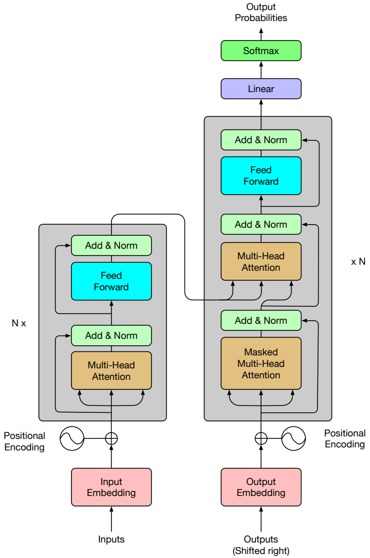
Overview

Transformer Architecture

Transformer Architecture



Transformer Architecture

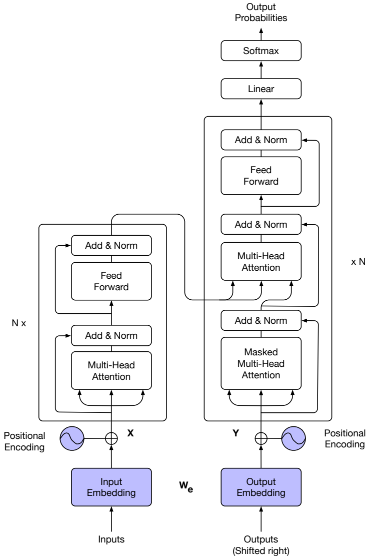


Key Components

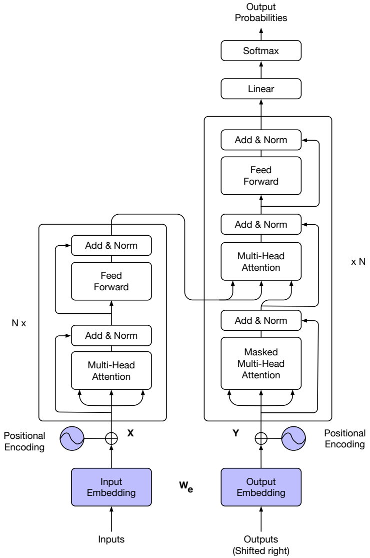
- Positional Encoding;
- Multi-Head Attention;
- Residual Connection;
- Weight Tying;

Positional Encoding

Positional Encoding



Positional Encoding

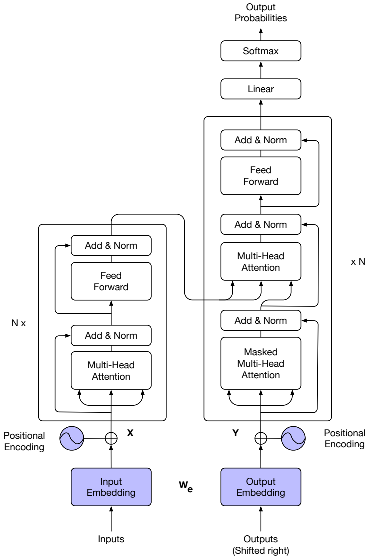


Word Embedding

$$w = (w_1, w_2, \ldots, w_n)$$

- $w_i \in \mathbb{R}^{d_{\text{model}}}$
- $W_e \in \mathbb{R}^{|V| \times d_{\text{model}}}$

Positional Encoding



Word Embedding

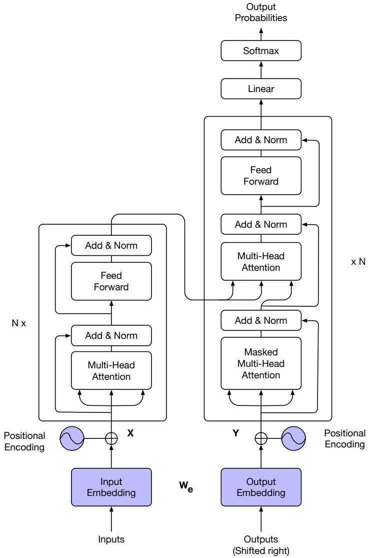
$$w = (w_1, w_2, \ldots, w_n)$$

- $w_i \in \mathbb{R}^{d_{\text{model}}}$
- $W_e \in \mathbb{R}^{|V| \times d_{\text{model}}}$

Positional Encoding

- pos,2i: $sin(\text{pos}/10000^{2i/d_{\text{model}}})$;
- pos,2i+1: $cos(\text{pos}/10000^{2i/d_{\text{model}}})$;
- relative pos: $sin(\text{pos}+k) = sin(\text{pos})cos(k) + cos(\text{pos})sin(k)$

Positional Encoding



Word Embedding

$$w = (w_1, w_2, \ldots, w_n)$$

- $w_i \in \mathbb{R}^{d_{\text{model}}}$
- $W_e \in \mathbb{R}^{|V| \times d_{\text{model}}}$

