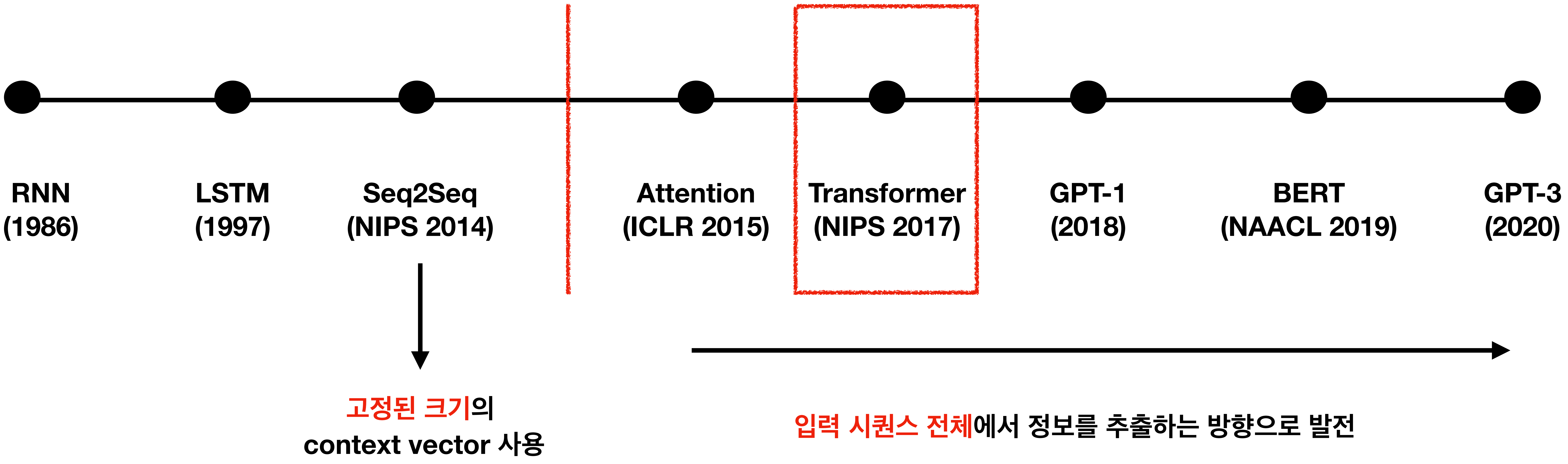
Attention is all you need

Google Brain
University of Toronto
(Nips 2017)

Jun-Hyung Lee

Development of Machine Translation



Self-Attention

1. Long term dependency problem

2. Computational complexity

Intro

- The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder

- We propose a new simple network architecture, the Transformer, based soley on attention mechanisms, dispensing with recurrence and convolution entirely.

Background -seq2seq

- Recurrent models typically factor computation along the symbol positions of the input and output sequences. Aligning the positions to steps in computation time, they generate a sequence of hidden states h_t, as a function of the previous hidden state ht-1 and the input for position t

- This inherently, sequential nature precludes parallelization within training examples, which becomes critical at longer sequence lengths, as memory constraints limit batching across examples.

Background -Attention Mechanism

- Attention mechanisms have become an **integral** part of **compelling** sequence modeling and transduction models in various tasks, allowing modeling of dependencies without regard to their distance in the input or output sequences.

- In all but a few cases, however such attention mechanisms are used in conjunction with a recurrent network

Self-Attention

1. Long term dependency problem

2. Computational complexity

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

- Different positions of a single sequence in order to compute a representation of the sequence

Summary -Transformer

- Transformer, a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output.

- The Transformer allows for significantly more parallelization and can reach a new state of the art in translation quality after being trained for as little as tweleve hours on eight P100 GPUs.

