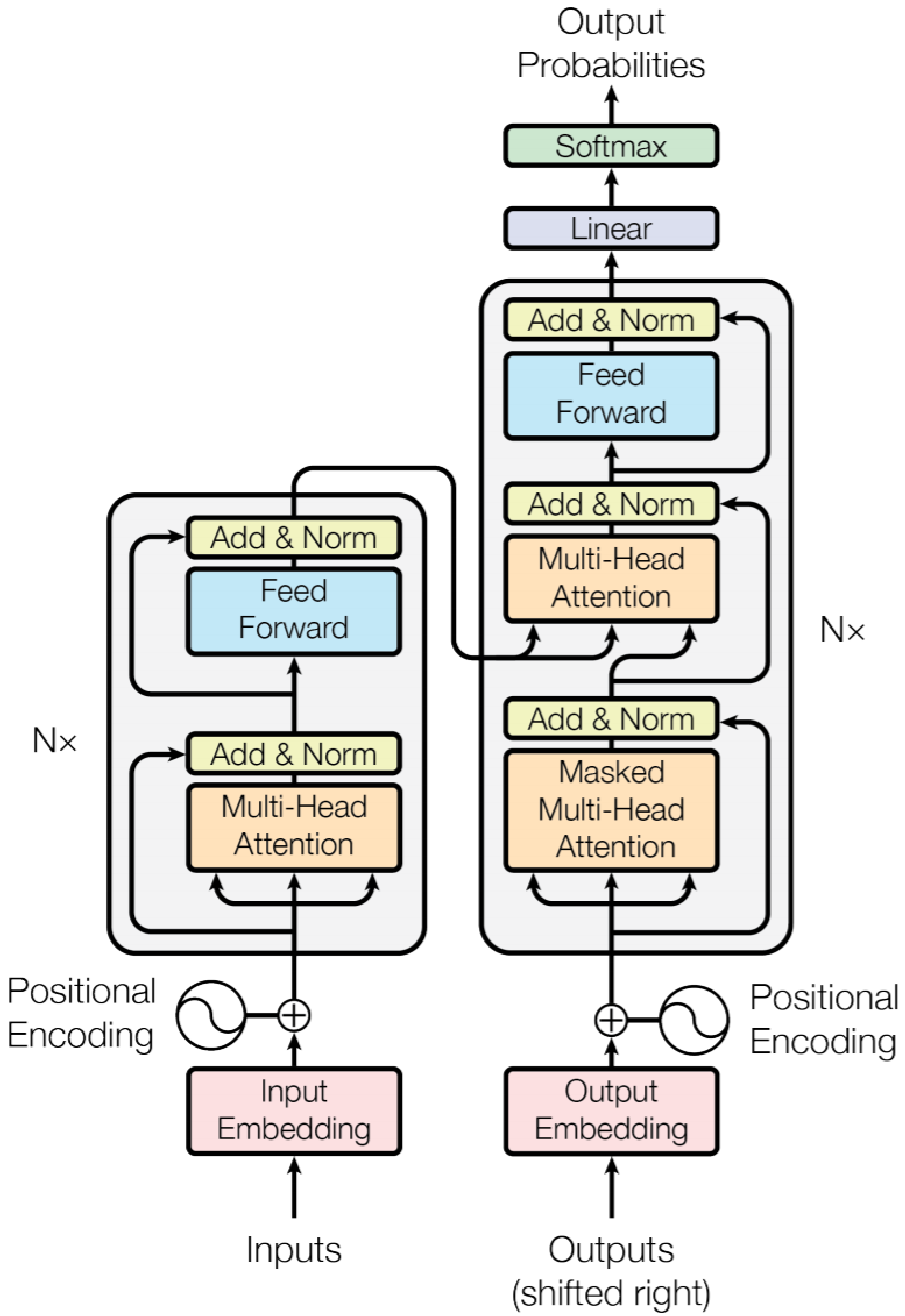
Sequence Encoding
Transformer

國立臺灣大學 資訊工程學系
陳縕儂 助理教授
http://vivianchen.idv.tw

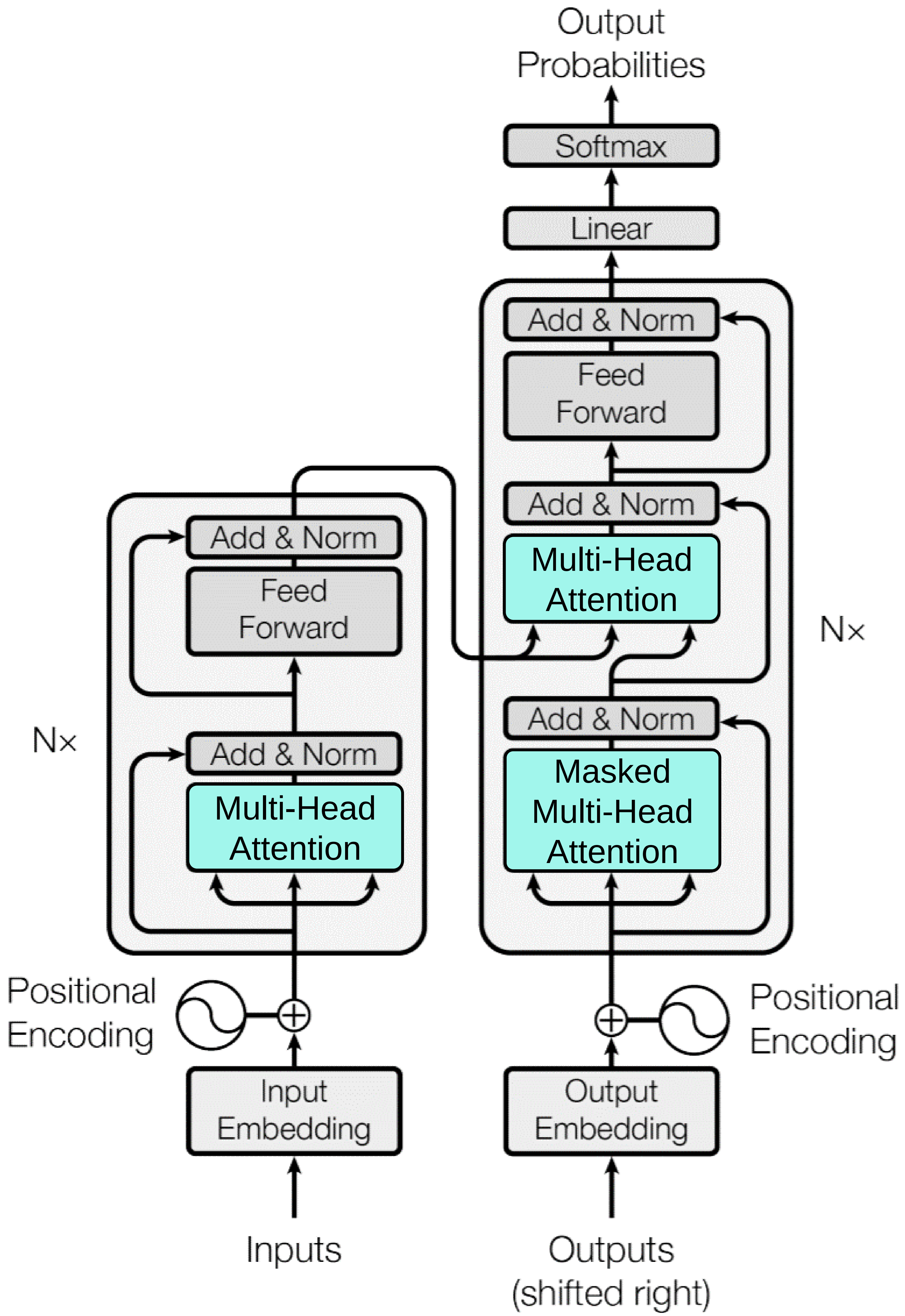
Transformer Overview

- Non-recurrent encoder-decoder for MT
- PyTorch explanation by Sasha Rush
  http://nlp.seas.harvard.edu/2018/04/03/attention.html



