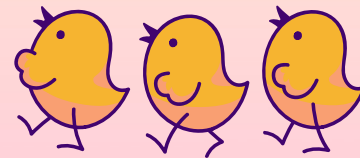


LangCon 2020



Transformer 구현하기

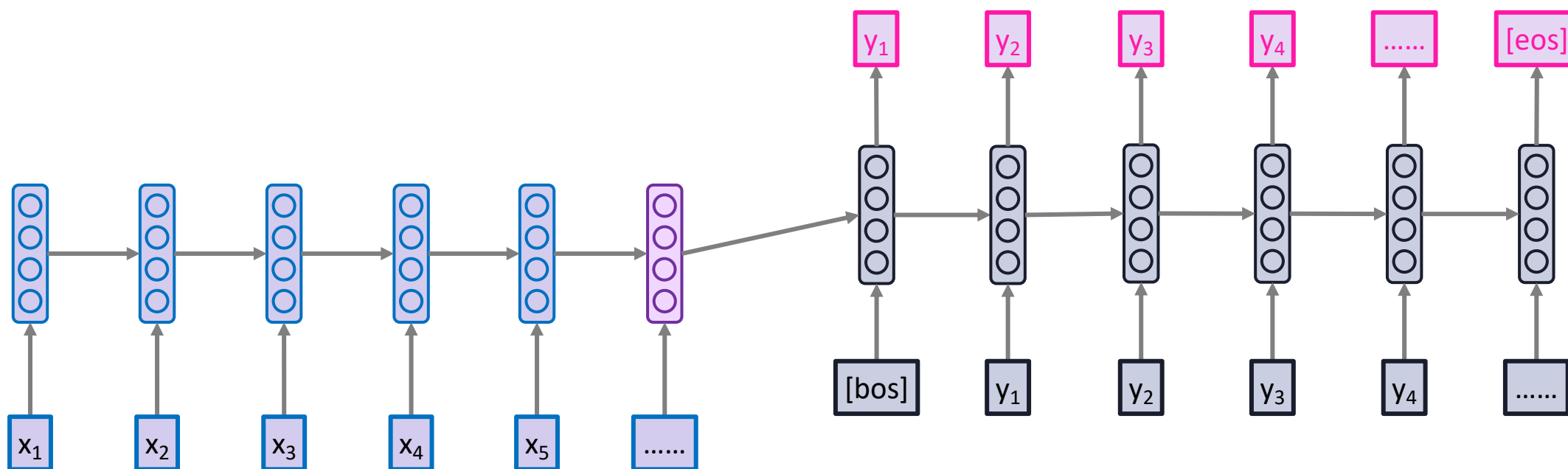
현청천 / cchyun@gmail.com

1. Why Transformer
2. Embedding
 1. Weight Shared Embedding
 2. Positional Encoding
3. Scaled Dot-Product Attention
4. Scaled Dot-Product Attention (masked)
5. Multi-Head Attention
6. Position-wise Feed-Forward Network
7. Reference



Why Transformer

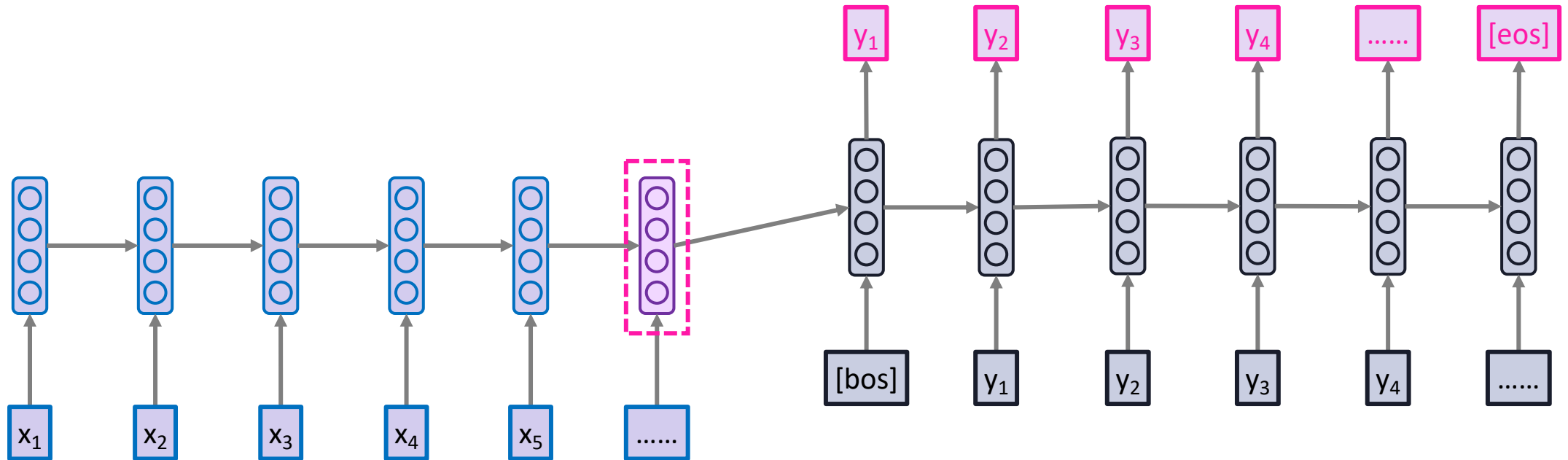
Why Transformer (Seq2Seq)



Why Transformer (Seq2Seq)

Source Sentence Encoding

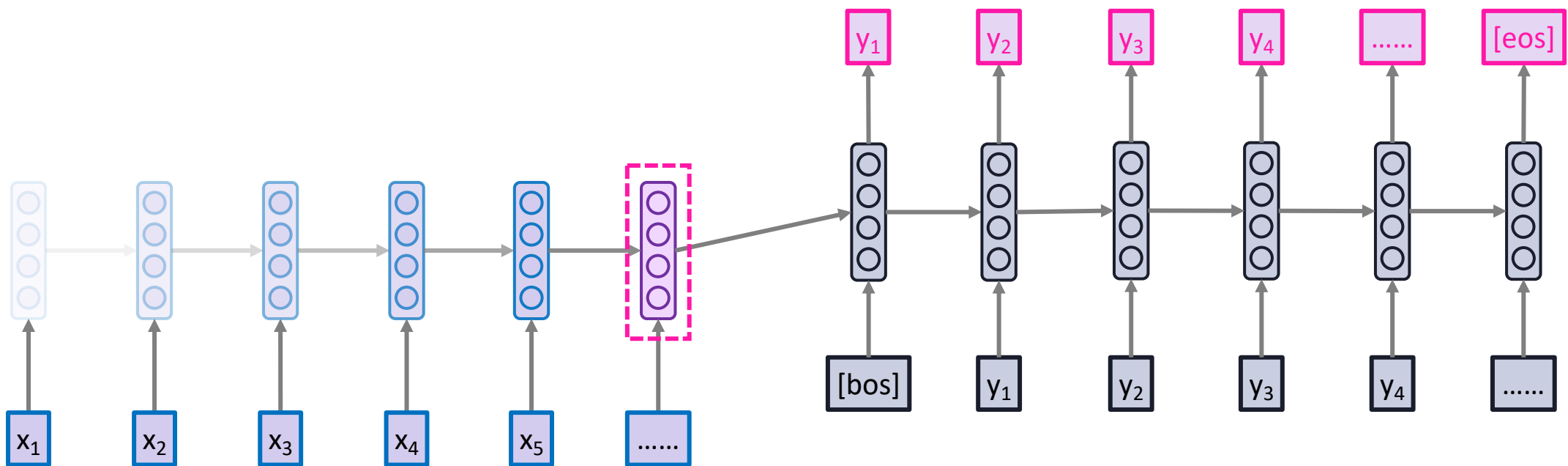
Source Sentence의 모든 정보를 저장 해야 함 (Information bottleneck)



