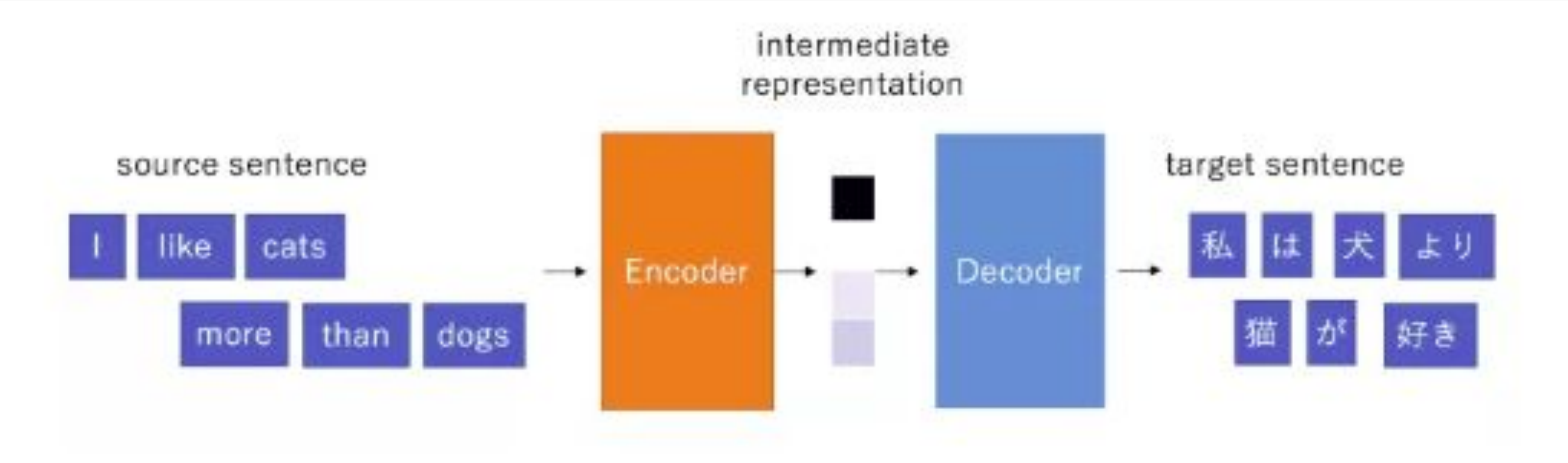
Transformer architecture + convolutional architecture

by Sriyal Jayasinghe, Maxi Fischer, Sophie Gräfnitz

Attention is all you need

by Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones

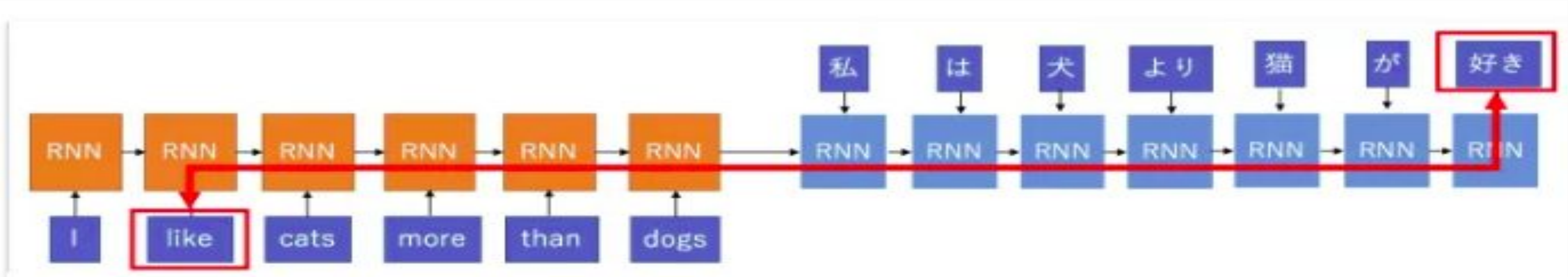
Modern NMT Architecture



src = [Attention is all you need explained]