Transformer Overview

- Non-recurrent encoder-decoder for MT
- PyTorch explanation by Sasha Rush
  http://nlp.seas.harvard.edu/2018/04/03/attention.html



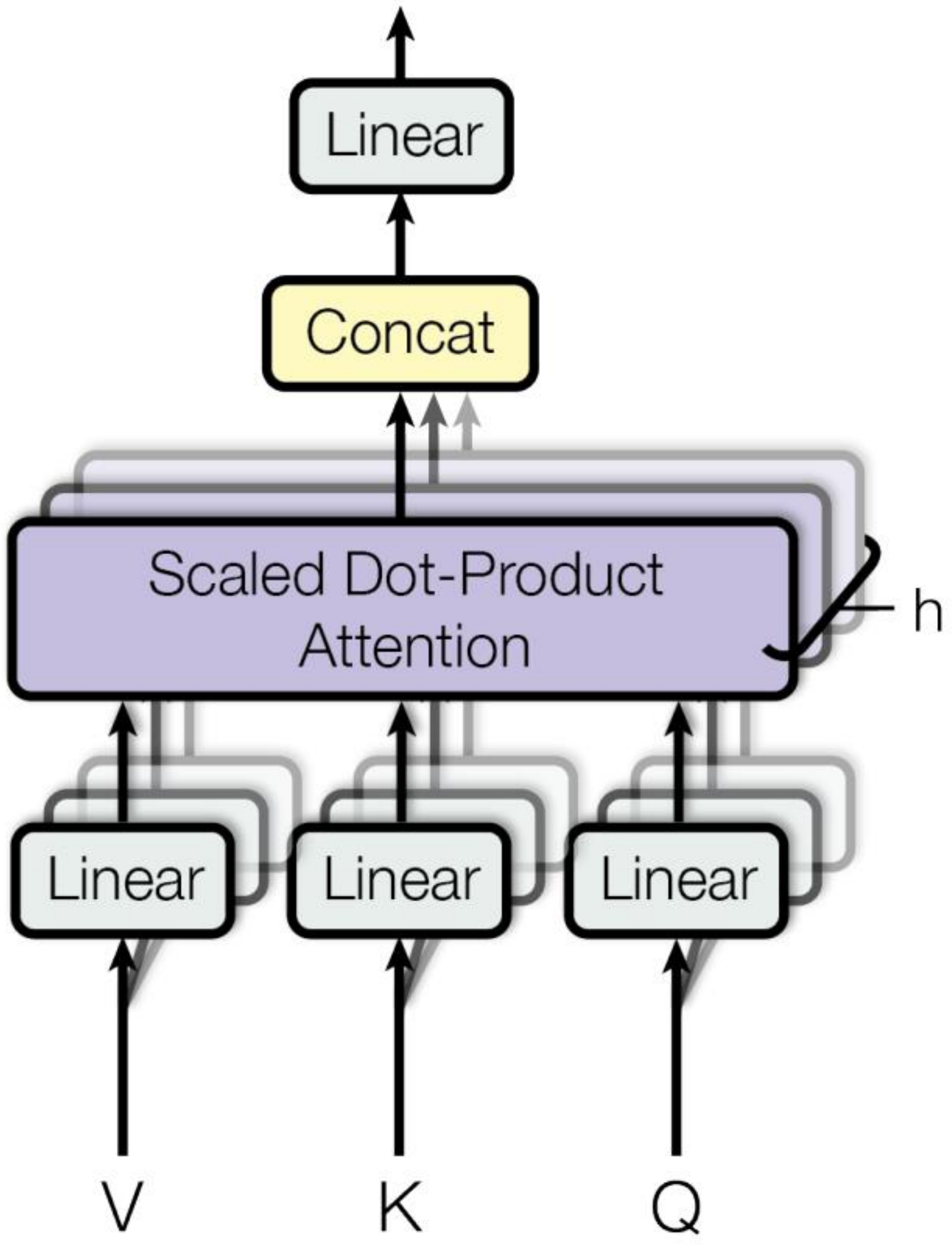
Vaswani et al., “Attention Is All You Need”, in *NIPS*, 2017.