Transformer Architecture

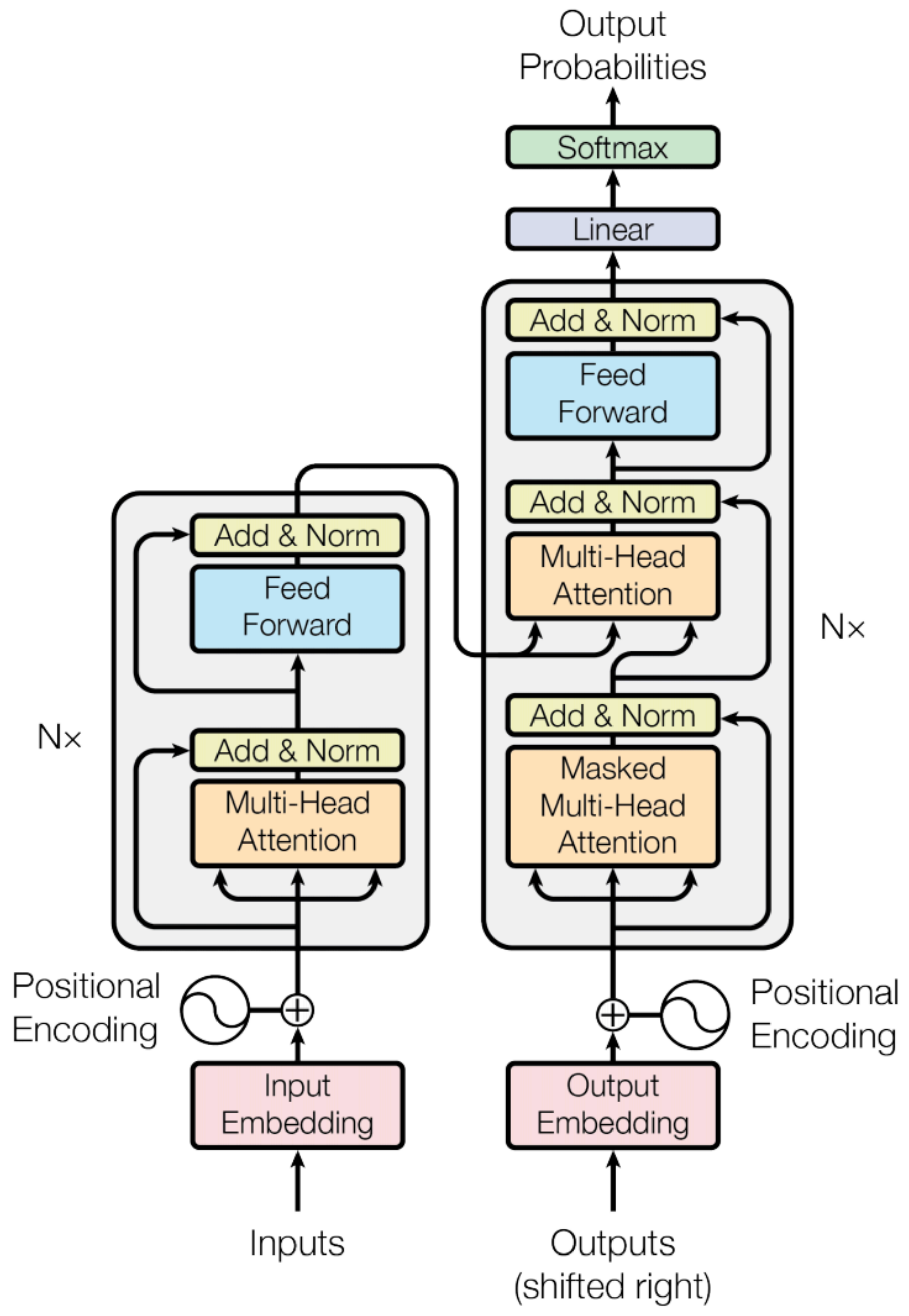


Figure 1: The Transformer - model architecture.