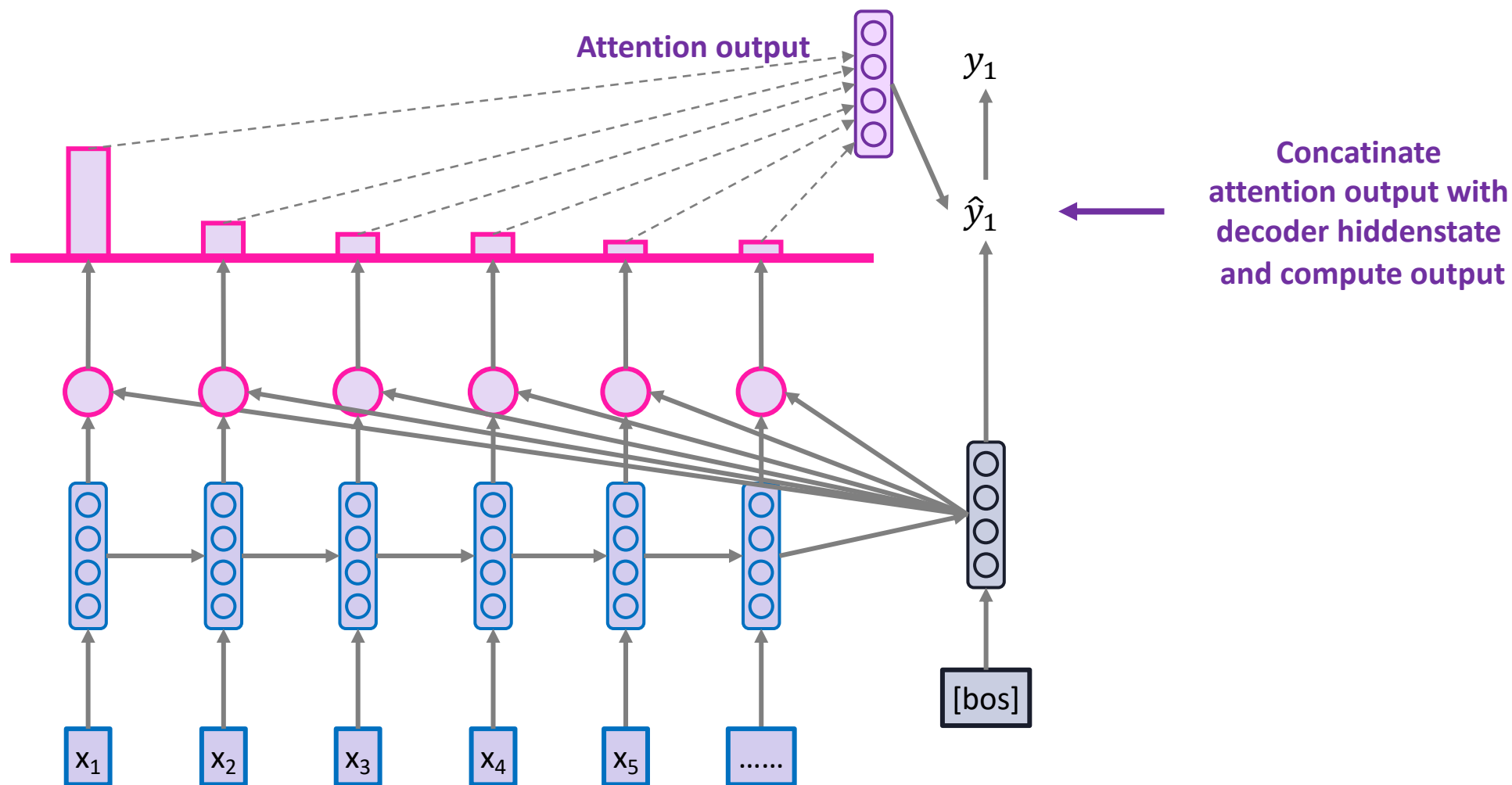
Why Transformer (Seq2Seq)

Source Sentence Encoding

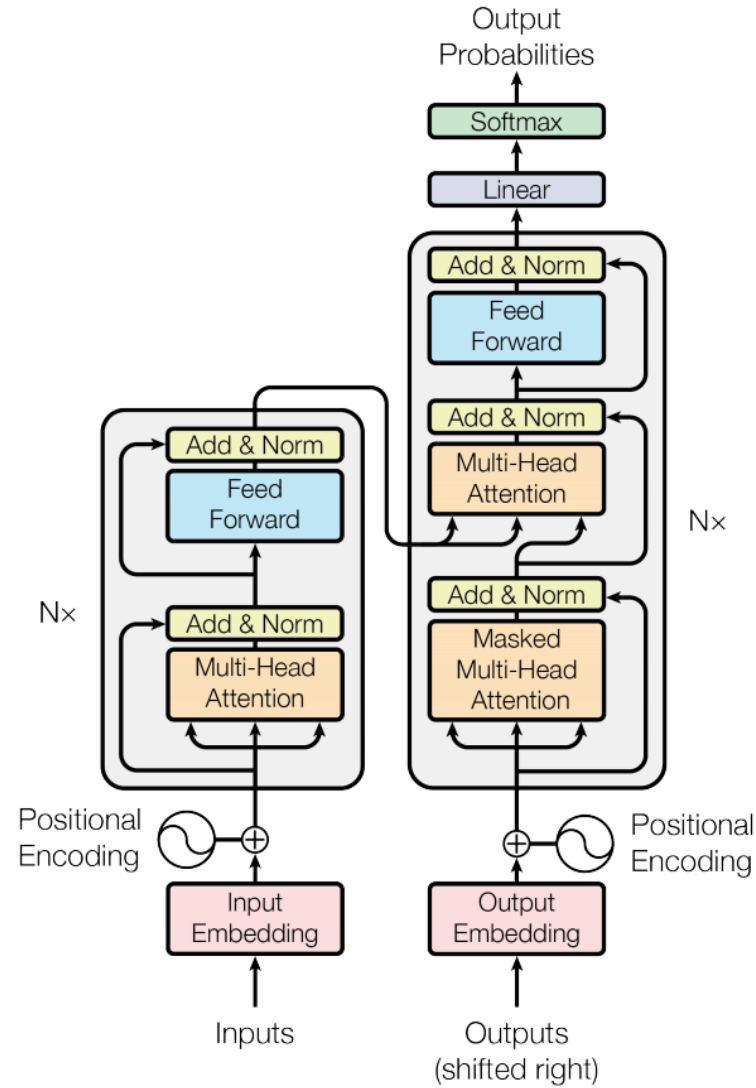
과거 step의 정보는 최신 step의 보다 덜 사용 됨 (Vanishing Gradient)



Why Transformer (Attention)



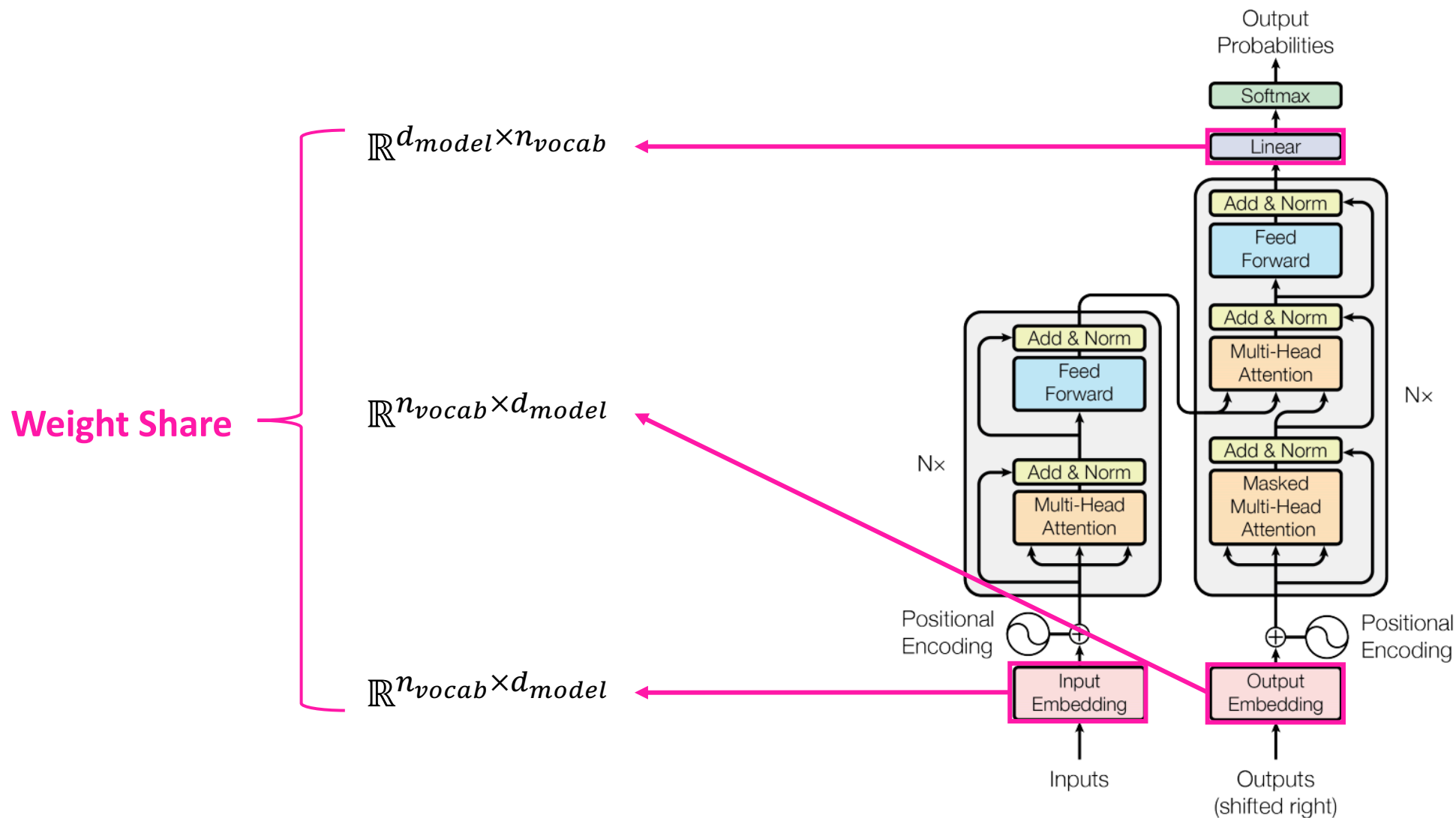
Why Transformer (Transformer)