Multi-Head Attention

- Idea: allow words to interact with one another
- Model
  - Map V, K, Q to lower dimensional spaces
  - Apply attention, concatenate outputs
  - Linear transformation

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \cdots, \mathrm{head}_h)W^O$$

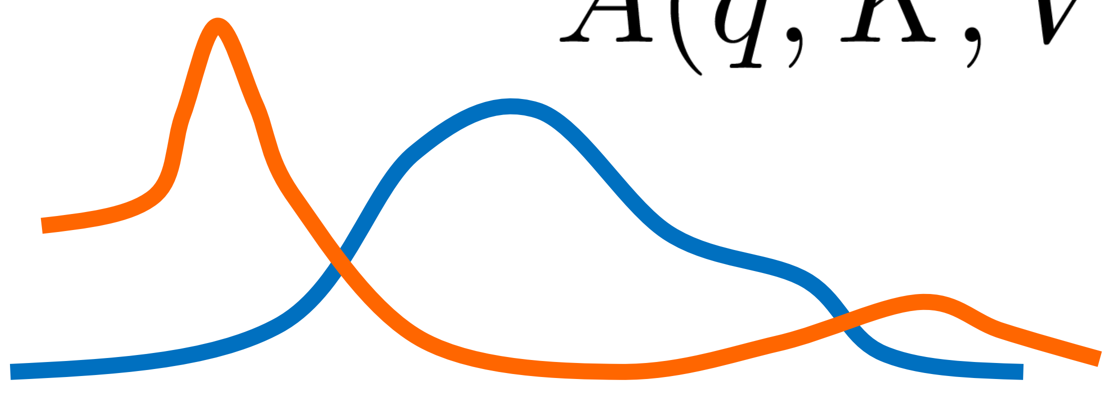
$$\mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



Vaswani et al., “Attention Is All You Need”, in *NIPS*, 2017.

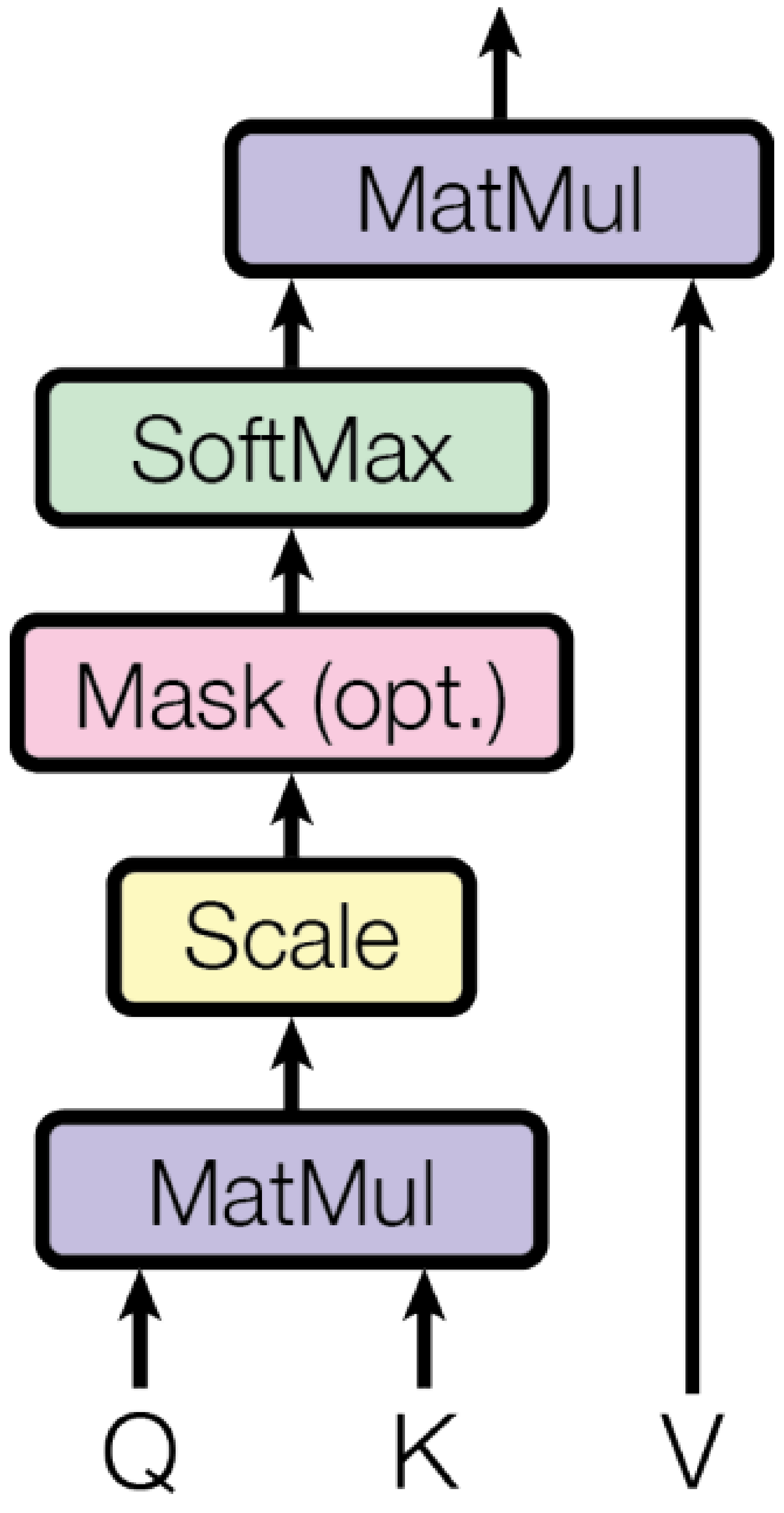
Scaled Dot-Product Attention

- Problem: when d_k gets large, the variance of $q^T k$ increases
  - → some values inside softmax get large
  - → the softmax gets very peaked
  - → hence its gradient gets smaller
- Solution: scale by length of query/key vectors