- Encoder
- Multi-Head Attention
- Decoder
- Positional Encoding
- Add & Norm

Attention

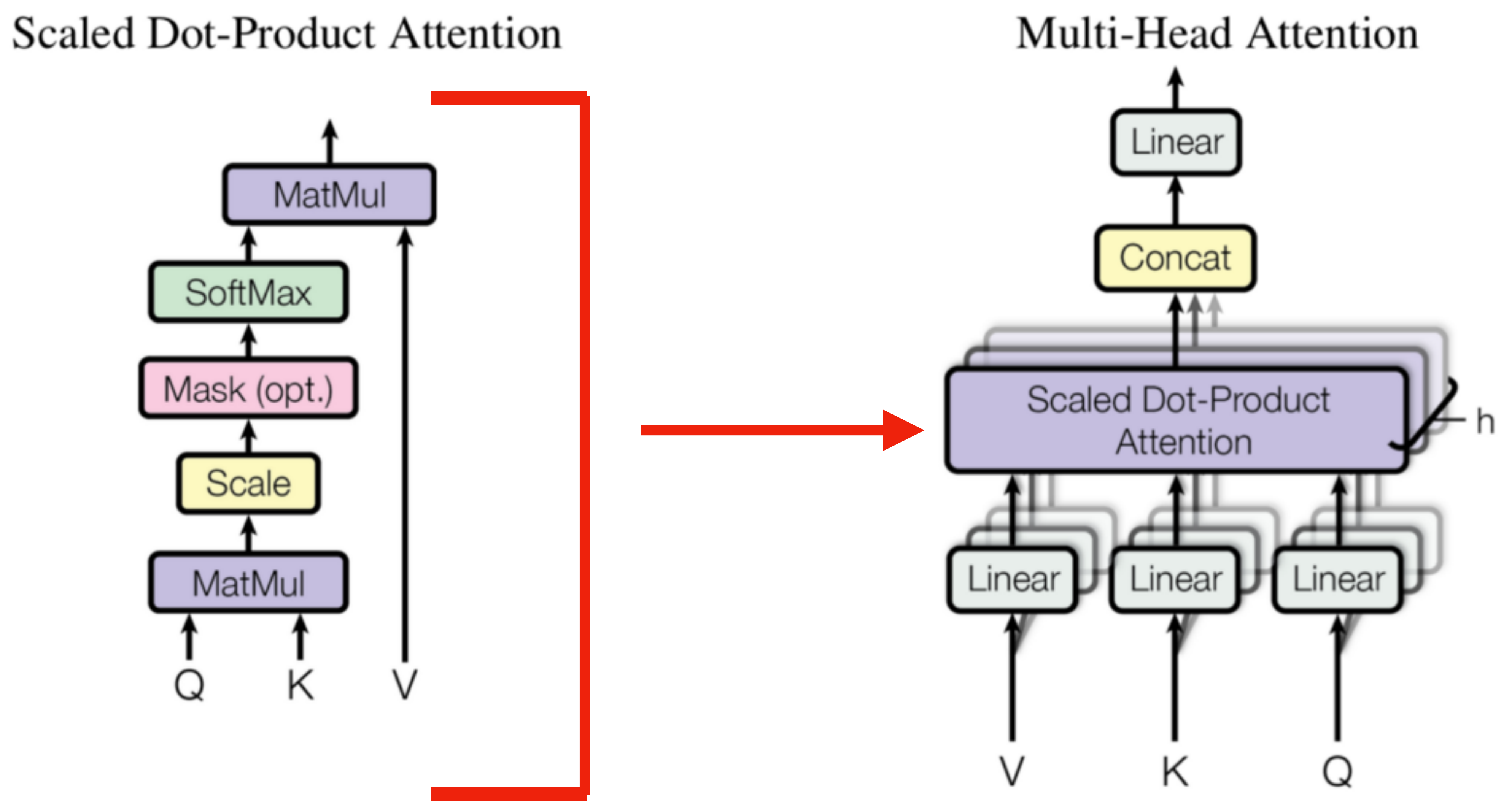


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

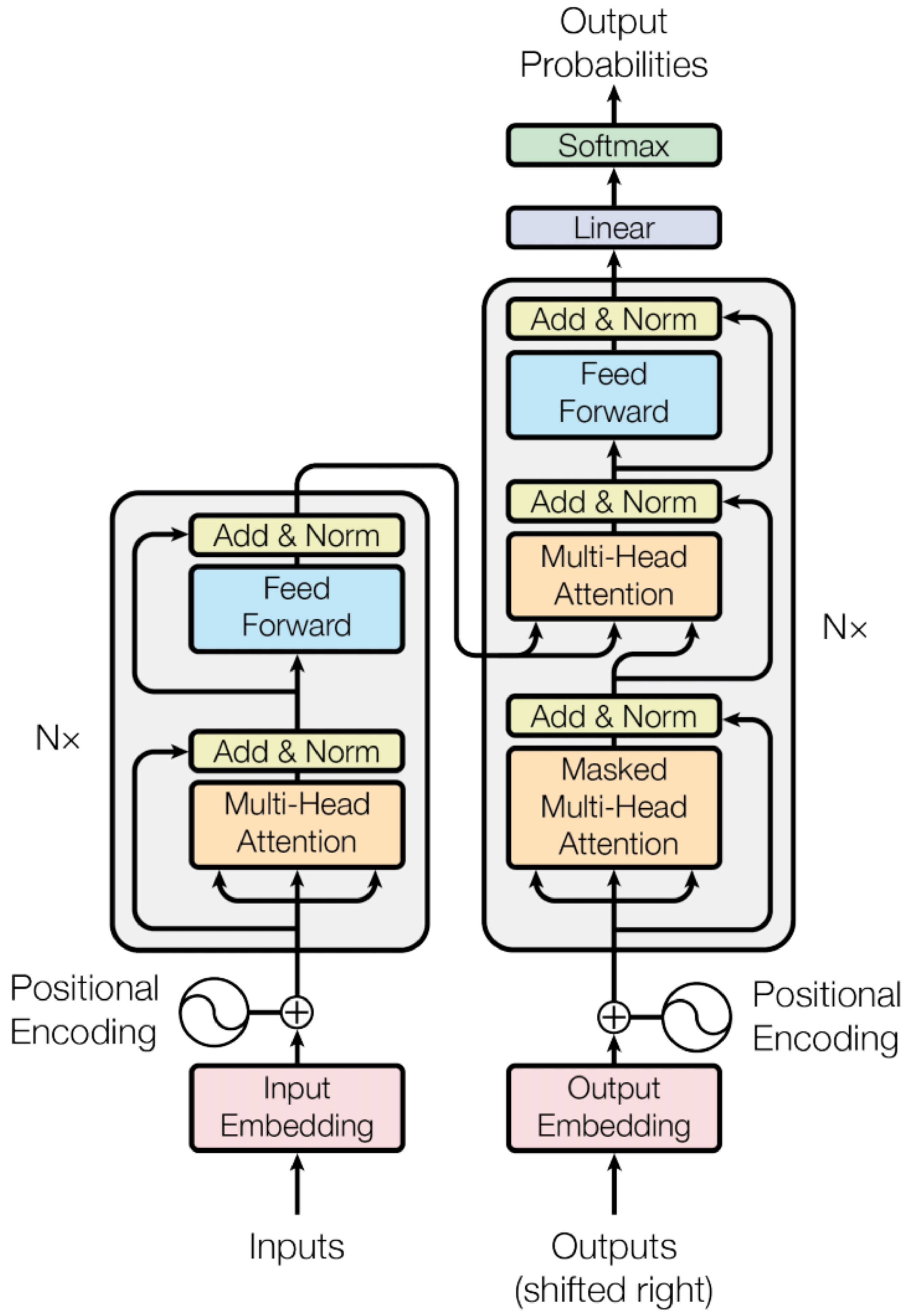