Limitations

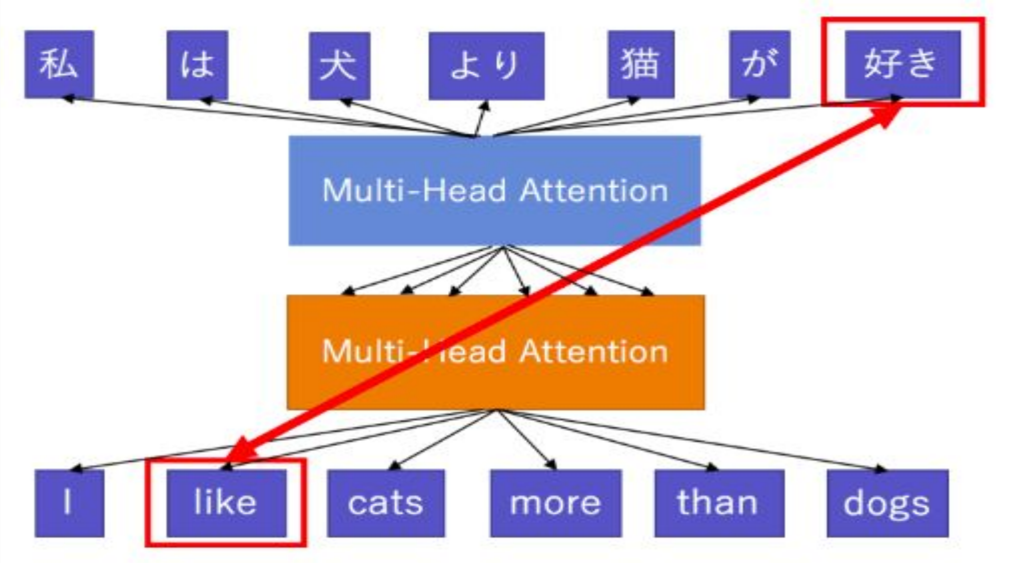
- Sequential processing
  - GPUs work far better for parallel inputs
- Difficulty of learning long-range dependencies



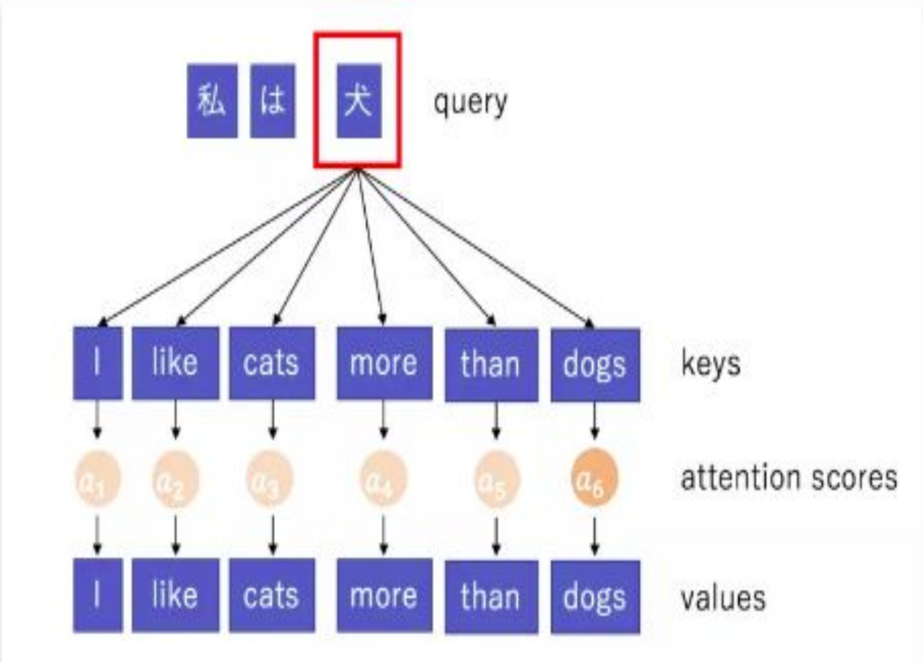
src = [Attention is all you need explained]

The Proposed Model

- Directly learn the dependencies using the attention mechanism
- Processes all the tokens in parallel and learns to “attend” only the relevant information