Embedding

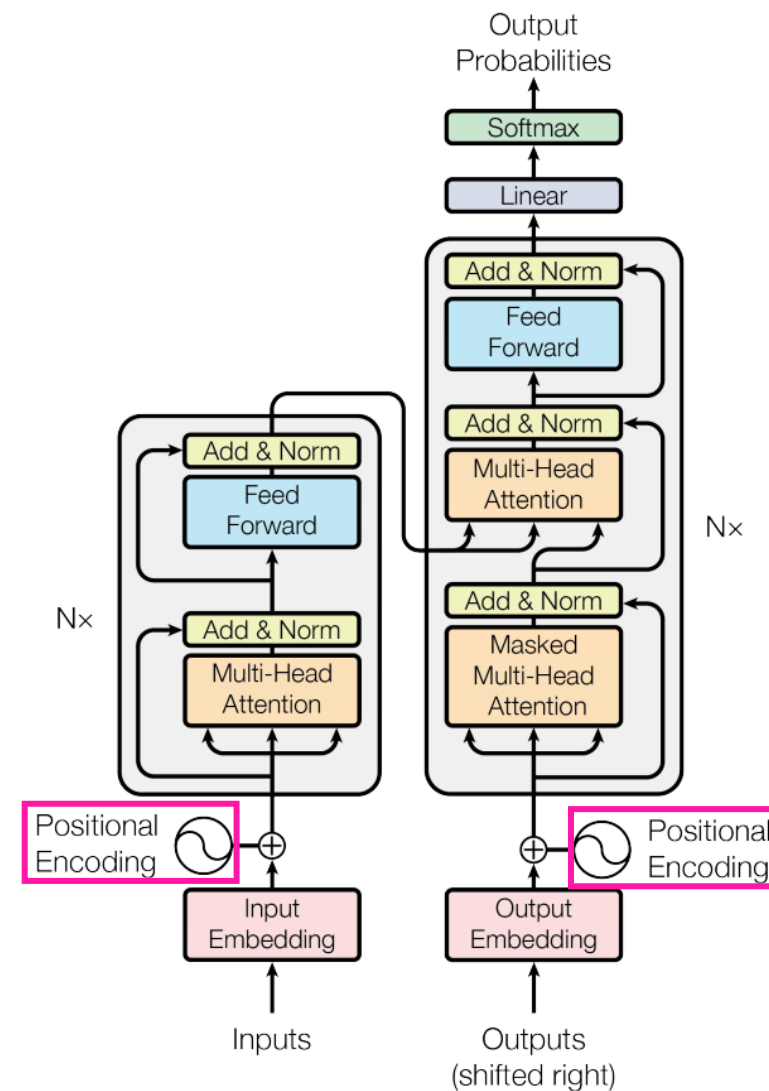
Embedding - Weight Shared



Embedding - Positional Encoding

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

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




Embedding - Positional Encoding

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

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

$pos/10000^{2i/d_{model}}$

where $seq = 16, d_{model} = 8$



0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	1.0	1.0	0.1	0.1	0.01	0.01	0.001	0.001
2	2.0	2.0	0.2	0.2	0.02	0.02	0.002	0.002
3	3.0	3.0	0.3	0.3	0.03	0.03	0.003	0.003
4	4.0	4.0	0.4	0.4	0.04	0.04	0.004	0.004
5	5.0	5.0	0.5	0.5	0.05	0.05	0.005	0.005

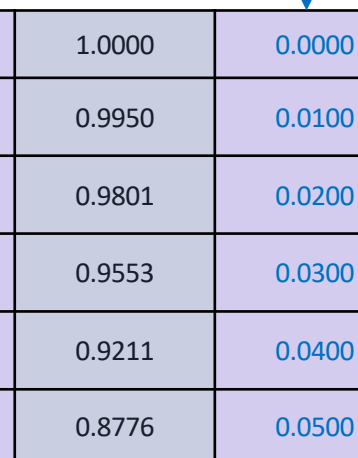
Embedding - Positional Encoding

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

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

$$\sin(pos/10000^{2i/d_{model}})$$

where $seq = 16, d_{model} = 8$



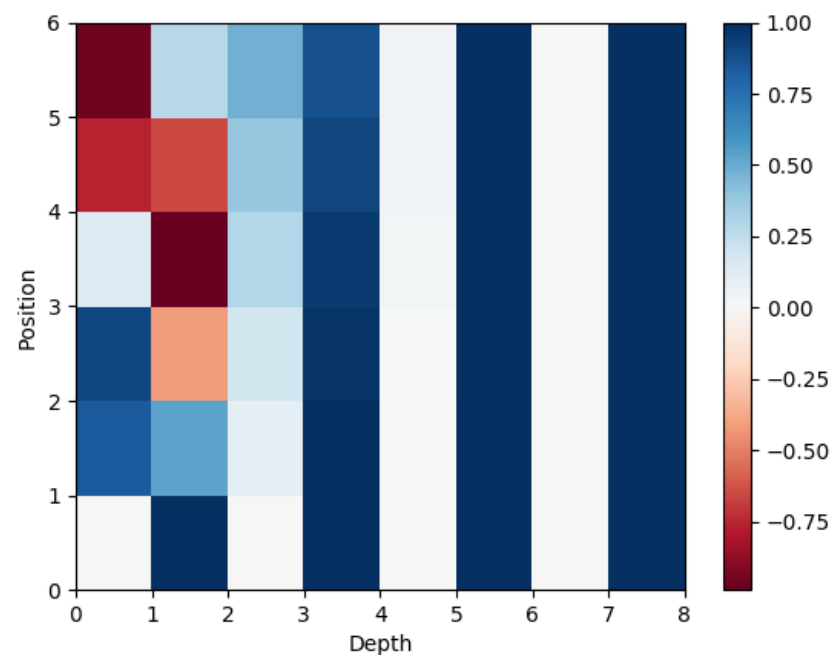
0	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000
1	0.8415	0.5403	0.0998	0.9950	0.0100	1.0000	0.0010	1.0000
2	0.9093	-0.4161	0.1987	0.9801	0.0200	0.9998	0.0020	1.0000
3	0.1411	-0.9900	0.2955	0.9553	0.0300	0.9996	0.0030	1.0000
4	-0.756	-0.6536	0.3894	0.9211	0.0400	0.9992	0.0040	1.0000
5	-0.959	0.2837	0.4794	0.8776	0.0500	0.9988	0.0050	1.0000