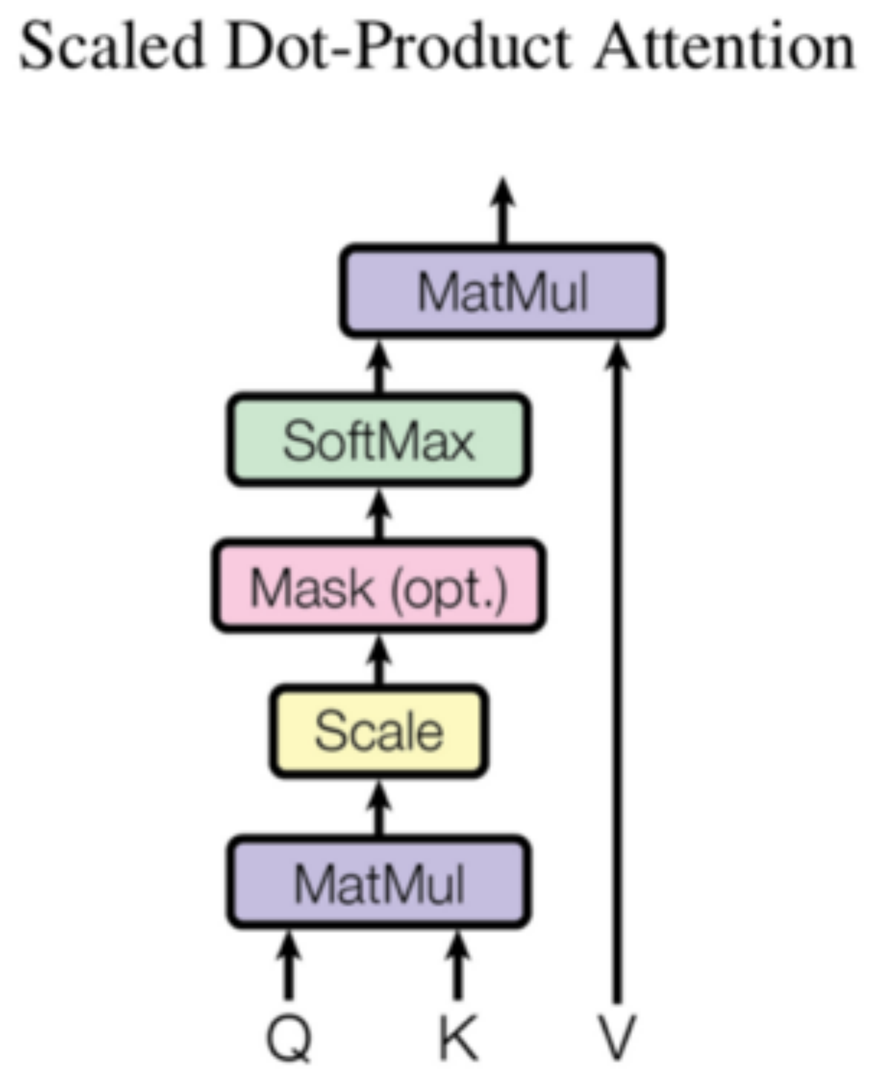
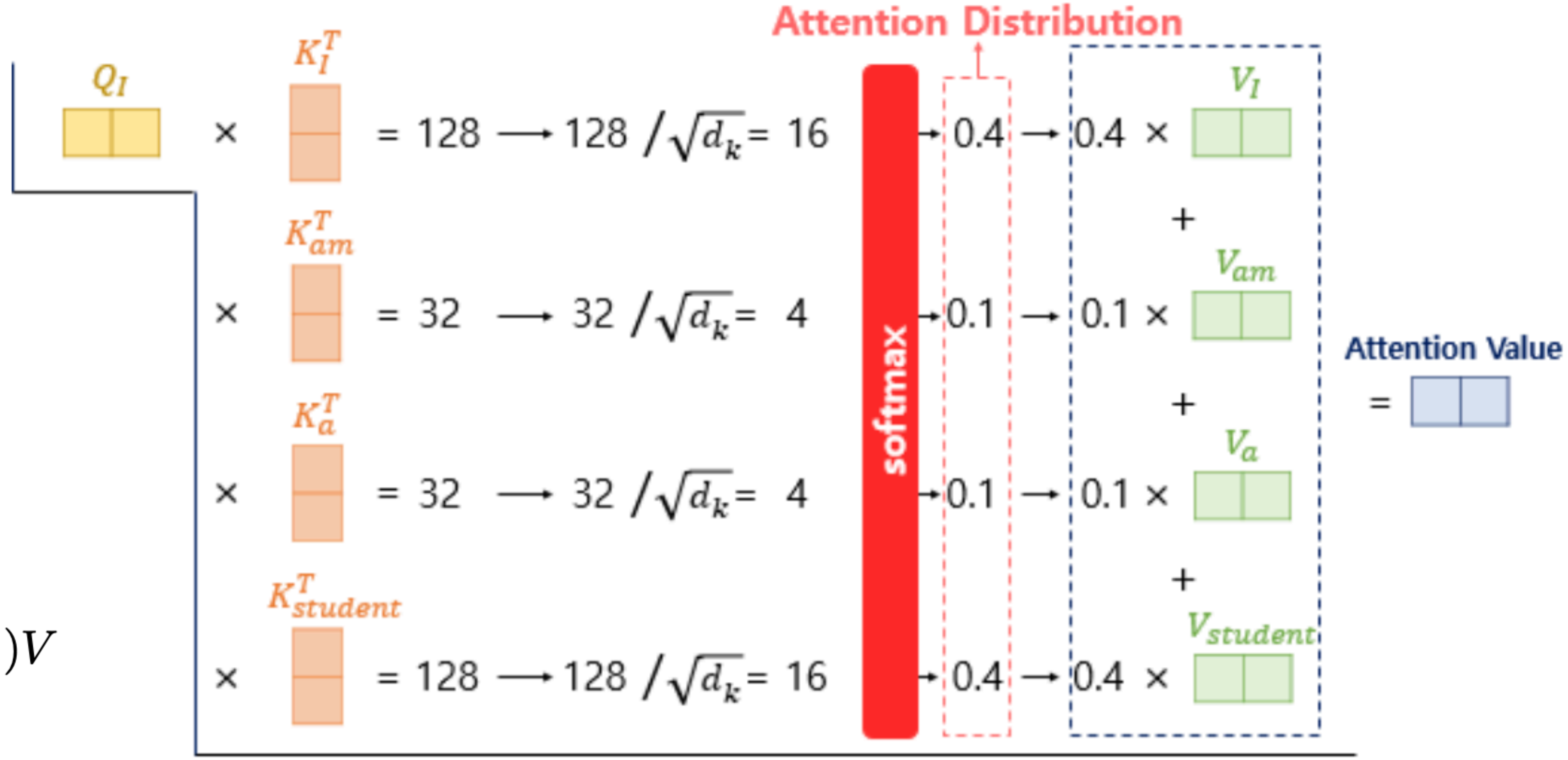