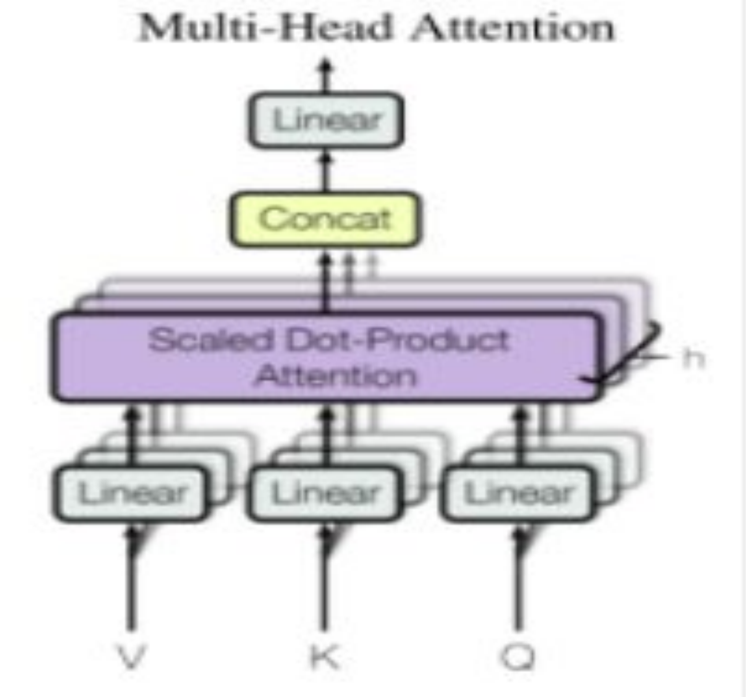
Attention Mechanism



- computes the relevance of a set of **values**(information) based on some **keys** and **queries**
- Uses "Scaled Dot-Product Attention"

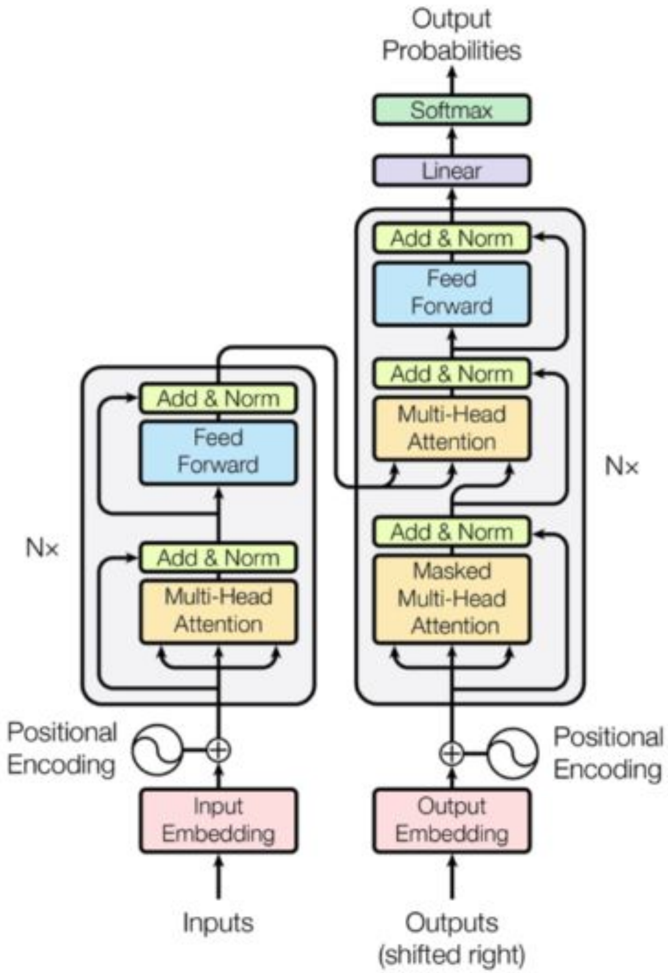
$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Multi-Head Attention