$$A(q, K, V) = \sum_i \frac{\exp(q \cdot k_i)}{\sum_j \exp(q \cdot k_j)} v_i$$

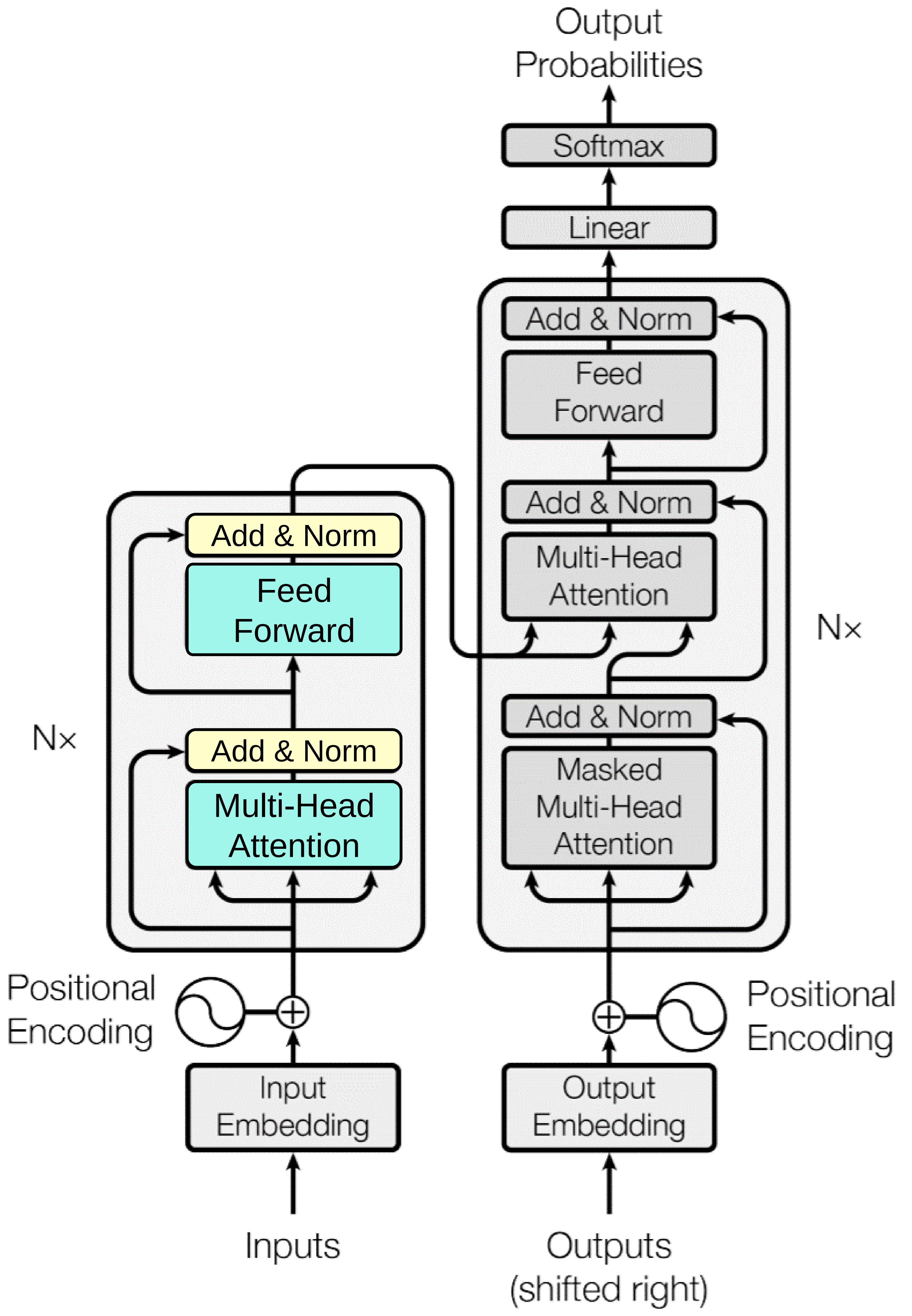
$$A(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



Vaswani et al., "Attention Is All You Need", in *NIPS*, 2017.

Transformer Overview

- Non-recurrent encoder-decoder for MT
- PyTorch explanation by Sasha Rush
  http://nlp.seas.harvard.edu/2018/04/03/attention.html



Vaswani et al., “Attention Is All You Need”, in *NIPS*, 2017.