Figure 1: The Transformer - model architecture.

Scaled Dot-Product Attention



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



Scaled Dot-Product Attention

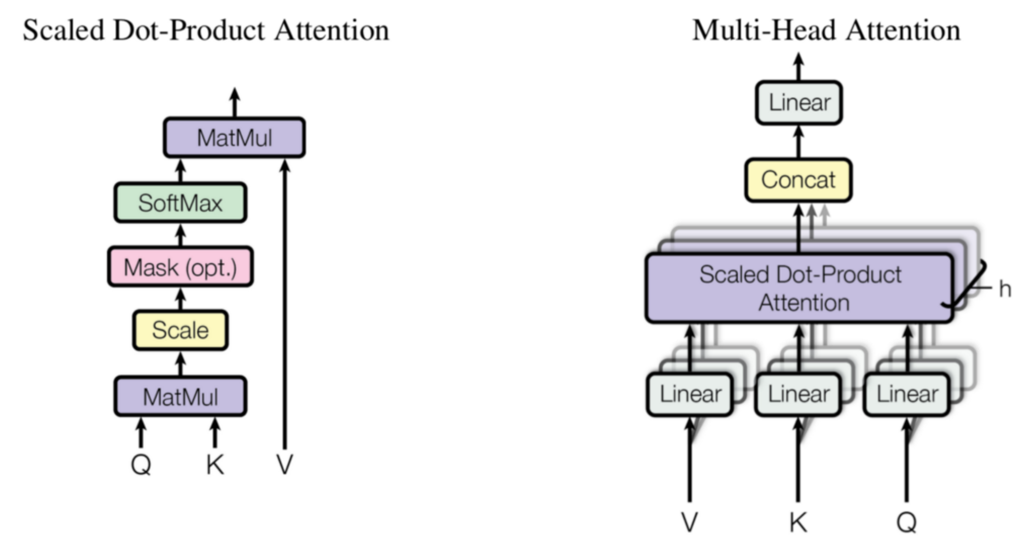
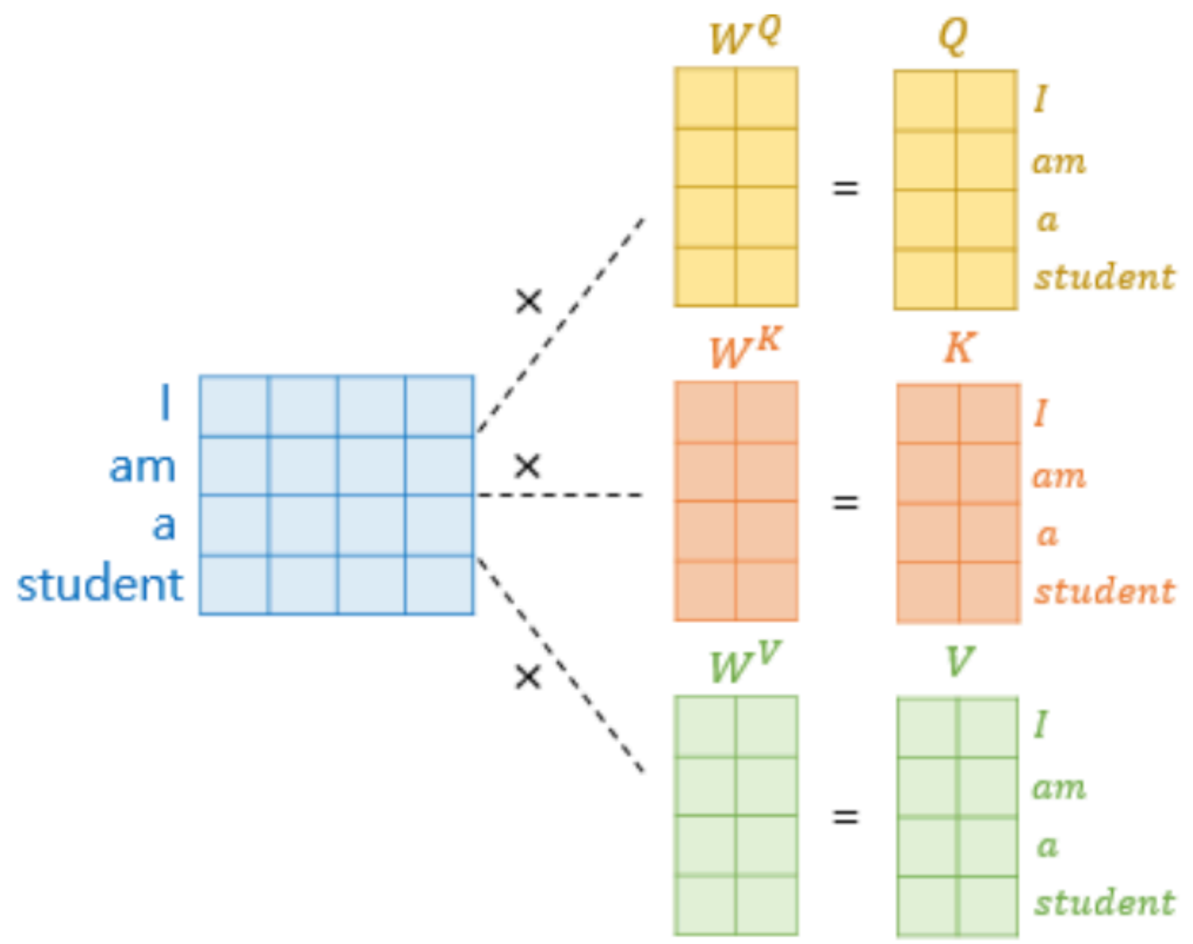
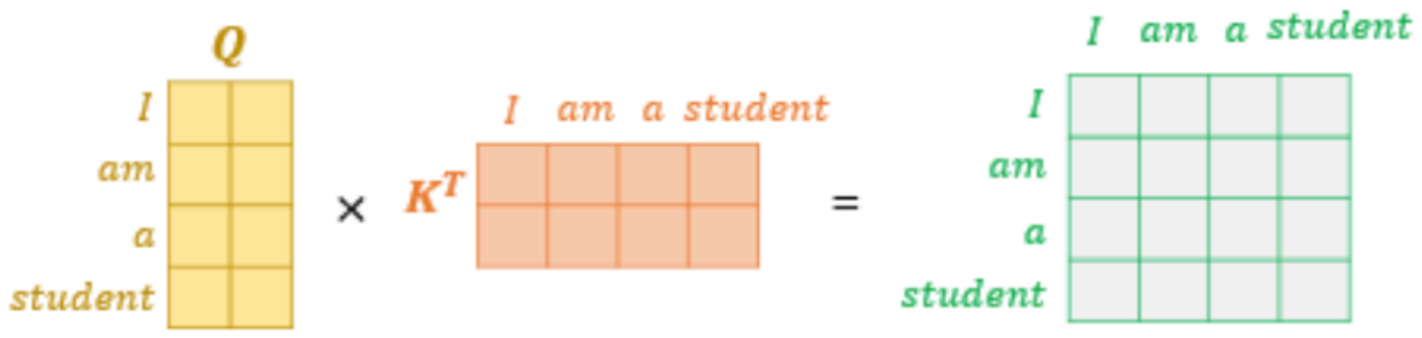
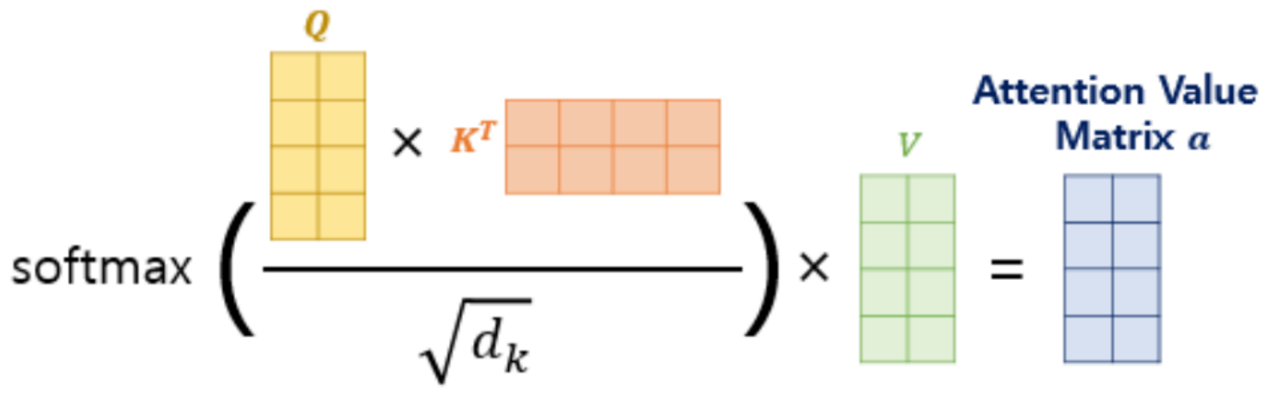