- With a single attention it is difficult to captapplies different linear

- Applies different linear transformations



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

Convolutional Sequence to Sequence Learning

by Jonas Gehring, Michael Auli, David Grangster, Denis Yarats, Yans D. Dauphin

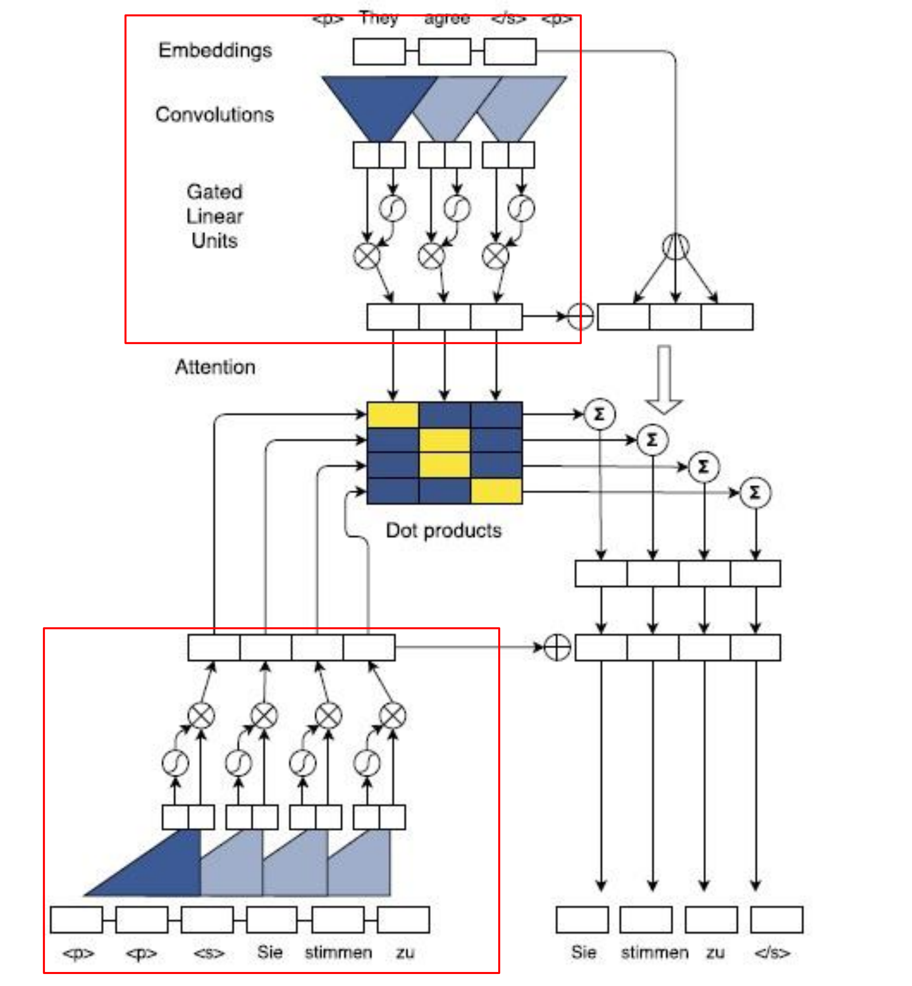
Introduction

- published 25.07.2017
- fully convolutional model (encoding and decoding)
- new level of performance in several benchmark tests
- several advantages:
  - speed (parallelizable),
  - use of compositional language structure

Convolutional NN vs RNN

- **Convolutional:**
- context is always represented in fixed size
- convolutions and pooling do not depend on previous computations
- hierarchical representation
- for a word distance of n, kernel size k, O(n/k) steps needed