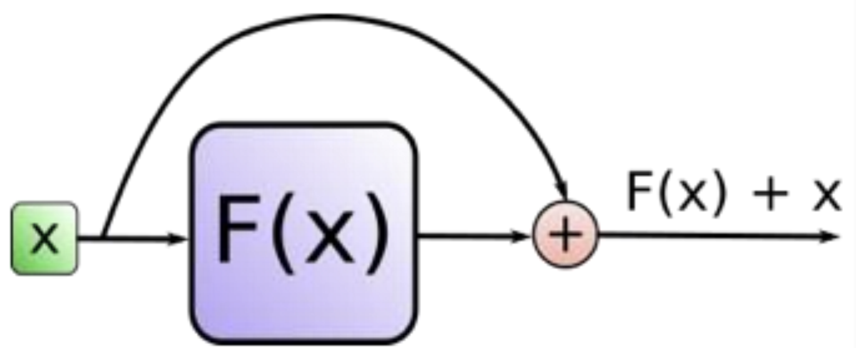
Transformer Encoder Block

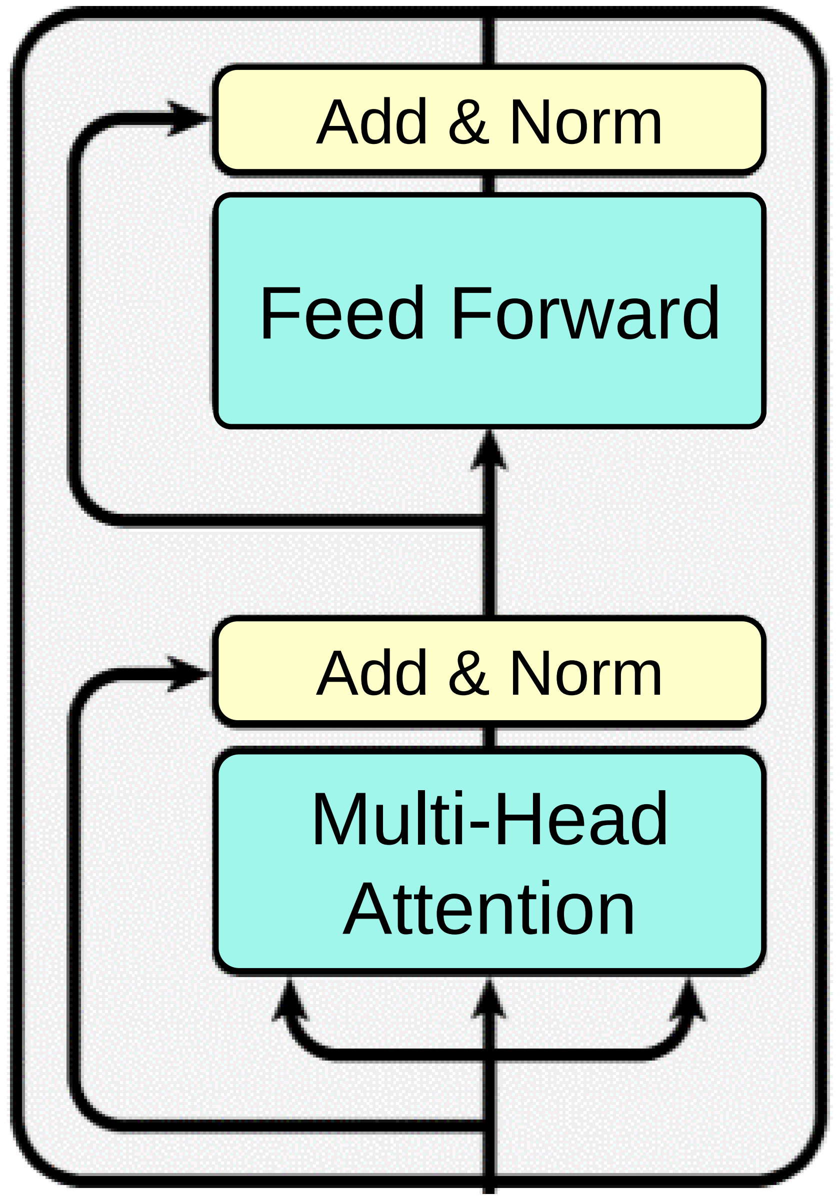
- Each block has
  - multi-head attention
  - 2-layer feed-forward NN (w/ ReLU)
- Both parts contain
  - Residual connection
  - Layer normalization (LayerNorm)

    Change input to have 0 mean and 1 variance per layer & per training point

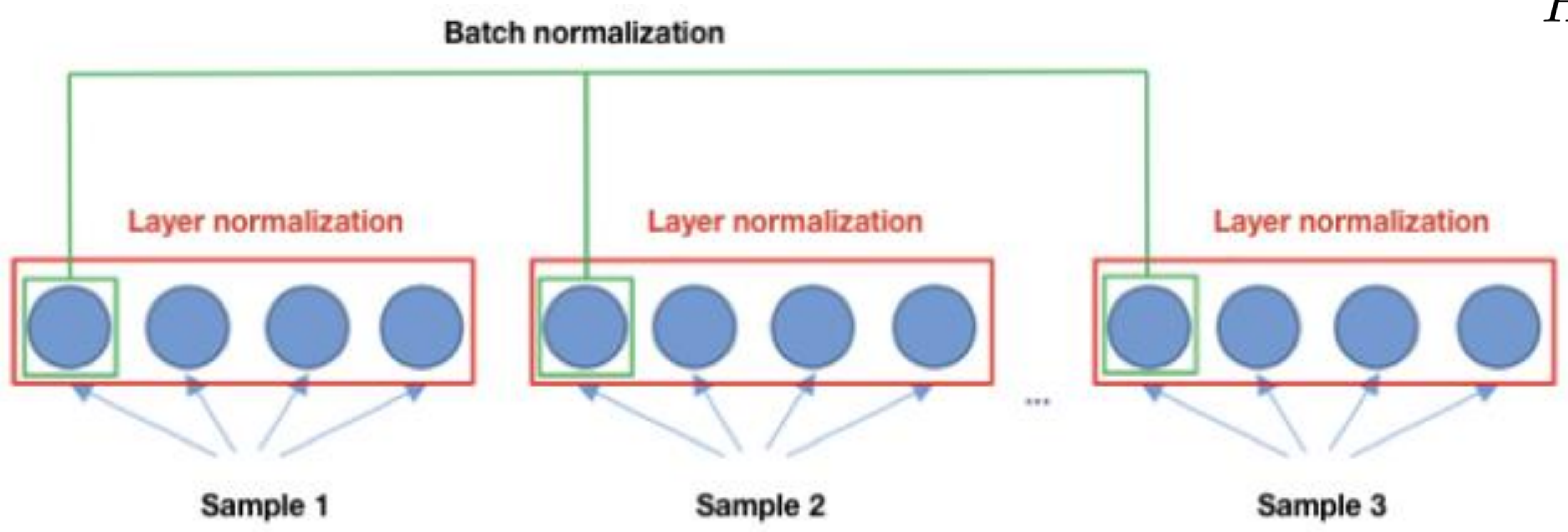
→ LayerNorm(x + sublayer(x))

$$H(x) = g(x) = x + F(x)$$





$$\mu^l = \frac{1}{H}\sum_{i=1}^{H} a_i^l \qquad \sigma^l = \sqrt{\frac{1}{H}\sum_{i=1}^{H}(a_i^l - \mu^l)^2} \qquad h_i = f\left(\frac{g_i}{\sigma_i}(a_i - \mu_i) + b_i\right)$$