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

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



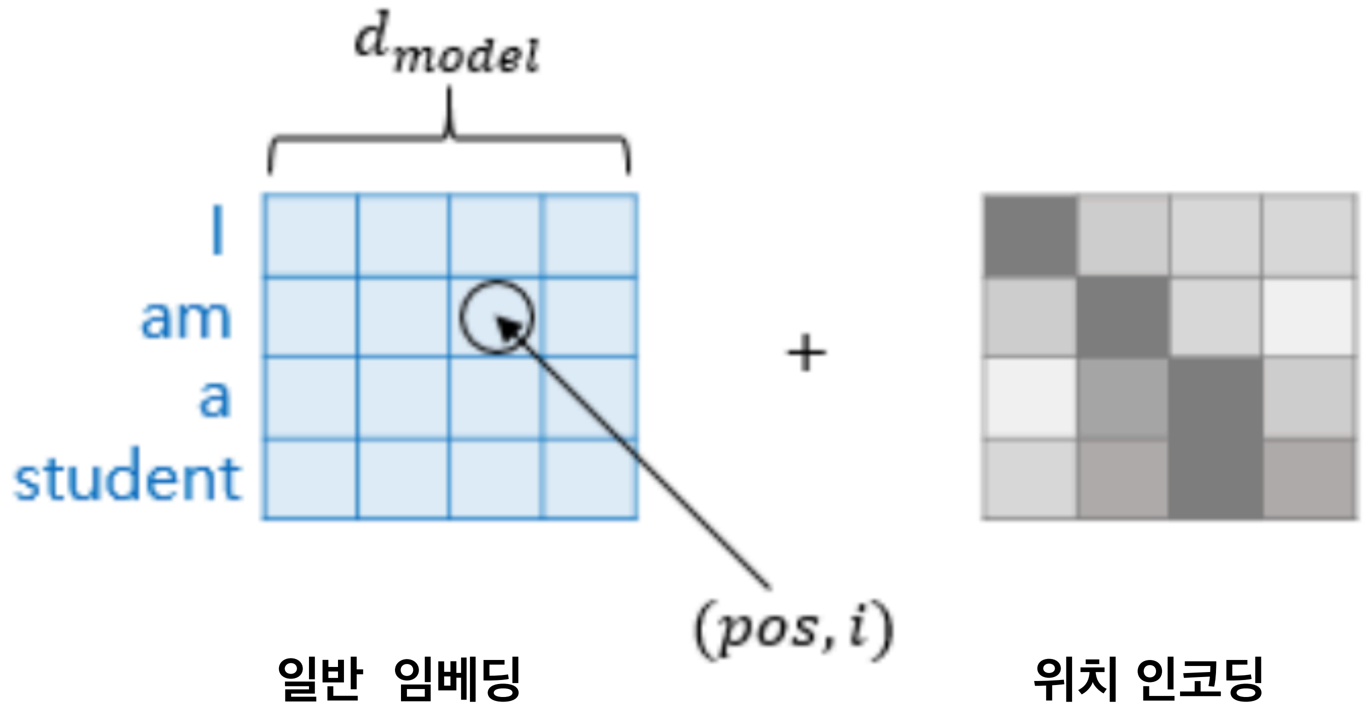




- We suspect that for large values of d_k, the dot products grow large in magnitude, pushing the softmax function into regions where it has extremely small gradients. To counteract this effect, we scale the dot products.

Positional Encoding-Why sinusoid?