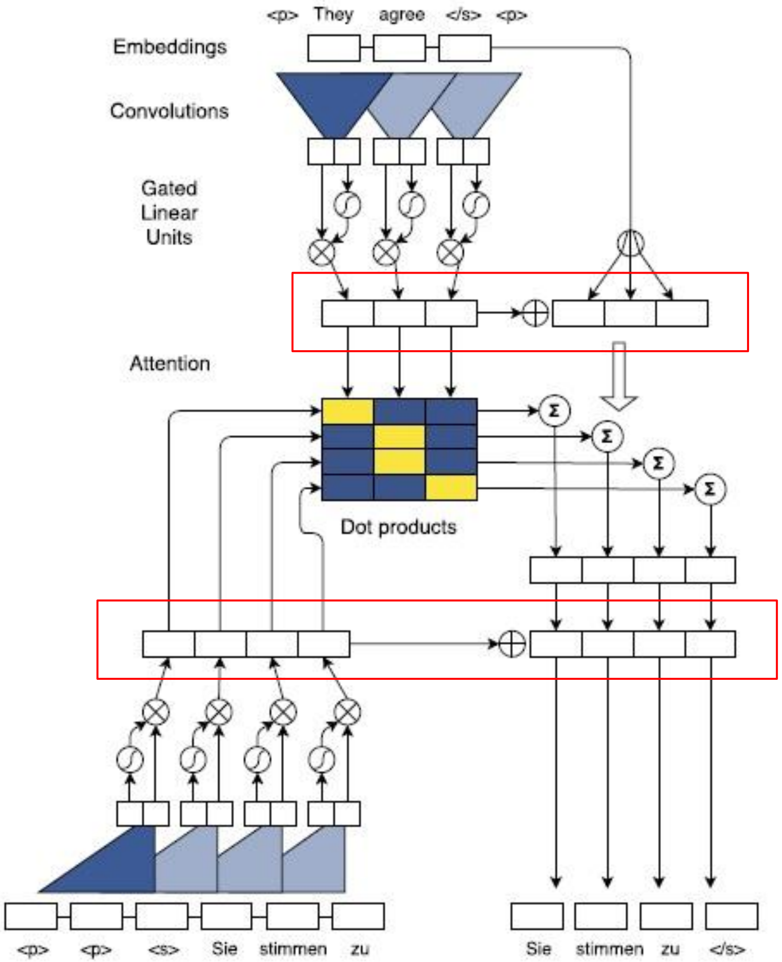
- **Recurrent NN**
- size of context depends on sentence length
- hidden states build up context consecutively
- chain representation
- O(n)



source: *Convolutional Sequence to Sequence Learning, Jonas Gehring et al.*

Convolutional Block Structure

- input is embedded in distributional space and combined with positional embeddings
- each block (= layer) contains 1-D convolution and
- **GLU** (Gated linear outputs):
  - input $Y = [AB] \in \mathbb{R}^{2d}$, $A, B \in \mathbb{R}^{d}$
  - output $v([AB]) = A \times \sigma(B)$
  - with σ the nonlinear gate function
  - $\times$ the elementwise multiplication)