$$\cos(pos/10000^{2i/d_{model}})$$

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

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

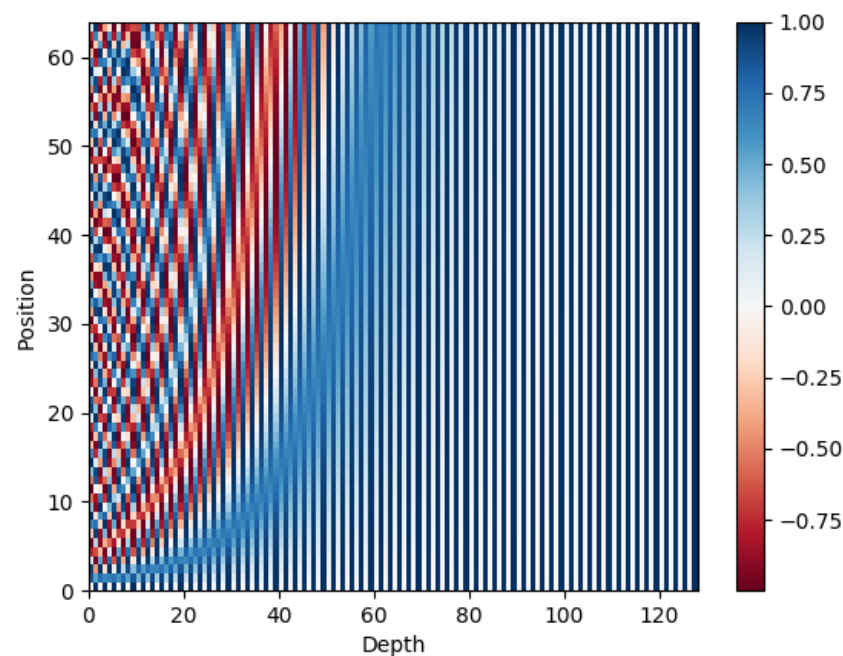
where $seq = 16, d_{model} = 8$



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

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

where $seq = 64, d_{model} = 128$

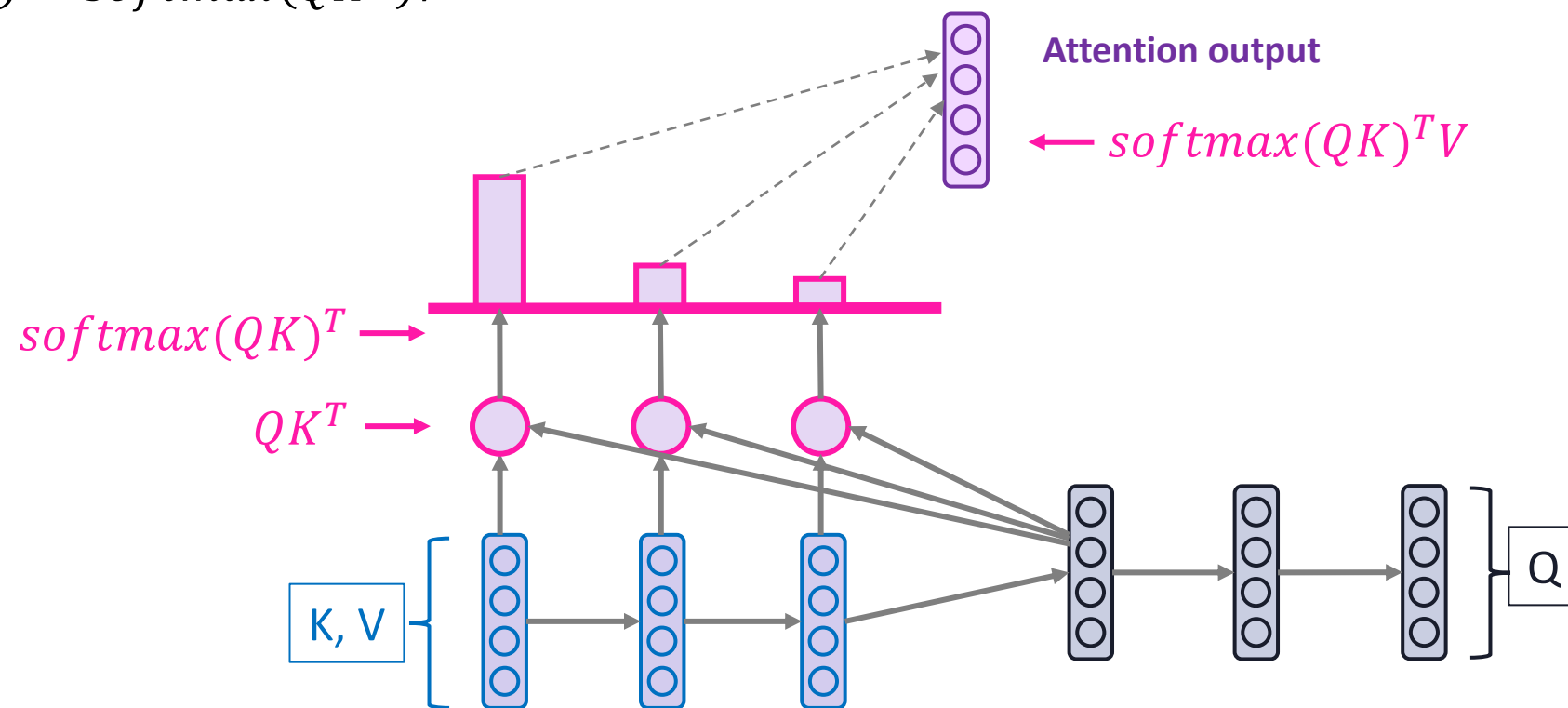




Scaled Dot-Product Attention

Dot-product Attention

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T)V$$



Dot-product Attention

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T)V$$

- $Q \in \mathbb{R}^{|Q| \times d_k}$
- $K \in \mathbb{R}^{|K| \times d_k}$
- $V \in \mathbb{R}^{|K| \times d_v}$
- $QK^T = [|Q| \times d_k] \times [d_k \times |K|]$

Problem of Dot-product Attention

- d_k 가 커지면 QK^T 의 결과값의 편차가 커짐
- $\text{softmax}(QK^T)$ 의 결과 값이 편차가 커짐
- Gradient가 작아짐
- 학습이 잘 안됨



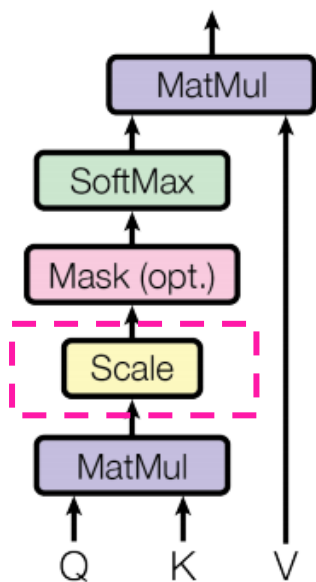
Scaled Dot-product Attention

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

- QK^T 의 결과를 $\sqrt{d_k}$ 로 나눔
- 값의 편차가 줄어듦

Scaled Dot-Product Attention

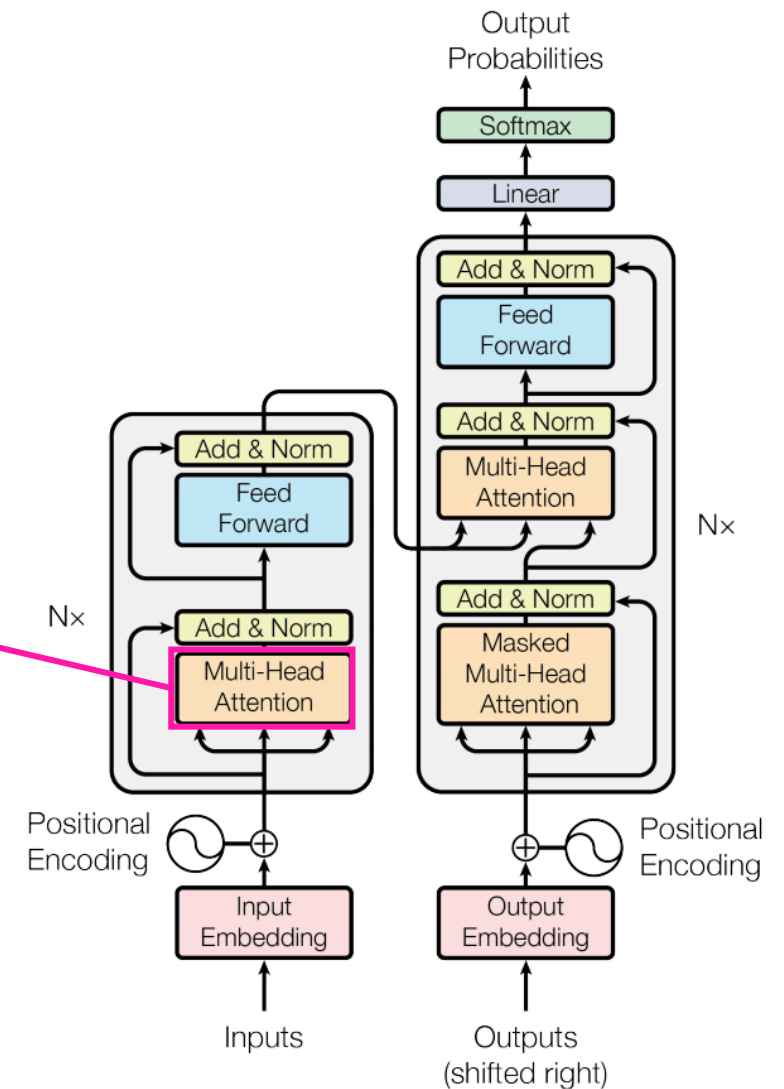
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