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

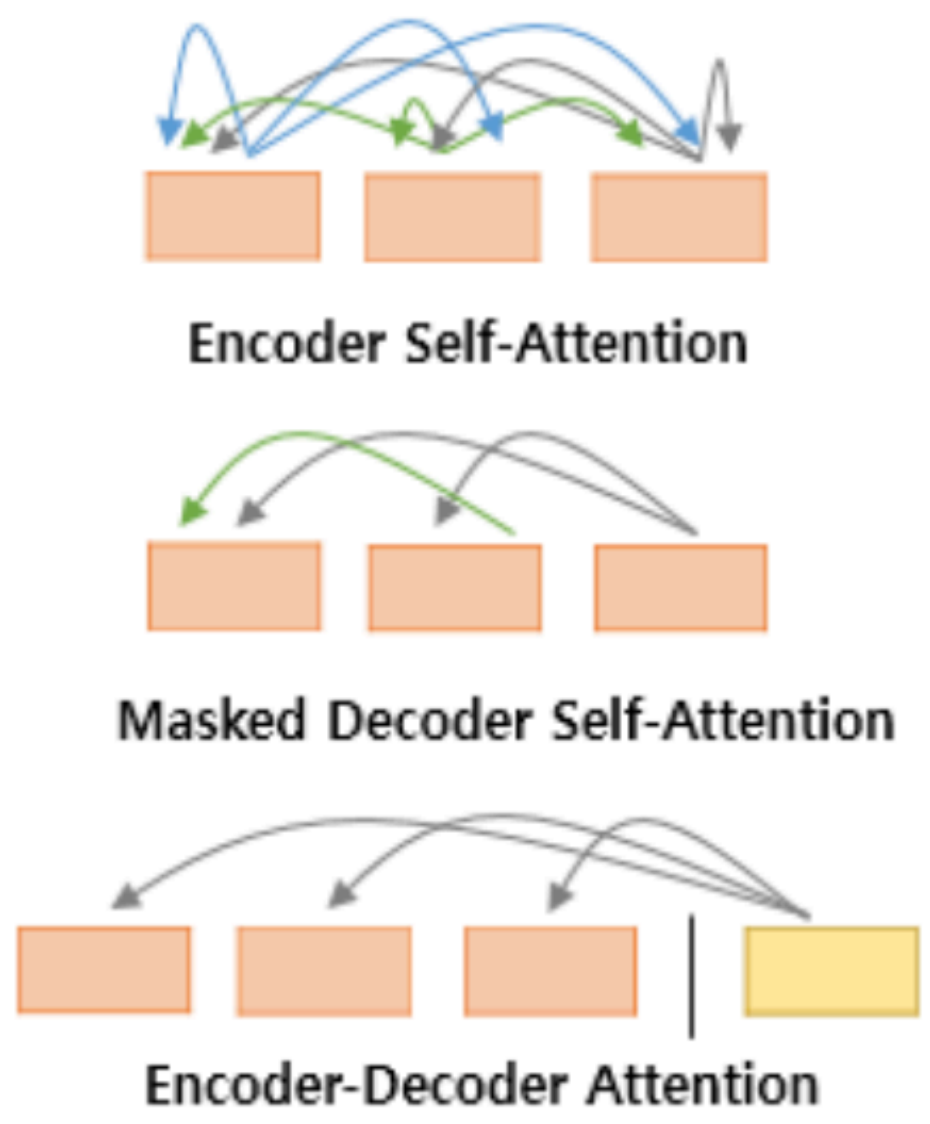


- We chose the sinusoidal version because it may allow the model to extrapolate to sequence lengths longer than the ones encountered during training

- pairwise distance - by how far two positions are apart in a sequence

- learned position embeddings also can be used

Three Attention Layers



1. Encoder : Query = Key = Value
2. Decoder : Query = Key = Value
3. Decoder Vector: Query / Encoder Vector : Key = Value

Attention Score Matrix

- Masked Decoder Self-Attention
- This masking, combined with fact that the output embeddings are offset by one position, ensures that the predictions for position i can depend only on the known outputs at position less than i

Add&Norm / Position-wise FFNN

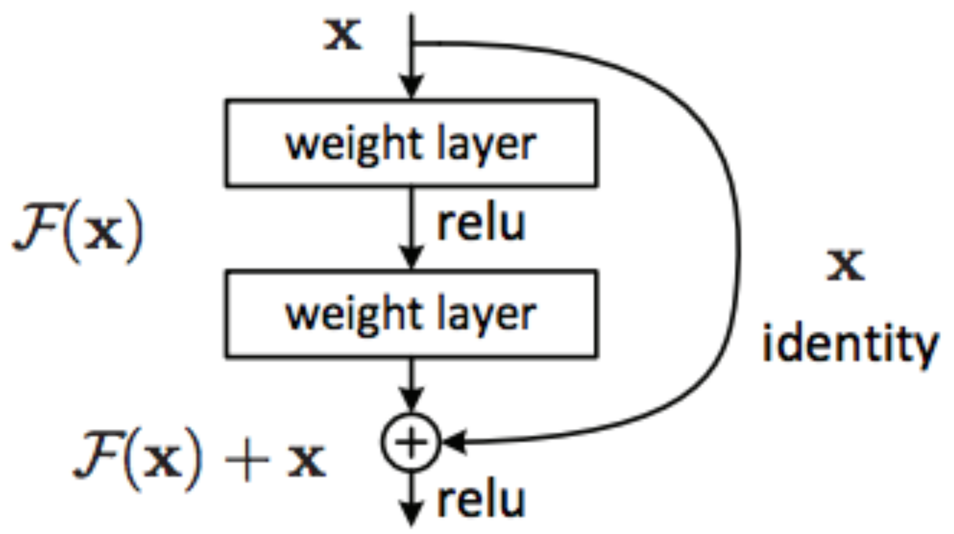


Figure 2. Residual learning: a building block.

“Resnet eases the optimization by providing faster convergence at the early stage”

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

- Fully connected feed-forward network
- Two linear transformations with a ReLU activation in between
- While the linear transformations are the same across different positions, they use different parameters from layer to layer

Experiment

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

- Dataset: WMT 2014 English-German dataset(4.5M sentences, 37000 vocab)
- Batch size = 25000
- Hardware = 8 P100 GPU
- Optimizer = Adam
- 12 hours of training
- Label Smoothing

Conclusion

- Transformer, the first sequence transduction model based entirely on attention, replacing the recurrent layers most commonly used in encoder-decoder architectures with multi-headed self-attention

- For translation tasks, the Transformer can be trained significantly faster than architectures based on recurrent or convolutional layers.