Attention Is All You Need

Ashish Vaswani, et al

Google

NeurIPS 2017

Abstract

- Prior works for sequence transduction
  - RNN: hard to parallelize due to sequential dependencies
  - CNN: lack of long-range dependencies due to fixed window

- Transformer: New simple architecture
  - It is purely based on attention
  - It can process sequences in parallel
  - It has global context understanding

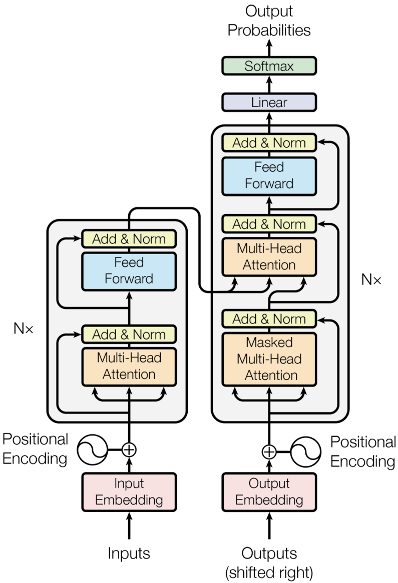
- Performance
  - SOTA on English to German, French tasks
  - It is faster to train, not only better performance
  - It generalizes well to other tasks

Framework

- Encoder stack
  - Input **x** to representation **z**
  - Multihead attention

- Decoder stack
  - Given **z**, output **y**
  - Masked attention
  - Encoder-Decoder attention

- Common component
  - Stack of $N = 6$ identical layers
  - Dimension is $d_{model} = 512$
  - LayerNorm & Residual
  - Fully connected FFN

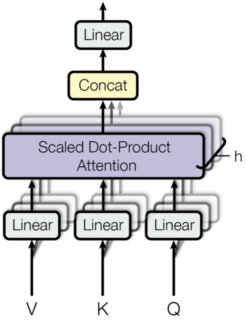


Framework

Word	Query (Q)	Key (K)	Value (V)
	What is it looking for?	What does it represent?	What information does it give?
I	It performed which action?	Subject (pronoun)	"I" (the speaker)
saw	What is being seen?	Verb (past tense of "see")	"saw" (the act of seeing)
a	Is there a noun following?	Article (indicates a noun is next)	"a" (introduces a noun)
blue	What is the adjective describing?	Adjective (describes a noun)	"blue" (describes the bird)
bird	What kind of bird?	Noun (the object)	"bird" (the thing being seen)

- Scaled Dot-Product Attention
  - Query Q, Key K, Value V
  - $\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$
  - Large d_k causes extreme softmax outputs
  - Scaling to prevent vanishing gradient

Framework



- Multi-Head Attention ($h = 8$)
  - $\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, \cdots, head_h)W^O$
  - $head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$
  - Each head focuses on different aspects of relationships

Framework

- Self Attention
  - Helps each word attend to all other words
  - Long-range dependency & Parallelizable

- Masked Self Attention
  - During training, also parallelizable
  - Decoder must only attends to previous tokens
  - Future token attention scores are $-\infty$

- Encoder-Decoder Attention
  - Decoder focus on relevant parts of the input context
  - It acts as a bridge between the encoder and decoder

Framework

- Position-wise Feed-Forward Networks
  - $FFN(x) = ReLU(xW_1 + b_1)W_2 + b_2$
  - Input, Output dimension is 512
  - Inner-layer's dimension is 2048

- Positional Encoding
  - $PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$
  - $PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}})$
  - Transformer doesn't process tokens sequentially
  - It can't distinguish "I go to school" and "to I school go"
  - Learned encoding is slightly better but costly

Why Self-Attention

Type	Computational Complexity	Path Length
RNN	O(n)	O(n)
CNN	O(kn)	O(log n)
Transformer	O(n^2)	O(1)

- Computational Complexity
  - Transformer's matrix multiplications are expensive
  - However, it is parallelizable, making it faster than others
  - When n is very large, restricted window may be used
- Path Length for Long-Range Dependencies
  - Each word attends to every other word directly
- Interpretability
  - Some heads focus on syntax, others focus on semantics

Training

- Dataset
  - English to German translation
  - English to French translation

- Hardware and schedule
  - 8 NVIDIA P100 GPUs (2017 top GPUs)
  - Base model: 100K steps (12 hours)
  - Big model: 300K steps (3.5 days)

Results

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

- Machine translation task
  - Transformer outperformed all previous models, even ensembles
  - Trains much faster than previous models (3.5 days vs. weeks)

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

- Ablation
  - Optimal number of heads exists
  - Higher dimension, Larger model size is better
  - Dropout is effective, Learned PE is not effective

Analysis

Parser	Training	WSJ 23 F1
Vinyals & Kaiser el al. (2014) [37]	WSJ only, discriminative	88.3
Petrov et al. (2006) [29]	WSJ only, discriminative	90.4
Zhu et al. (2013) [40]	WSJ only, discriminative	90.4
Dyer et al. (2016) [8]	WSJ only, discriminative	91.7
Transformer (4 layers)	WSJ only, discriminative	91.3
Zhu et al. (2013) [40]	semi-supervised	91.3
Huang & Harper (2009) [14]	semi-supervised	91.3
McClosky et al. (2006) [26]	semi-supervised	92.1
Vinyals & Kaiser el al. (2014) [37]	semi-supervised	92.1
Transformer (4 layers)	semi-supervised	92.7
Luong et al. (2015) [23]	multi-task	93.0
Dyer et al. (2016) [8]	generative	93.3

- English Constituency Parsing (Analyzing syntax)
  - Small dataset: Wall Street Journal (40K Sentences)
  - Large dataset: Semi-supervised setting (17M Sentences)
  - Close to RNNG, the SOTA [1]

[1] Chris Dyer et al. "Recurrent Neural Network Grammars". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. June 2016.

Conclusion

- Contributions
  - Transformer with only attention mechanisms
  - It is parallelizable thus faster than RNNs and CNNs

- Future Directions
  - Input and output beyond text
  - Restricted attention for large IO
  - Improving Sequential Generation