Scaled Dot-Product Attention

- Q: encoder hidden
- K: encoder hidden
- V: encoder hidden

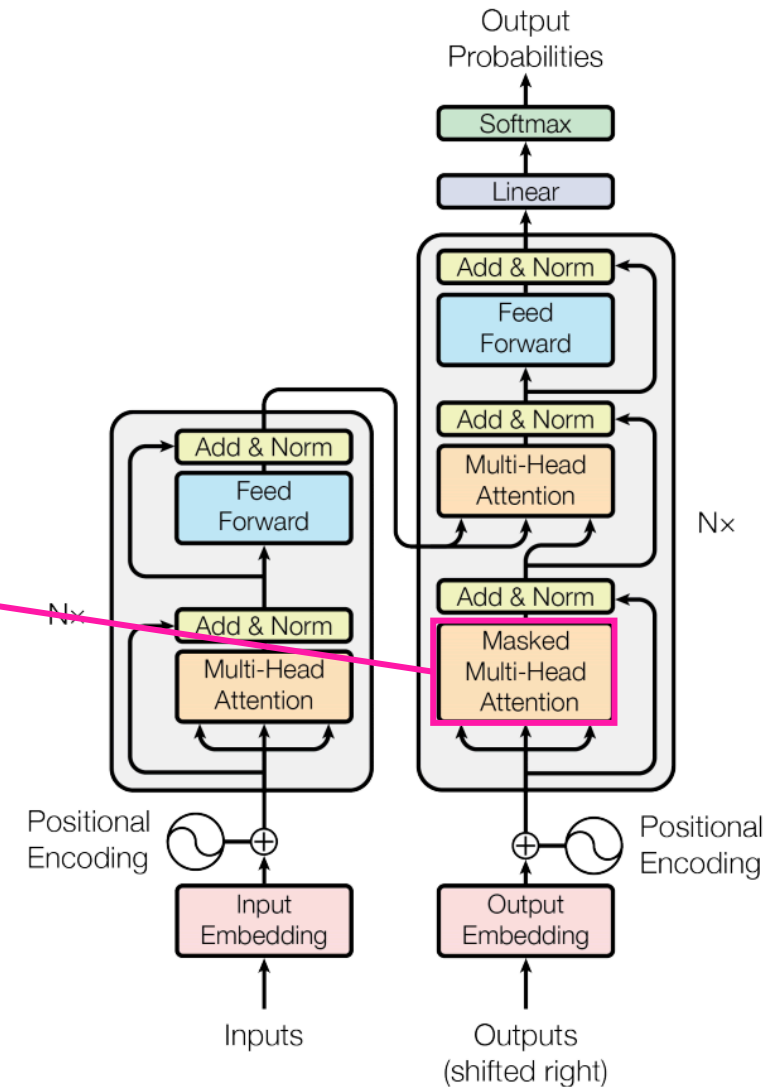
Encoder Self Attention



Scaled Dot-Product Attention

- Q: decoder hidden
- K: decoder hidden
- V: decoder hidden

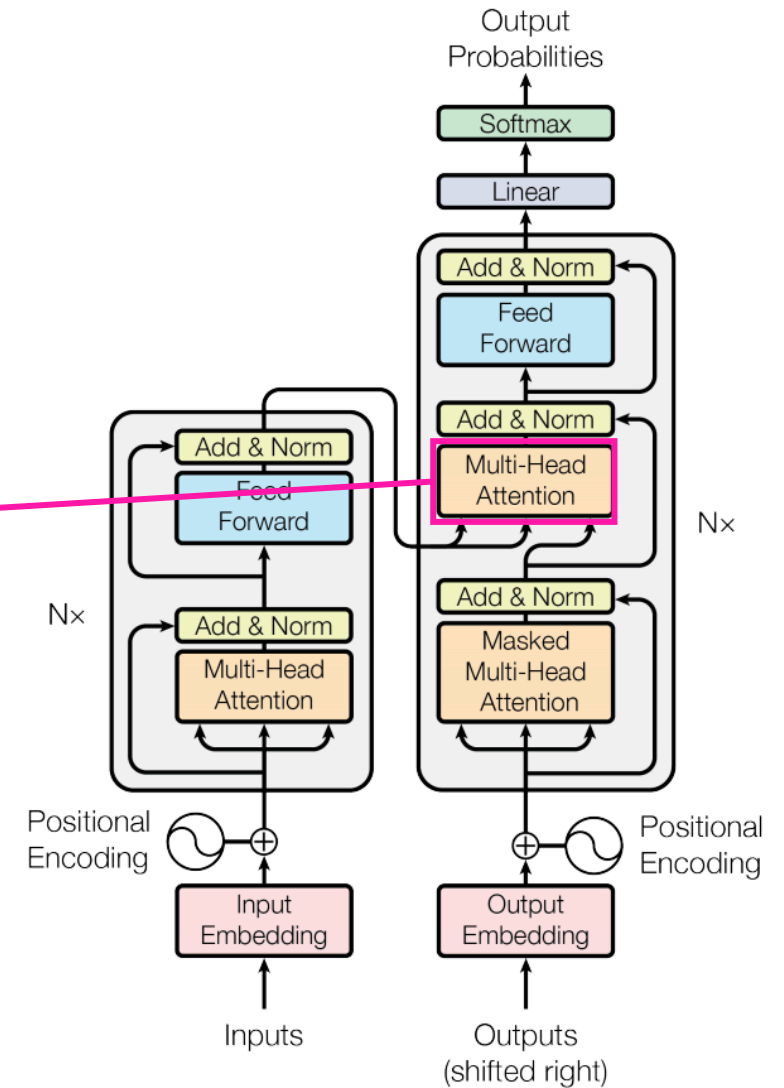
Decoder Self Attention (masked)



Scaled Dot-Product Attention

- Q: decoder hidden
- K: encoder hidden
- V: encoder hidden

Encoder-Decoder Attention



Scaled Dot-Product Attention

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

K
V



Education	0.1	0.2	0.3	0.4
is	0.2	0.3	0.4	0.5
most	0.3	0.4	0.5	0.6
powerful	0.4	0.3	0.2	0.1
weapon	0.5	0.4	0.3	0.2
[pad]	0.1	0.1	0.1	0.1

$[seq_K \times d_{model}]$

- seq_K : 6
- d_{model} : 4

Q



교육은	0.1	0.2	0.3	0.4
가장	0.2	0.3	0.4	0.5
중요한	0.3	0.4	0.5	0.6
무기이다	0.4	0.3	0.2	0.1
[pad]	0.1	0.1	0.1	0.1

$[seq_Q \times d_{model}]$

- seq_Q : 5
- d_{model} : 4

Scaled Dot-Product Attention

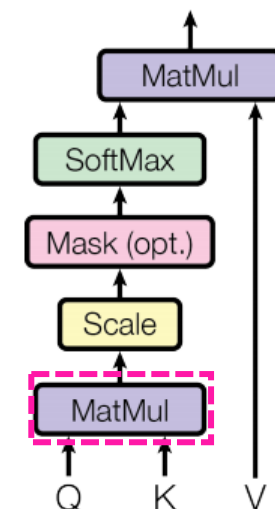
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

QK^T



	Education	is	most	powerful	weapon	[pad]
교육은	0.3	0.4	0.5	0.2	0.3	0.1
가장	0.4	0.54	0.68	0.3	0.44	0.14
중요한	0.5	0.68	0.86	0.4	0.58	0.18
무기이다	0.2	0.3	0.4	0.3	0.4	0.1
[pad]	0.1	0.14	0.18	0.1	0.14	0.04

$$[seq_Q \times d_{model}] \times [d_{model} \times seq_K] = [seq_Q \times seq_K]$$



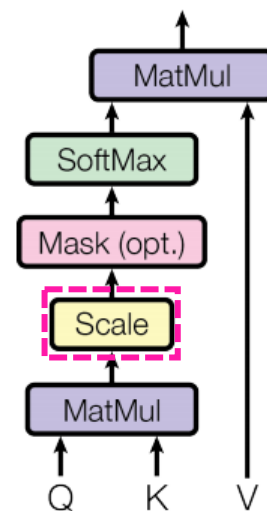
Scaled Dot-Product Attention

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

$$\frac{QK^T}{\sqrt{d_k}}$$



	Education	is	most	powerful	weapon	[pad]
교육은	0.15	0.2	0.25	0.1	0.15	0.05
가장	0.2	0.27	0.34	0.15	0.22	0.07
중요한	0.25	0.34	0.43	0.2	0.29	0.09
무기이다	0.1	0.15	0.2	0.15	0.2	0.05
[pad]	0.05	0.07	0.09	0.05	0.07	0.02



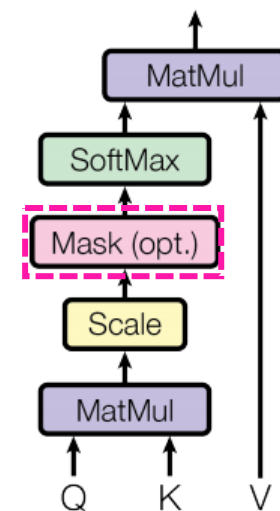
Scaled Dot-Product Attention

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

$$\text{mask}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$



	Education	is	most	powerful	weapon	[pad]
교육은	0.15	0.2	0.25	0.1	0.15	-inf
가장	0.2	0.27	0.34	0.15	0.22	-inf
중요한	0.25	0.34	0.43	0.2	0.29	-inf
무기이다	0.1	0.15	0.2	0.15	0.2	-inf
[pad]	0.05	0.07	0.09	0.05	0.07	-inf



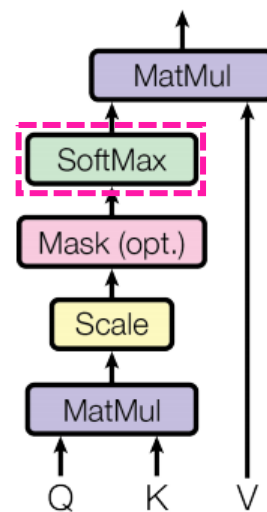
Scaled Dot-Product Attention

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

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$



	Education	is	most	powerful	weapon	[pad]
교육은	0.19	0.20	0.21	0.18	0.10	0
가장	0.19	0.20	0.22	0.18	0.19	0
중요한	0.18	0.20	0.22	0.18	0.19	0
무기이다	0.18	0.19	0.20	0.19	0.20	0
[pad]	0.19	0.20	0.20	0.19	0.20	0



Scaled Dot-Product Attention

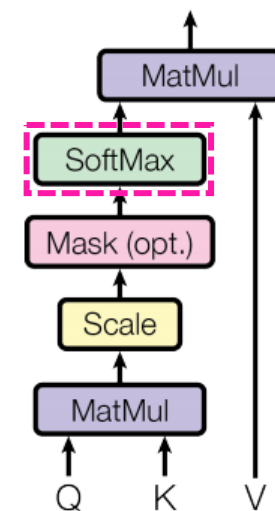
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



교육은	0.29	0.32	0.34	0.36
가장	0.29	0.32	0.34	0.37
중요한	0.29	0.32	0.34	0.37
무기이다	0.30	0.32	0.34	0.36
[pad]	0.30	0.32	0.34	0.36

← Weighted sum of V



$$[seq_Q \times seq_K] \times [seq_K \times d_{model}] = [seq_Q \times d_{model}]$$



Scaled Dot-Product Attention (masked)

Scaled Dot-Product Attention (masked)

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

K
V



Education	0.1	0.2	0.3	0.4
is	0.2	0.3	0.4	0.5
most	0.3	0.4	0.5	0.6
powerful	0.4	0.3	0.2	0.1
weapon	0.5	0.4	0.3	0.2
[pad]	0.1	0.1	0.1	0.1

$[seq_K \times d_{model}]$

- seq_K : 6
- d_{model} : 4

Q



교육은	0.1	0.2	0.3	0.4
가장	0.2	0.3	0.4	0.5
중요한	0.3	0.4	0.5	0.6
무기이다	0.4	0.3	0.2	0.1
[pad]	0.1	0.1	0.1	0.1

$[seq_Q \times d_{model}]$

- seq_Q : 5
- d_{model} : 4

Scaled Dot-Product Attention (masked)

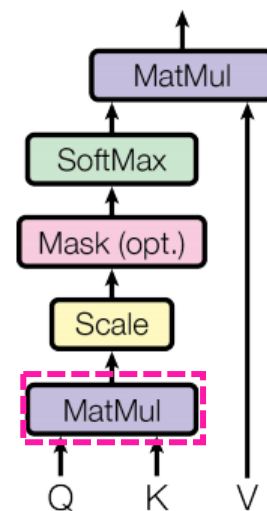
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

QK^T



	Education	is	most	powerful	weapon	[pad]
교육은	0.3	0.4	0.5	0.2	0.3	0.1
가장	0.4	0.54	0.68	0.3	0.44	0.14
중요한	0.5	0.68	0.86	0.4	0.58	0.18
무기이다	0.2	0.3	0.4	0.3	0.4	0.1
[pad]	0.1	0.14	0.18	0.1	0.14	0.04

$$[seq_Q \times d_{model}] \times [d_{model} \times seq_K] = [seq_Q \times seq_K]$$