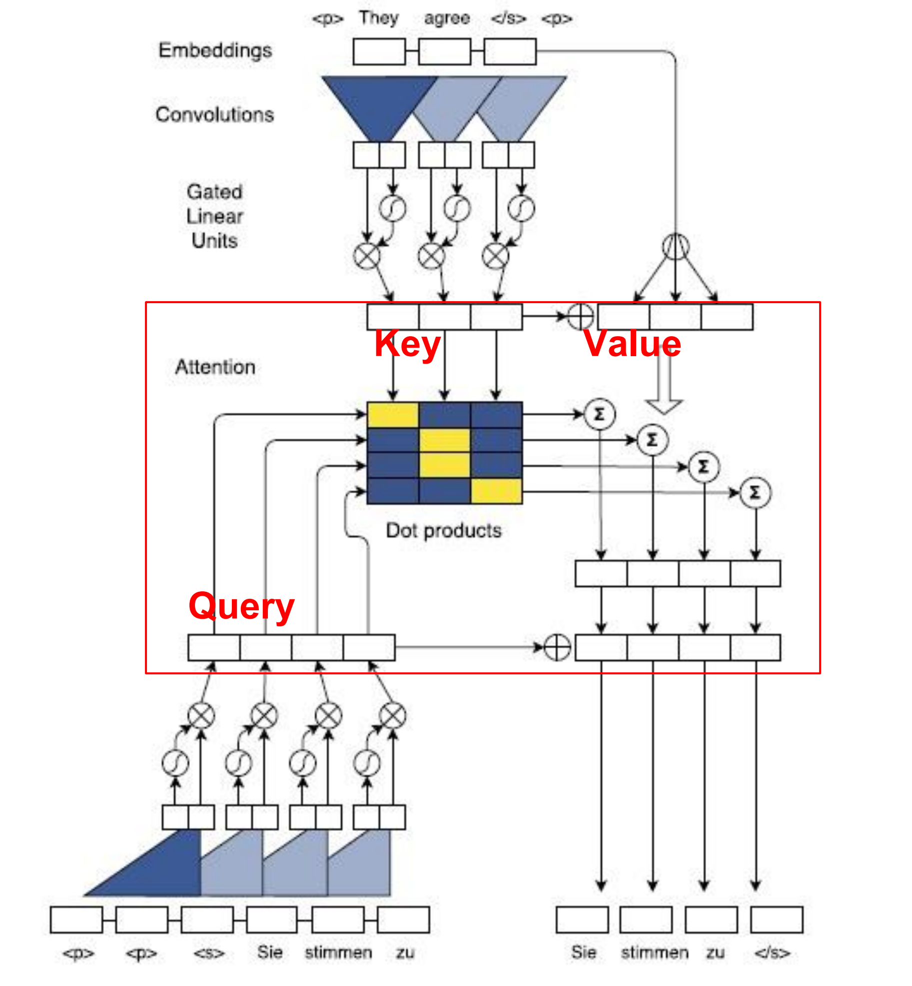


source: *Convolutional Sequence to Sequence Learning, Jonas Gehring et al.*

Residual connection

- output of a block consists of convolution output and input to convolution (first box: encoder, second box: decoder)
- this “skip connection” influences gradient without need of passing a non-linear gate function
- easier optimization and reduction of training error

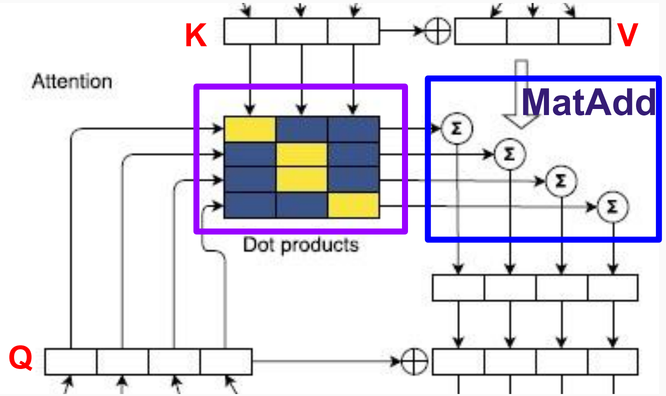
Source: Deep Neural Networks with Residual Connections for Precipitation Forecasting, M.Nguyen et al., Nov. 2017, Vietnam



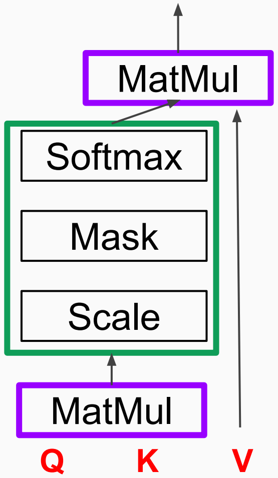
source: *Convolutional Sequence to Sequence Learning, Jonas Gehring et al.*

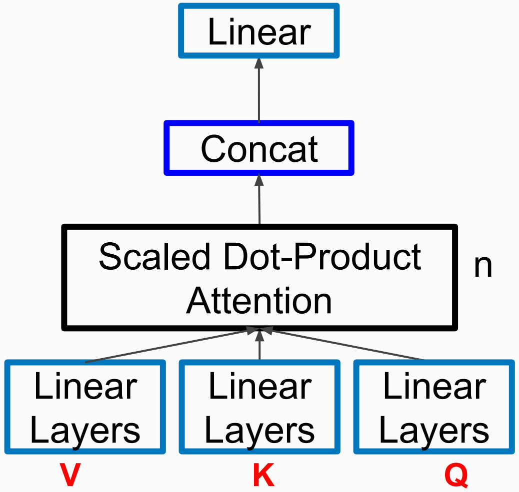
Multi-Step Attention vs. Multi-Head Attention