Positional Encoding

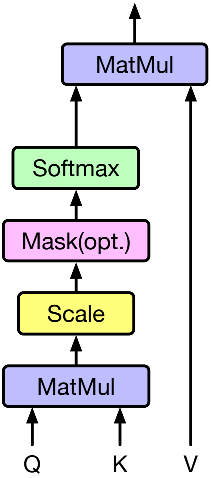
- pos,2i: $sin(\text{pos}/10000^{2i/d_{\text{model}}})$;
- pos,2i+1: $cos(\text{pos}/10000^{2i/d_{\text{model}}})$;
- relative pos: $sin(\text{pos}+k) = sin(\text{pos})cos(k)+cos(\text{pos})sin(k)$

Input to NN

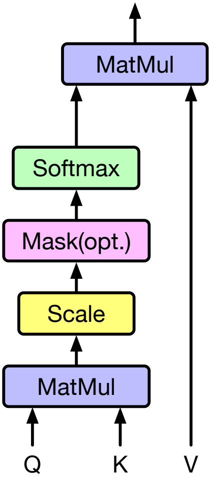
$$X = w + \text{PE} \in \mathbb{R}^{n \times d_{\text{model}}}$$

Scaled Dot-Product Attention

Scaled Dot-Product Attention



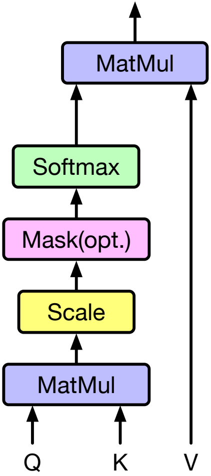
Scaled Dot-Product Attention



Notation

- A set of queries: $Q \in \mathbb{R}^{q \times d_k}$;
- Key-value pairs: $K \in \mathbb{R}^{n \times d_k}, V \in \mathbb{R}^{n \times d_v}$;

Scaled Dot-Product Attention



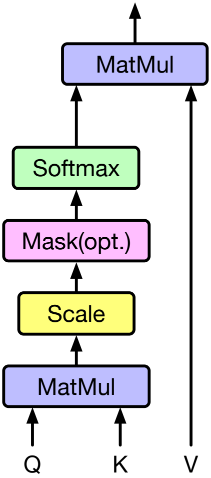
Notation

- A set of queries: $Q \in \mathbb{R}^{q \times d_k}$;
- Key-value pairs: $K \in \mathbb{R}^{n \times d_k}, V \in \mathbb{R}^{n \times d_v}$;

Attention

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \in \mathbb{R}^{q \times d_v}$$

Scaled Dot-Product Attention



Notation

- A set of queries: $Q \in \mathbb{R}^{q \times d_k}$;
- Key-value pairs: $K \in \mathbb{R}^{n \times d_k}, V \in \mathbb{R}^{n \times d_v}$;

Attention

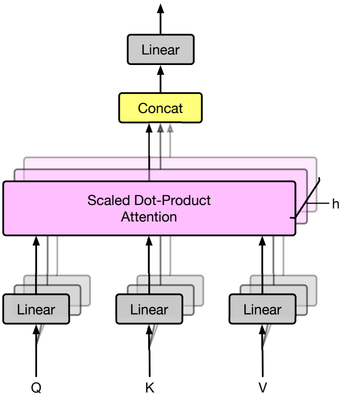
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^\top}{\sqrt{d_k}})V \in \mathbb{R}^{q \times d_v}$$

Masking for Decoder

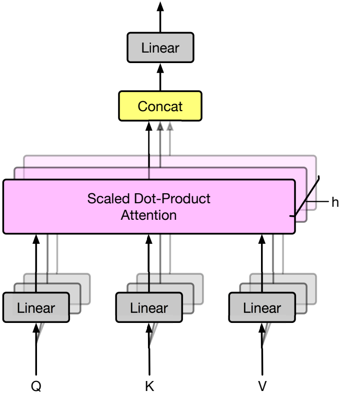
- Illegal connections

Multi-Head Attention

Multi-Head Attention



Multi-Head Attention

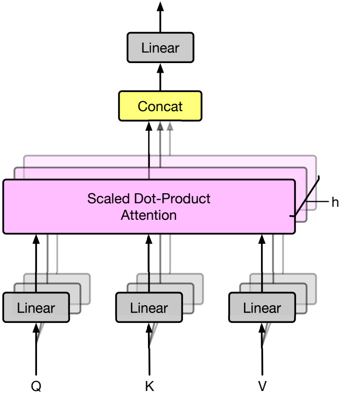


Multi-Head Attention

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_\mathrm{h})W^O$$

$$\text{where } \mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Multi-Head Attention