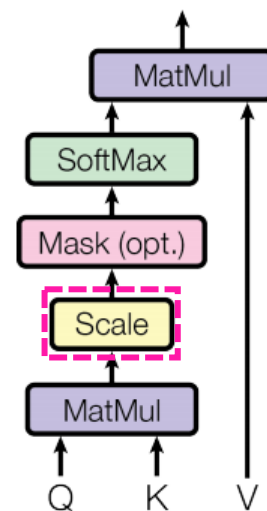
Scaled Dot-Product Attention (masked)

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

$$\frac{QK^T}{\sqrt{d_k}}$$



	Education	is	most	powerful	weapon	[pad]
교육은	0.15	0.2	0.25	0.1	0.15	0.05
가장	0.2	0.27	0.34	0.15	0.22	0.07
중요한	0.25	0.34	0.43	0.2	0.29	0.09
무기이다	0.1	0.15	0.2	0.15	0.2	0.05
[pad]	0.05	0.07	0.09	0.05	0.07	0.02



Scaled Dot-Product Attention (masked)

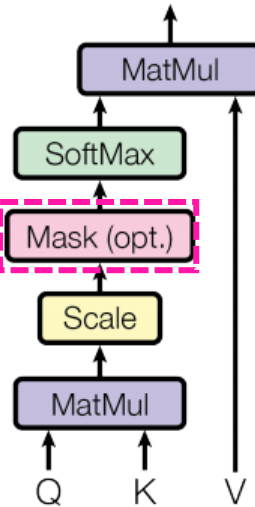
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\text{mask}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$



	Education	is	most	powerful	weapon	[pad]
교육은	0.15	-inf	-inf	-inf	-inf	-inf
가장	0.2	0.27	-inf	-inf	-inf	-inf
중요한	0.25	0.34	0.43	-inf	-inf	-inf
무기이다	0.1	0.15	0.2	0.15	-inf	-inf
[pad]	0.05	0.07	0.09	0.05	0.07	-inf

← Can't see next value



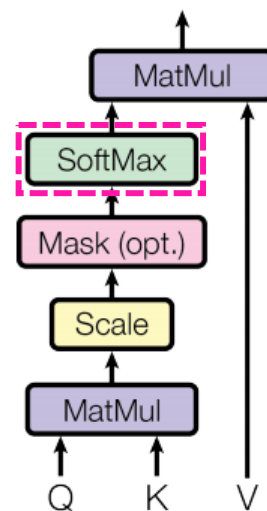
Scaled Dot-Product Attention (masked)

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

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$



	Education	is	most	powerful	weapon	[pad]
교육은	1.00	0	0	0	0	0
가장	0.48	0.51	0	0	0	0
중요한	0.30	0.33	0.36	0	0	0
무기이다	0.23	0.24	0.26	0.24	0	0
[pad]	0.19	0.20	0.20	0.19	0.20	0



Scaled Dot-Product Attention (masked)

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

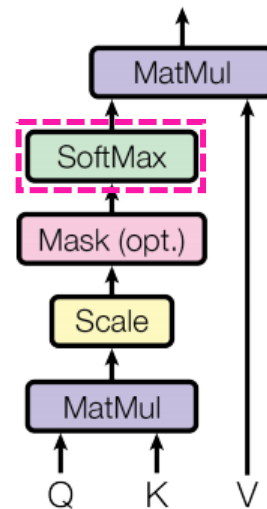
$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



교육은	0.1	0.2	0.3	0.4
가장	0.15	0.25	0.35	0.45
중요한	0.20	0.30	0.40	0.50
무기이다	0.25	0.30	0.35	0.40
[pad]	0.30	0.32	0.34	0.36

← Weighted sum of V

$$[seq_Q \times seq_K] \times [seq_K \times d_{model}] = [seq_Q \times d_{model}]$$

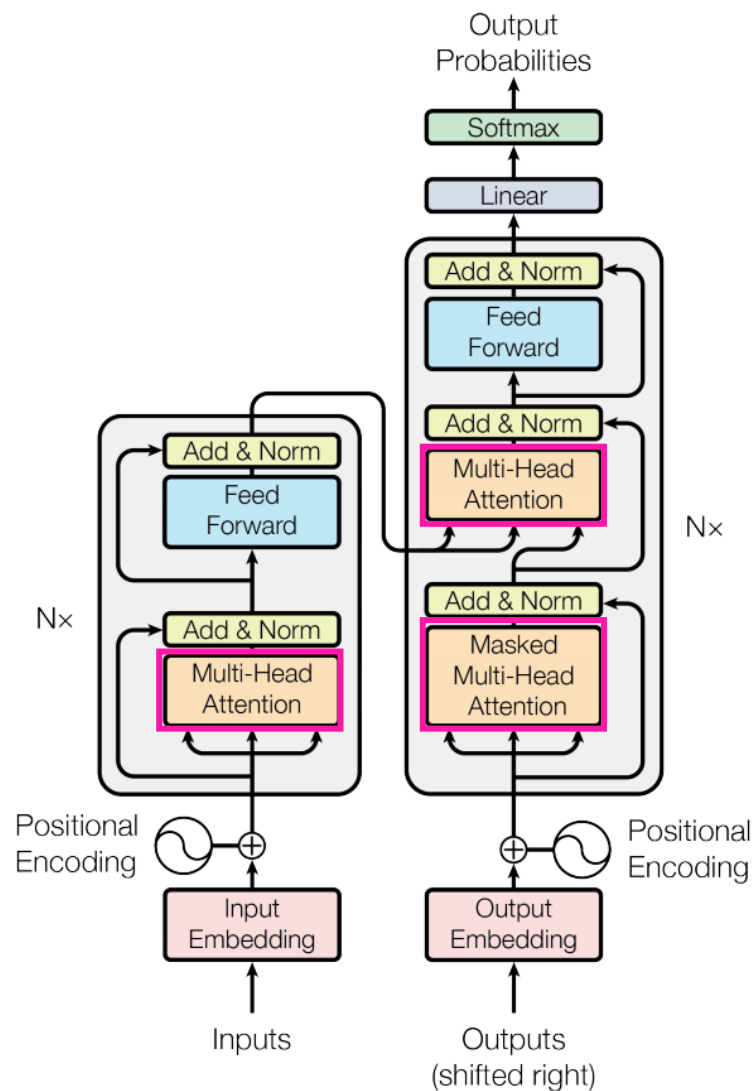
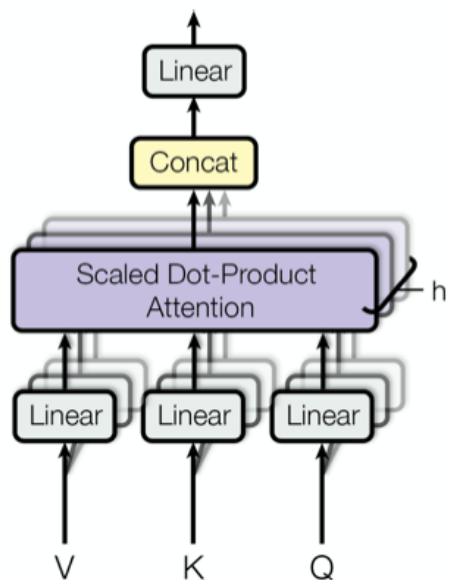




Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = \text{concatenate}(\text{head}_1, \dots, \text{head}_h)W^O$$

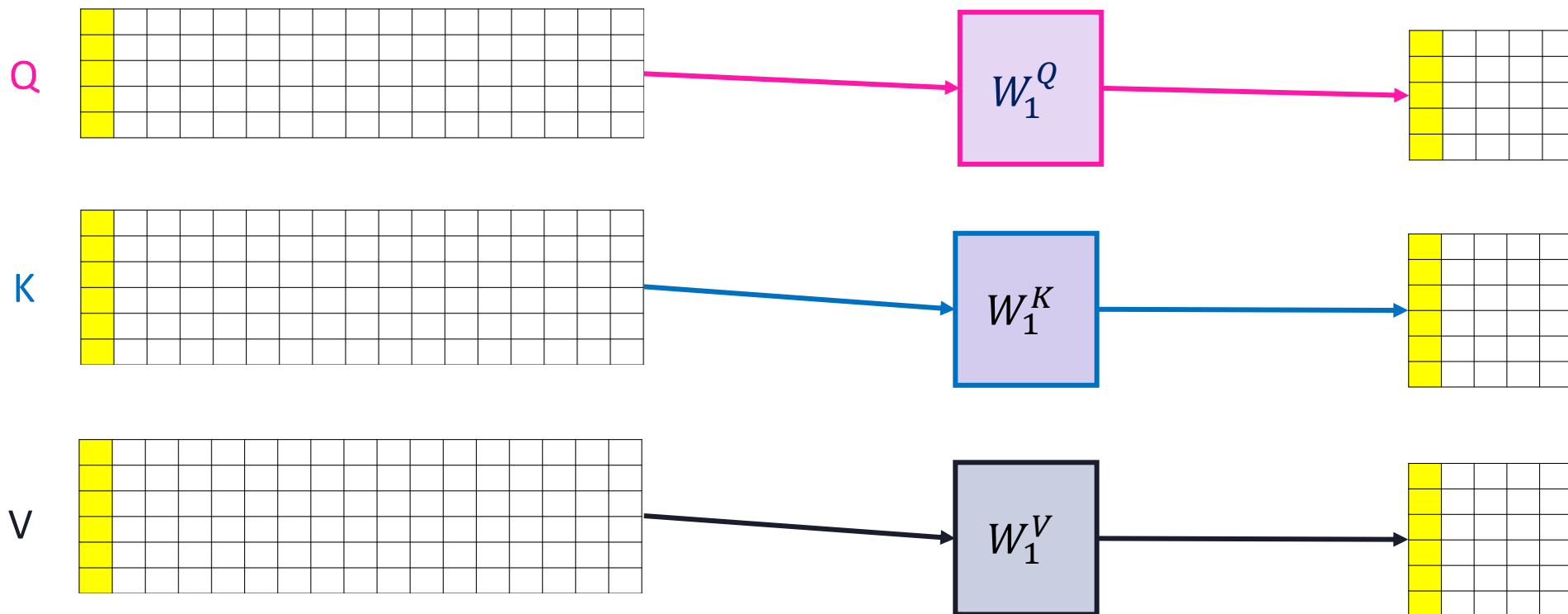
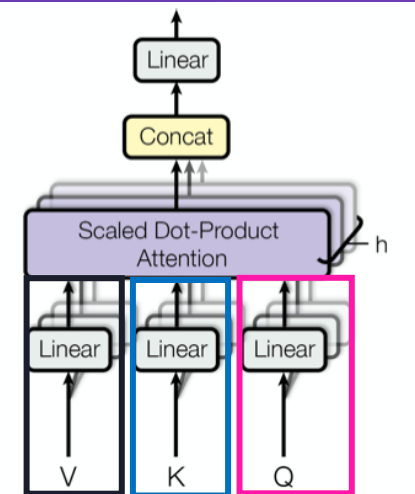
$$\text{where } \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = \text{concatenate}(\text{head}_1, \dots, \text{head}_h) W^O$$

