Attention Is All You Need

Ashish Vaswani[1] **Noam Shazeer**[1] **Niki Parmar**[2] **Jakob Uszkoreit**[2] **Llion Jones**[2]
Aidan N. Gomez[3] **Łukasz Kaiser**[1] **Illia Polosukhin**[2]

[1]Google Brain [2]Google Research [3]University of Toronto

Abstract

Sequence transduction models of the time encoded and decoded text with recurrent or convolutional networks, often joined by an attention mechanism. The Transformer drops recurrence and convolution altogether and builds both encoder and decoder from attention and position-wise feed-forward layers. Because no step waits for the previous one, training parallelizes across the whole sequence. On the WMT 2014 English-to-German task the large model reaches 28.4 BLEU, and on English-to-French 41.8 BLEU, at a fraction of the training cost of earlier systems.

About this edition. A study edition typeset with LeftBlank Office from the paper by Vaswani et al. [1] (arXiv:1706.03762). The text is a summary written for this example, not the original wording; the equations follow the paper, and Tables 1 and 2 reproduce its Tables 1 and 2, which Google permits for journalistic and scholarly use with attribution. Read the original for the full argument.

1 Introduction

Recurrent networks, long short-term memory [2] in particular, were the standard tool for language modelling and machine translation. A recurrent model reads a sentence one position at a time: the hidden state at position t depends on the state at $t-1$. That chain is what makes these models hard to parallelize within a training example, and memory limits how many examples fit in a batch.

Attention lets a model relate positions regardless of their distance [3], but it was almost always used alongside a recurrent network. The Transformer asks what happens without the recurrence: every layer relates all positions to all others through attention alone.

2 Background

Convolutional models such as ByteNet [4] and ConvS2S [5] also compute hidden representations for all positions in parallel. In them, the number of operations needed to relate two positions grows with their distance, linearly for ConvS2S and logarithmically for ByteNet. In the Transformer it is a constant number of operations. The price is that averaging over attention-weighted positions blurs resolution, which multi-head attention (Section 3.2) counteracts.

3 Model architecture

The model keeps the familiar encoder–decoder shape (Figure 1). The encoder maps an input sequence of symbols $(x_1, ..., x_n)$ to continuous representations $\boldsymbol{z} = (z_1, ..., z_n)$. Given $\boldsymbol{z}$, the decoder generates the output $(y_1, ..., y_m)$ one symbol at a time, feeding back the symbols it has already produced.

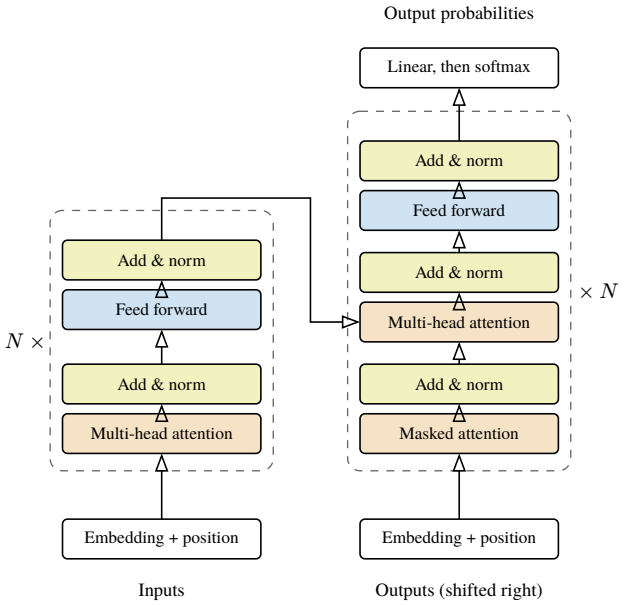


Figure 1: The Transformer, redrawn after Figure 1 of the paper. Left: the encoder, a stack of $N = 6$ identical layers. Right: the decoder, whose middle sub-layer attends over the encoder output.

3.1 Encoder and decoder stacks

Each encoder layer has two sub-layers: multi-head self-attention, then a position-wise feed-forward network. Each sub-layer is wrapped in a residual connection [6] followed by layer normalization [7], so its output is $\mathrm{LayerNorm}(x + \mathrm{Sublayer}(x))$. All sub-layers and embeddings produce vectors of dimension $d_{\text{model}} = 512$ so that the residual sums line up.

Decoder layers add a third sub-layer that attends over the encoder output. Their self-attention is masked so that position i can only see positions before it; with outputs shifted right by one, each prediction depends only on what has already been generated.

3.2 Attention

An attention function maps a query and a set of key–value pairs to a weighted sum of the values, with weights given by how well the query matches each key. The Transformer uses *scaled dot-product attention* over matrices of queries Q, keys K and values V:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{1}$$

Without the factor $\frac{1}{\sqrt{d_k}}$, dot products grow with the key dimension d_k and push the softmax into regions with very small gradients.

Rather than one attention over full-width vectors, the model projects queries, keys and values h times with learned matrices and attends in each projection in parallel:

$$\begin{aligned} \mathrm{MultiHead}(Q, K, V) &= \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O \\ \mathrm{head}_i &= \mathrm{Attention}\big(QW_i^Q, KW_i^K, VW_i^V\big) \end{aligned} \tag{2}$$

With $h = 8$ heads and $d_k = d_v = \frac{d_{\text{model}}}{h} = 64$, the total cost is close to that of a single head at full width, yet each head can attend to a different kind of relation.