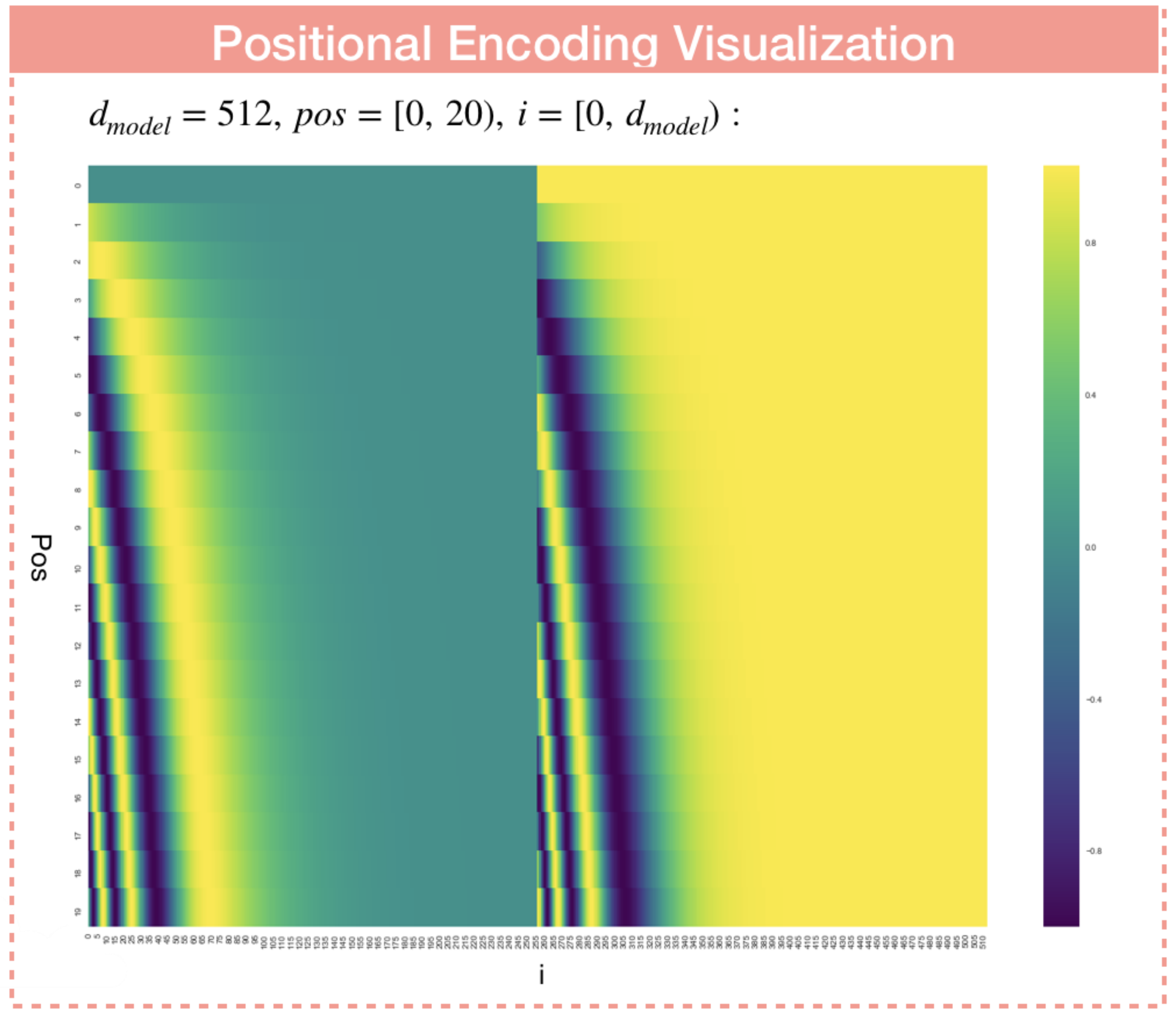
https://medium.com/@bgg/seq2seq-pay-attention-to-self-attention-part-2-中文版-ef2ddf8597a4

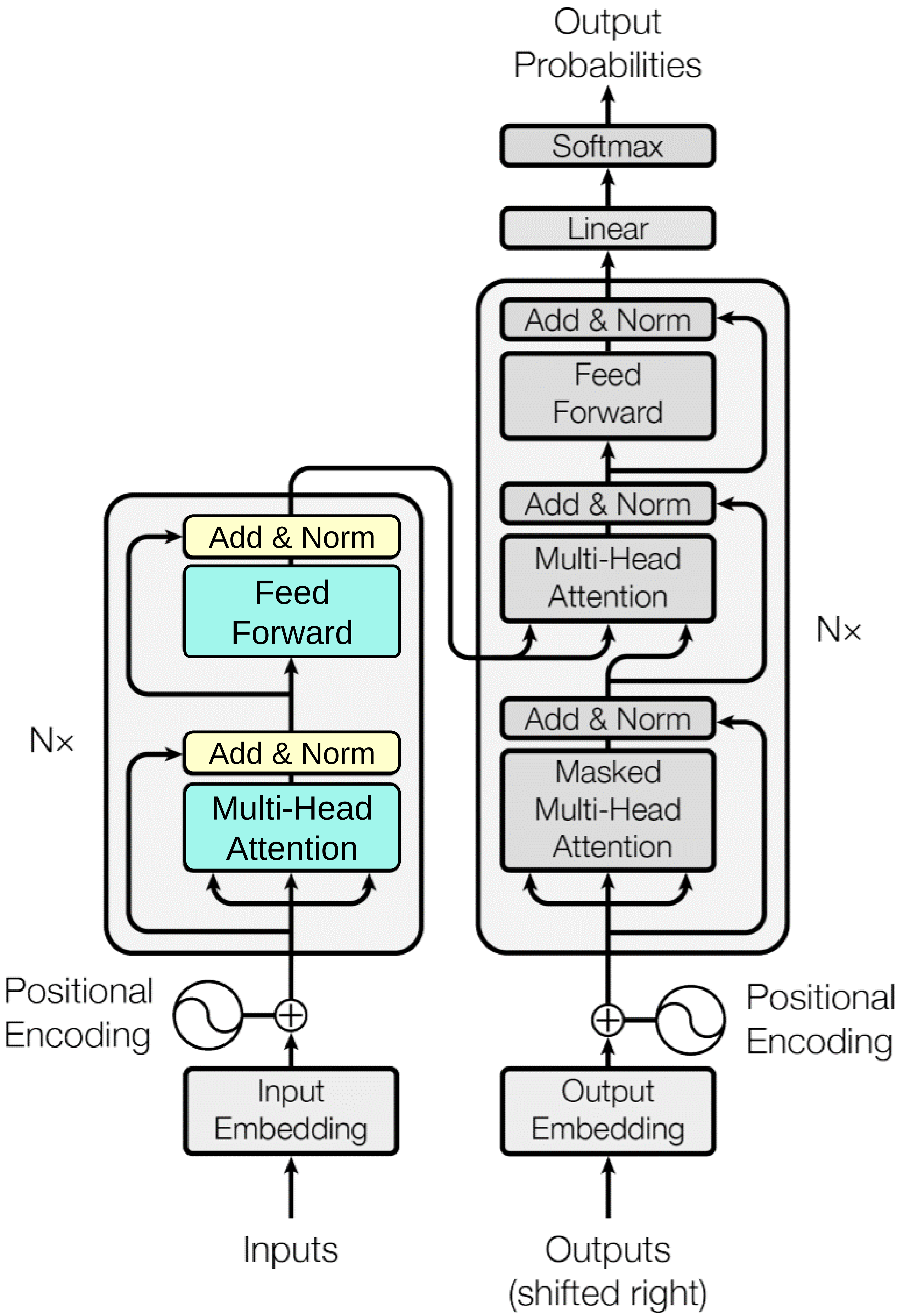
Encoder Input

- Problem: temporal information is missing
- Solution: **positional encoding** allows words at different locations to have different embeddings with fixed dimensions