Multi-Head Attention

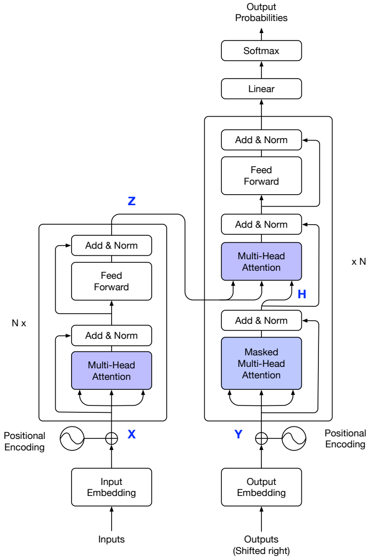
$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_\mathrm{h})W^O$$

$$\text{where } \mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

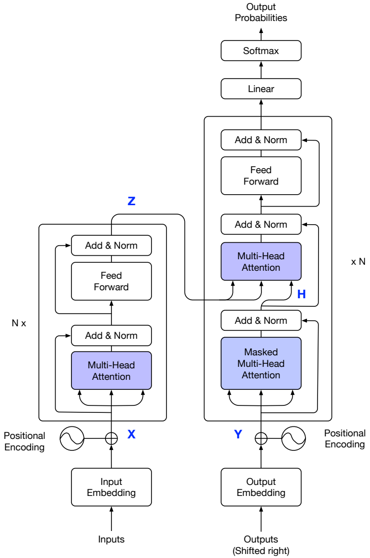
- $W_i^Q, W_i^K \in \mathbb{R}^{d_\mathrm{model} \times d_k}$;
- $W_i^V \in \mathbb{R}^{d_\mathrm{model} \times d_v}$;
- $W^O \in \mathbb{R}^{hd_v \times d_\mathrm{model}}$;