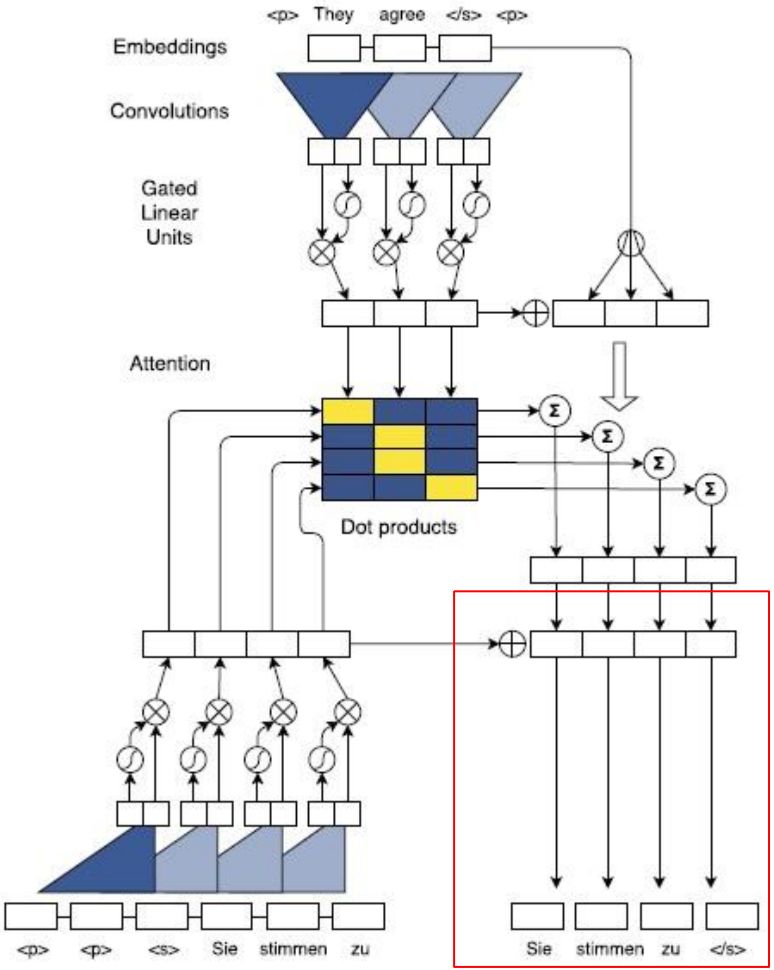
- additive vs. multiplicative addition



vs.







source: *Convolutional Sequence to Sequence Learning, Jonas Gehring et al.*

Output Generation

- goal: maintain activation variance throughout whole network
- actions:
  - normalization of input and output of residual connections
  - output is fed back to attention mechanism
- prediction of target words:

$$p(y_{i+1}|y_1, \ldots, y_i, \mathbf{x}) = \mathrm{softmax}(W_o h_i^L + b_o) \in \mathbb{R}^T$$

Benchmark Results

WMT'14 English-German	BLEU
Wu et al. (2016) GNMT	26.20
Wu et al. (2016) GNMT + RL	26.30
ConvS2S	26.43

WMT'14 English-French	BLEU
Zhou et al. (2016)	40.4
Wu et al. (2016) GNMT	40.35
Wu et al. (2016) GNMT + RL	41.16
ConvS2S	41.44
ConvS2S (10 models)	41.62

Table 2. Accuracy of ensembles with eight models. We show both likelihood and Reinforce (RL) results for GNMT; Zhou et al. (2016) and ConvS2S use simple likelihood training.

Kernel width	Encoder layers		
	5	9	13
3	20.61	21.17	21.63
5	20.80	21.02	21.42
7	20.81	21.30	21.09

Table 7. Encoder with different kernel width in terms of BLEU.

Kernel width	Decoder layers		
	3	5	7
3	21.10	21.71	21.62
5	21.09	21.63	21.24
7	21.40	21.31	21.33

Table 8. Decoder with different kernel width in terms of BLEU.