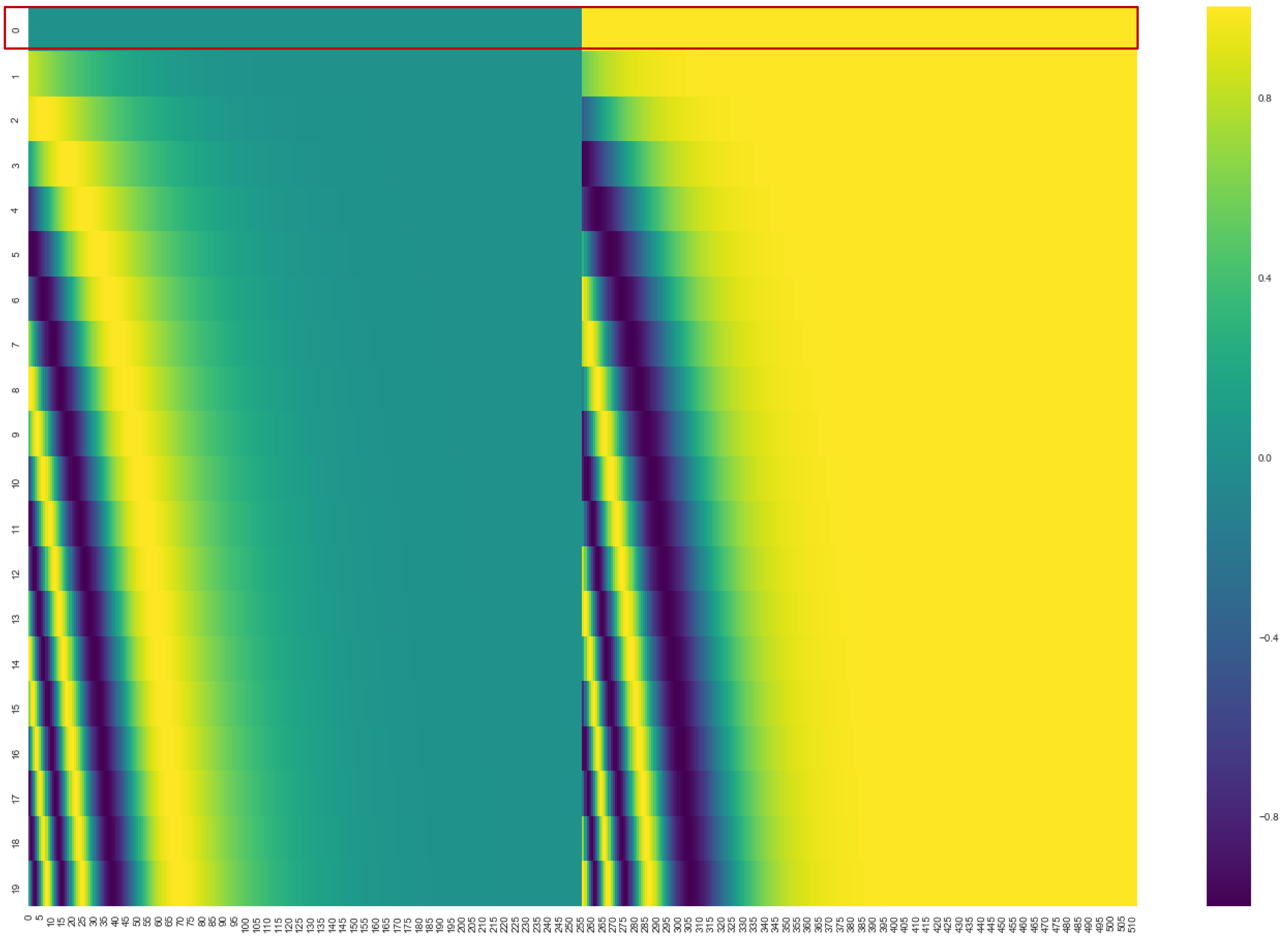
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$$

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





https://medium.com/@bgg/seq2seq-pay-attention-to-self-attention-part-2-中文版-ef2ddf8597a4


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130 135 140 145 150 155 160 165 170 175 180 185 190 195 200 205 210 215 220 225 230 235 240 245 250 255 260 265 270 275 280 285 290 295 300 305 310 315 320 325 330 335 340 345 350 355 360 365 370 375 380 385 390 395 400 405 410 415 420 425 430 435 440 445 450 455 460 465 470 475 480 485 490 495 500 505 510
0.8
0.4
0.0
−0.4
−0.8