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



$$QW_1^Q$$

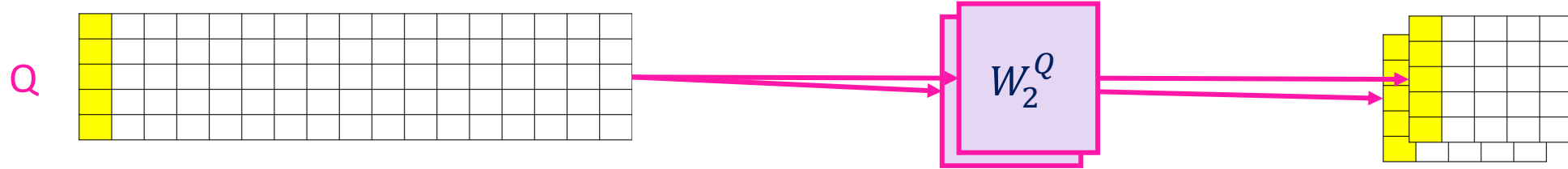
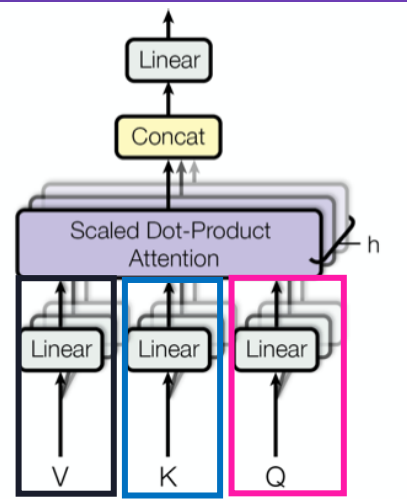
$$KW_1^K$$

$$VW_1^V$$

Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = \text{concatenate}(\text{head}_1, \dots, \text{head}_h) W^O$$

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



$$QW_2^Q$$



$$KW_2^K$$

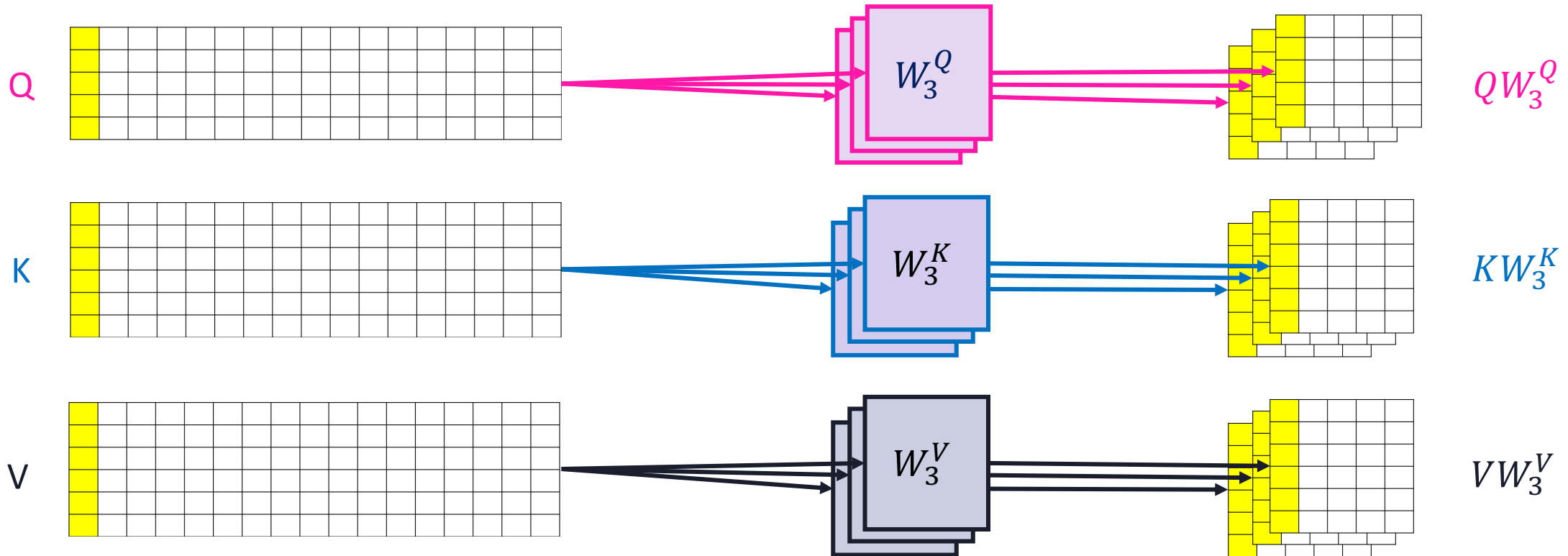
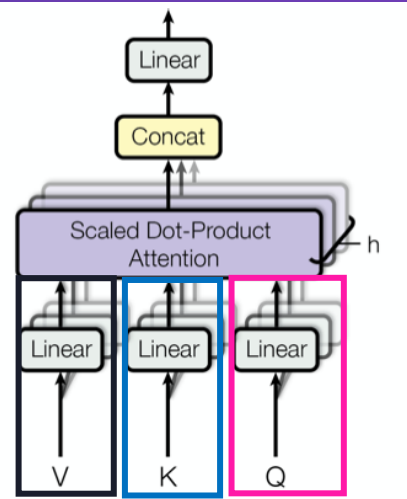


$$VW_2^V$$

Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = \text{concatenate}(\text{head}_1, \dots, \text{head}_h) W^O$$

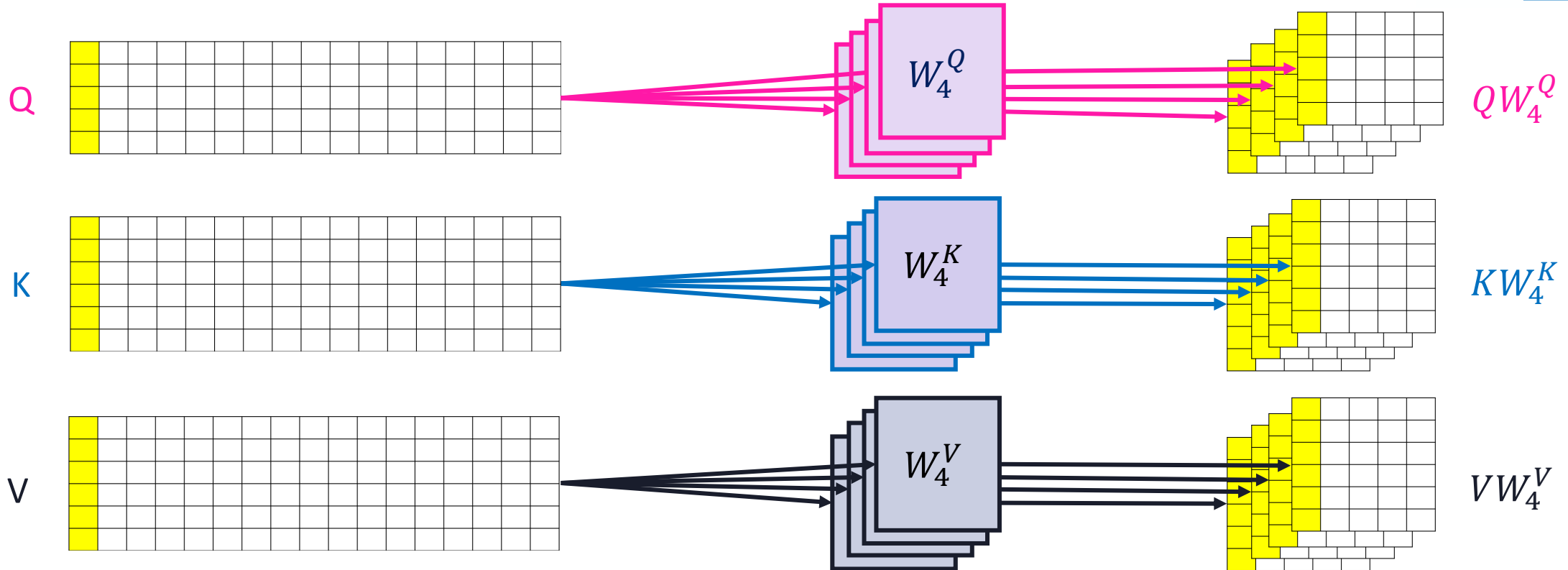
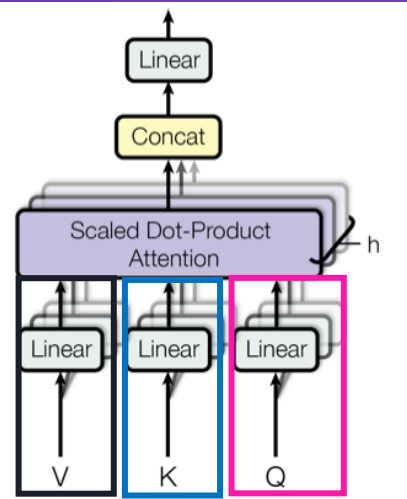
$$\text{where } \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = \text{concatenate}(\text{head}_1, \dots, \text{head}_h) W^O$$