3.3 Position-wise feed-forward networks

Every layer also applies the same two-layer network to each position separately:

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{3}$$

The inner layer is wider, $d_{\text{ff}} = 2048$.

3.4 Positional encoding

Attention by itself ignores word order, so the model adds a positional encoding to the input embeddings. The paper uses sinusoids of geometrically spaced wavelengths, from 2π to $10000 \cdot 2\pi$:

$$\mathrm{PE}_{\text{pos},2i} = \sin\!\left(\frac{\text{pos}}{10000^{2i/d_{\text{model}}}}\right), \quad \mathrm{PE}_{\text{pos},2i+1} = \cos\!\left(\frac{\text{pos}}{10000^{2i/d_{\text{model}}}}\right) \tag{4}$$

For a fixed offset k, $\mathrm{PE}_{\text{pos}+k}$ is a linear function of PE_{pos}, which the authors expected to make relative positions easy to learn. Figure 2 plots three of the dimensions.

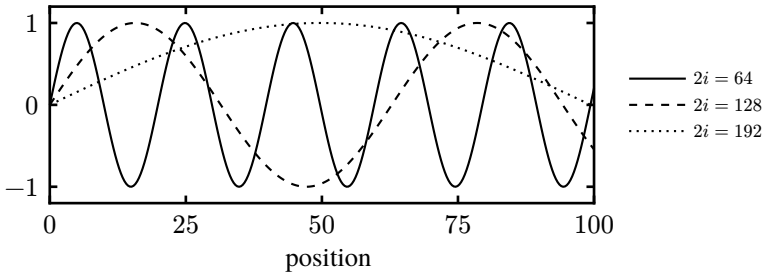


Figure 2: Three even dimensions of the sinusoidal positional encoding with $d_{\text{model}} = 512$. Lower dimensions oscillate faster, so together they give each position a distinct pattern.

4 Why self-attention

Table 1 compares a self-attention layer with recurrent and convolutional layers on three counts: the total computation per layer, how much of it must run in sequence, and the longest path a signal travels between two positions. Self-attention connects any two positions in a constant number of steps and runs in parallel, and it is cheaper than a recurrent layer whenever the sequence length n is smaller than the representation dimension d, which is typical of sentence-level translation.

Table 1: Per-layer cost for sequence length n, representation dimension d, kernel size k and neighbourhood size r. Reproduced from Table 1 of [1].

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)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

5 Training

The models were trained on WMT 2014 English–German (about 4.5 million sentence pairs) and English–French (36 million). The base model ran for 100,000 steps, about 12 hours, on one machine with eight NVIDIA P100 GPUs; the big model ran for 300,000 steps, 3.5 days. Optimization used Adam [8] with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\varepsilon = 10^{-9}$, and a learning rate that rises linearly for the first $\text{warmup} = 4000$ steps and then decays with the inverse square root of the step number:

$$\text{lrate} = d_{\text{model}}^{-0.5} \cdot \min\!\left(\text{step}^{-0.5}, \text{step} \cdot \text{warmup}^{-1.5}\right) \tag{5}$$

Regularization combined dropout [9] with $P_{\text{drop}} = 0.1$ on every sub-layer output and on the embedding sums, and label smoothing [10] with $\varepsilon_{\text{ls}} = 0.1$.

6 Results

Table 2 sets the Transformer against the best published systems of the time. The big model scores above every earlier model, ensembles included, on both tasks, while its training cost is a small fraction of theirs.

Table 2: BLEU on newstest2014 and estimated training cost. Reproduced from Table 2 of [1].

	BLEU		Training cost (FLOPs)	
Model	EN–DE	EN–FR	EN–DE	EN–FR
ByteNet [4]	23.75			
Deep-Att + PosUnk		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [11]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [5]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [12]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk ensemble		40.4		$8.0 \cdot 10^{20}$
GNMT + RL ensemble	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S ensemble	26.36	41.29	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base)	27.3	38.1	$3.3 \cdot 10^{18}$	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

7 Conclusion

The Transformer showed that attention alone, without recurrence or convolution, is enough for state-of-the-art translation, and that dropping the sequential dependency makes training much faster. The authors pointed to other modalities, such as images, audio and video, and to less sequential generation as next steps.

References

[1] A. Vaswani *et al.*, "Attention Is All You Need," in *Advances in Neural Information Processing Systems 30*, 2017.

[2] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *International Conference on Learning Representations*, 2015.

[4] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. van den Oord, A. Graves, and K. Kavukcuoglu, "Neural Machine Translation in Linear Time," *arXiv preprint arXiv:1610.10099*, 2016.

[5] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional Sequence to Sequence Learning," in *International Conference on Machine Learning*, 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[7] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[8] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations*, 2015.

[9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[11] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, and others, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," *arXiv preprint arXiv:1609.08144*, 2016.

[12] N. Shazeer *et al.*, "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer," in *International Conference on Learning Representations*, 2017.