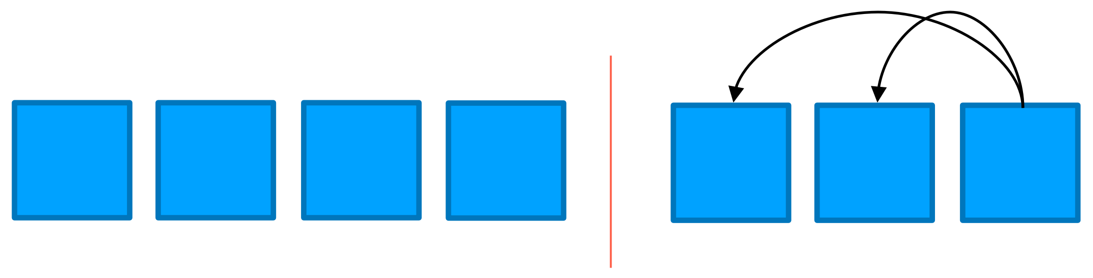
Multi-Head Attention Details

encoder self attention

1. Multi-head Attention
2. Query=Key=Value

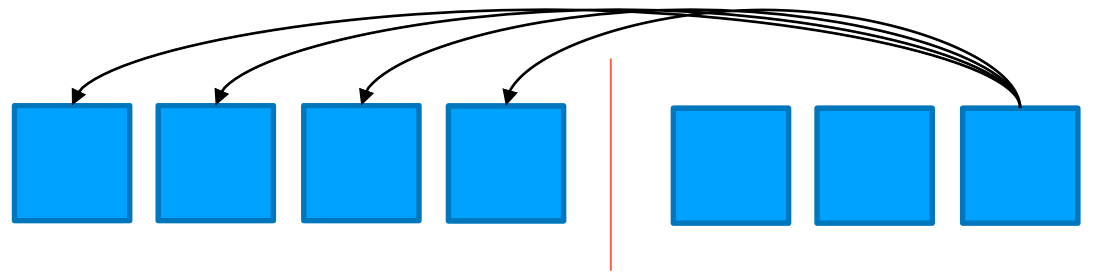
decoder self attention

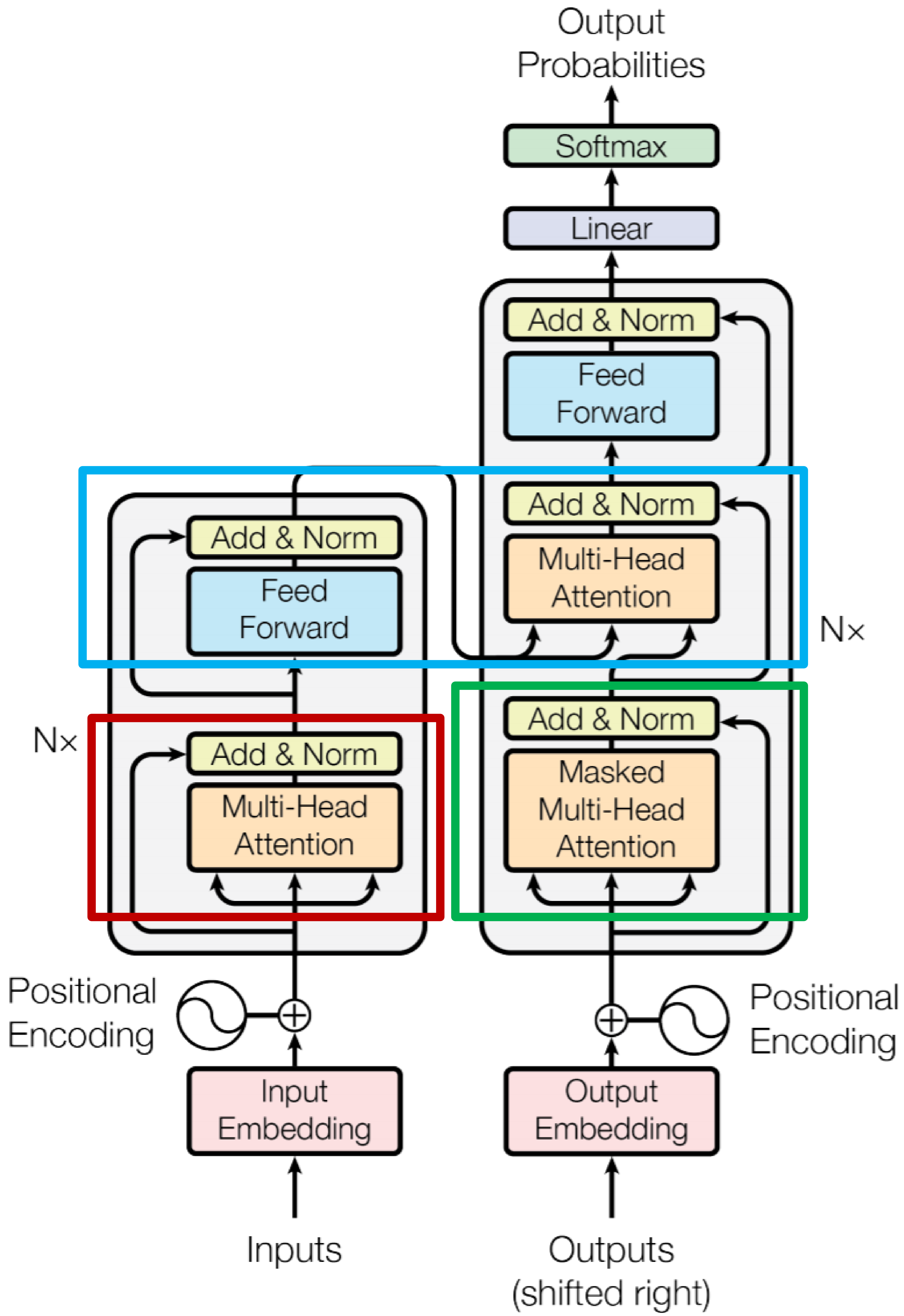
1. Masked Multi-head Attention
2. Query=Key=Value



encoder-decoder attention

1. Multi-head Attention
2. Encoder Self attention=Key=Value
3. Decoder Self attention=Query





https://medium.com/@bgg/seq2seq-pay-attention-to-self-attention-part-2-中文版-ef2ddf8597a4

Training Tips

- Byte-pair encodings (BPE)
- Checkpoint averaging
- ADAM optimizer with learning rate changes
- Dropout during training at every layer just before adding residual
- Label smoothing
- Auto-regressive decoding with beam search and length penalties

MT Experiments

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

Vaswani et al., “Attention Is All You Need”, in *NIPS*, 2017.

Parsing Experiments

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

Vaswani et al., “Attention Is All You Need”, in *NIPS*, 2017.

Concluding Remarks

- **Non-recurrence** model is easy to parallelize
- **Multi-head attention** captures different aspects by interacting between words
- **Positional encoding** captures location information
- Each transformer block can be applied to diverse tasks