Transformer Architecture

Transformer Architecture



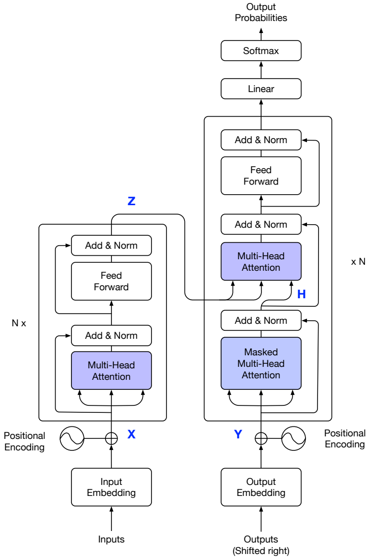
Transformer Architecture



Encoder Self-Attention

$$\mathrm{MultiHead}(X, X, X)$$

Transformer Architecture



Encoder Self-Attention

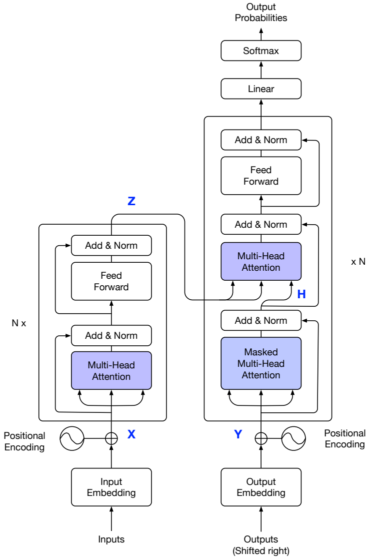
$$\mathrm{MultiHead}(X, X, X)$$

Decoder Masked Attention

$$\mathrm{MultiHead}(Y, Y, Y)$$

- Illegal connect: $-\infty$ in softmax

Transformer Architecture



Encoder Self-Attention

$$\mathrm{MultiHead}(X, X, X)$$

Decoder Masked Attention

$$\mathrm{MultiHead}(Y, Y, Y)$$

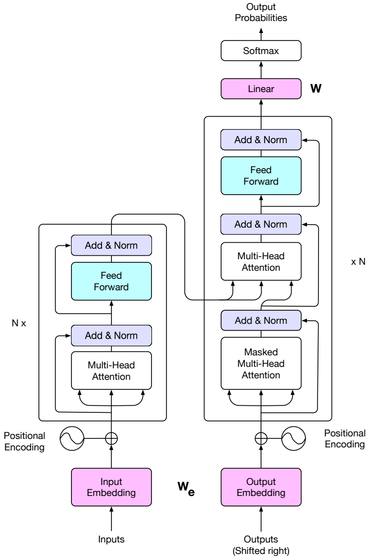
- Illegal connect: $-\infty$ in softmax

Encoder-Decoder Attention

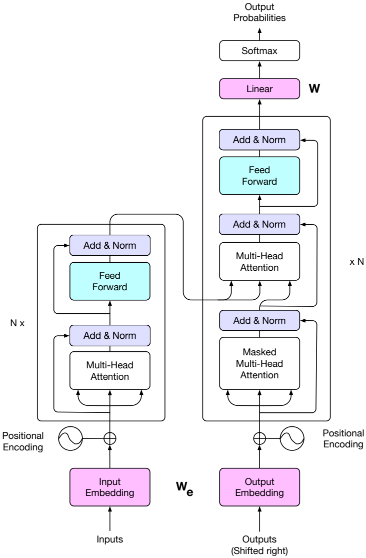
$$\mathrm{MultiHead}(H, Z, Z)$$

Transformer Architecture

Transformer Architecture



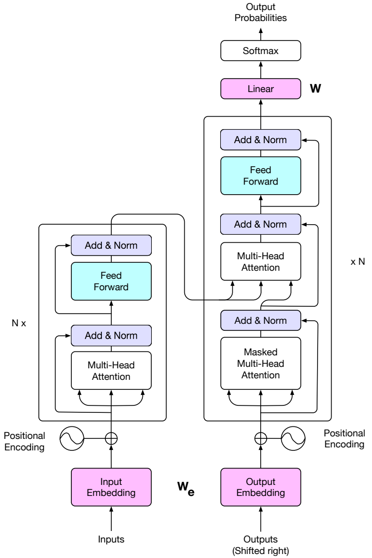
Transformer Architecture



Residual+LayerNorm

$$\mathrm{LayerNorm}(x + \mathrm{Sublayer}(x))$$

Transformer Architecture



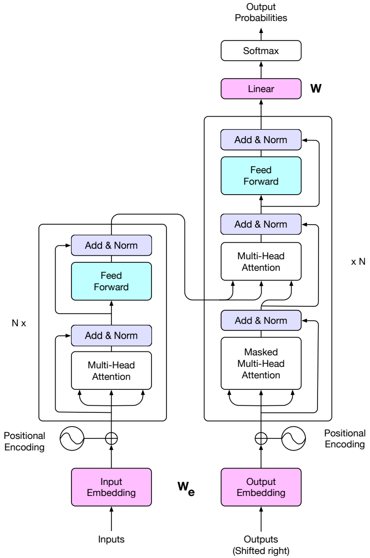
Residual+LayerNorm

$$\mathrm{LayerNorm}(x + \mathrm{Sublayer}(x))$$

Feed-Forward Network

$$\max(0, xW_1 + b_1)W_2 + b_2$$

Transformer Architecture



Residual+LayerNorm

$$\mathrm{LayerNorm}(x + \mathrm{Sublayer}(x))$$

Feed-Forward Network

$$\max(0, xW_1 + b_1)W_2 + b_2$$

Weight Tying

$$W = W_e^\top \in \mathbb{R}^{d_{\text{model}} \times |V|}$$

Why Self-Attention

Why Self-Attention

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted 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)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Why Self-Attention

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted 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)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

- Lower complexity per layer;
- Constant sequential operations;
- Constant path length of long-term dependencies;

Training Setup

Training Setup

Training Data

- WMT 2014 English-German;
- WMT 2014 English-French;

Training Setup

Training Data

- WMT 2014 English-German;
- WMT 2014 English-French;

Optimizer

- Adam: $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-9}$;
- $lrate = d_{\text{model}}^{-0.5} \cdot \min(\text{step\_num}^{-0.5}, \text{step\_num} \cdot \text{warmup\_steps}^{-1.5})$;
- $\text{warmup\_steps} = 400$;

Training Setup

Training Data

- WMT 2014 English-German;
- WMT 2014 English-French;

Optimizer

- Adam: $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-9}$;
- $lrate = d_{\text{model}}^{-0.5} \cdot \min(\text{step\_num}^{-0.5}, \text{step\_num} \cdot \text{warmup\_steps}^{-1.5})$;
- $\text{warmup\_steps} = 400$;

Regularization

- Residual dropout: $p = 0.1$;
- Label smoothing: $\epsilon_{ls} = 0.1$;

Machine Translation Results

Machine Translation Results

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 [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**	$2.3 \cdot 10^{19}$	

Model Variations

Model Variations

Table 3: Variations on the Transformer architecture. Unlisted values are identical to those of the base model. All metrics are on the English-to-German translation development set, newstest2013. Listed perplexities are per-wordpiece, according to our byte-pair encoding, and should not be compared to per-word perplexities.

	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

Overview

1. Sequence-to-Sequence Background
   - Sequence-to-Sequence Architecture
   - Limitations and Problems
2. Convolutional Sequence to Sequence Learning - ConvS2S
   - Convolution Architecture
   - Experiments
   - Results
3. Attention Is All You Need - Transformer
   - Transformer Architecture
   - Training Setup
   - Results

Thank You!
Questions & Comments?