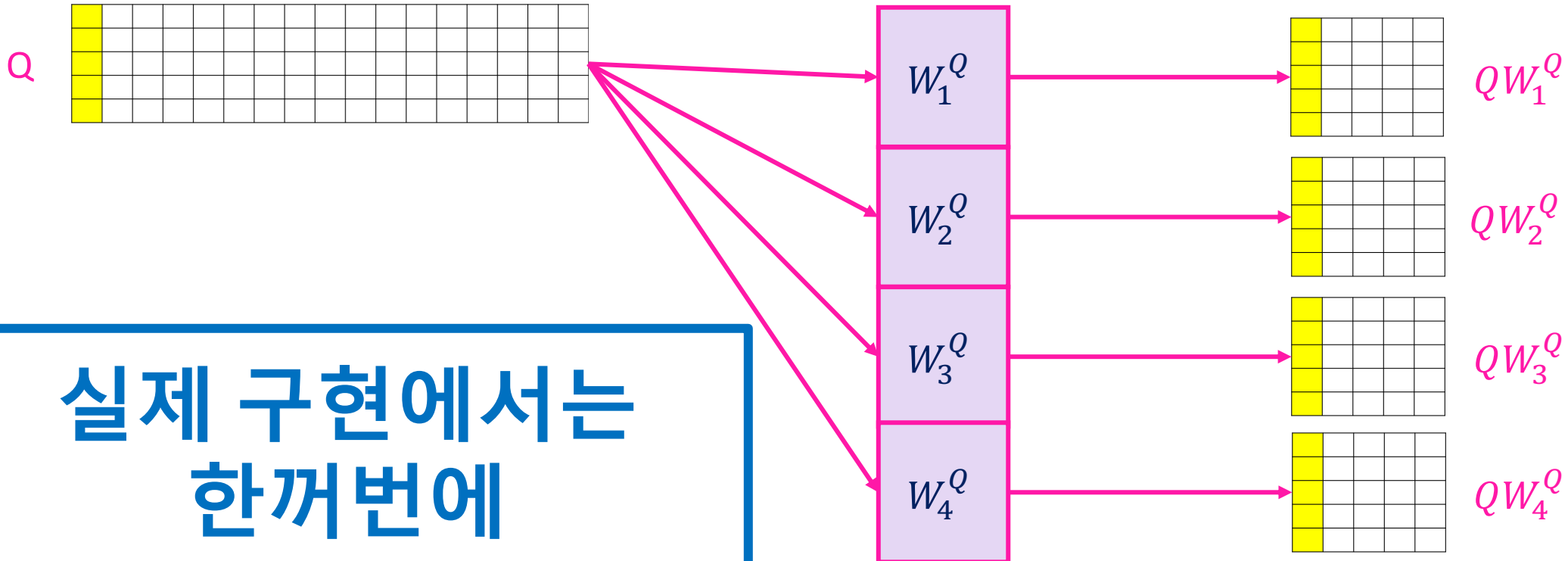
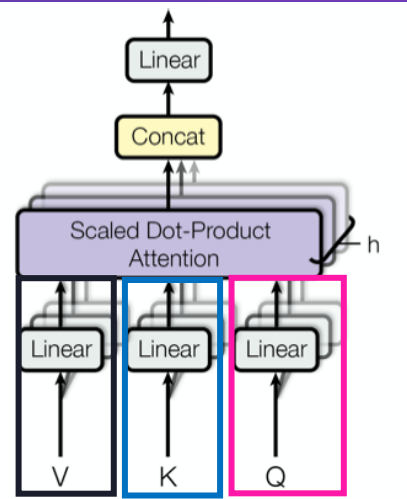
$$\text{where } \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = \text{concatenate}(\text{head}_1, \dots, \text{head}_h) W^O$$

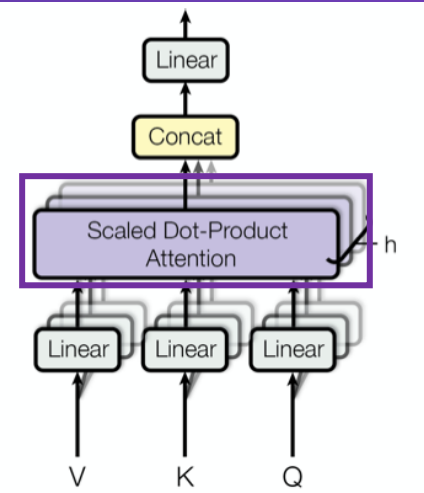
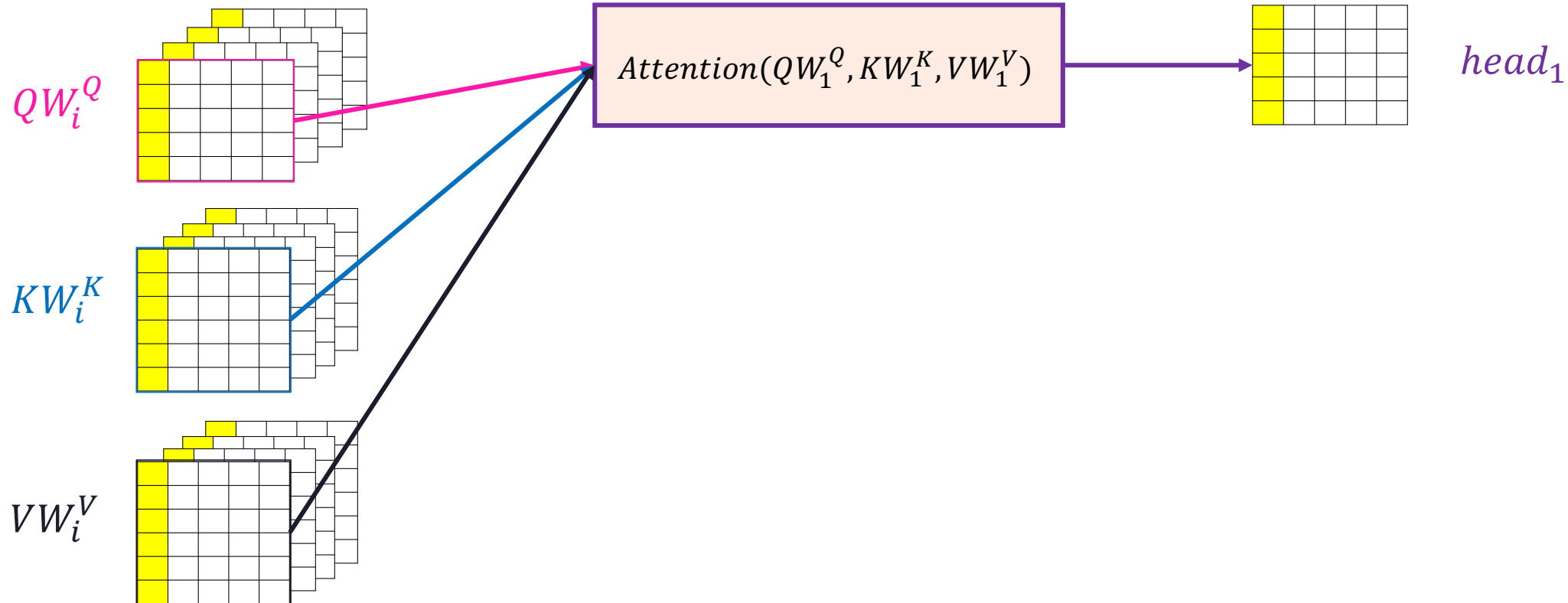
$$\text{where } \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = \text{concatenate}(\text{head}_1, \dots, \text{head}_h) W^O$$

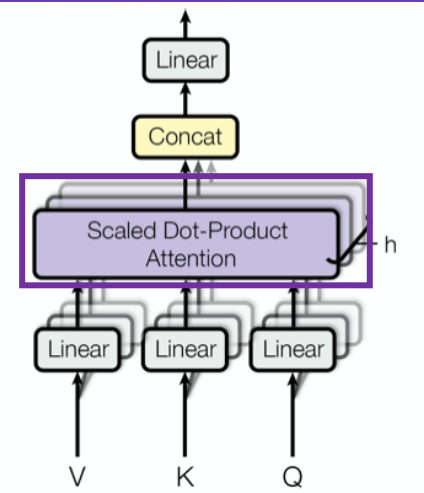
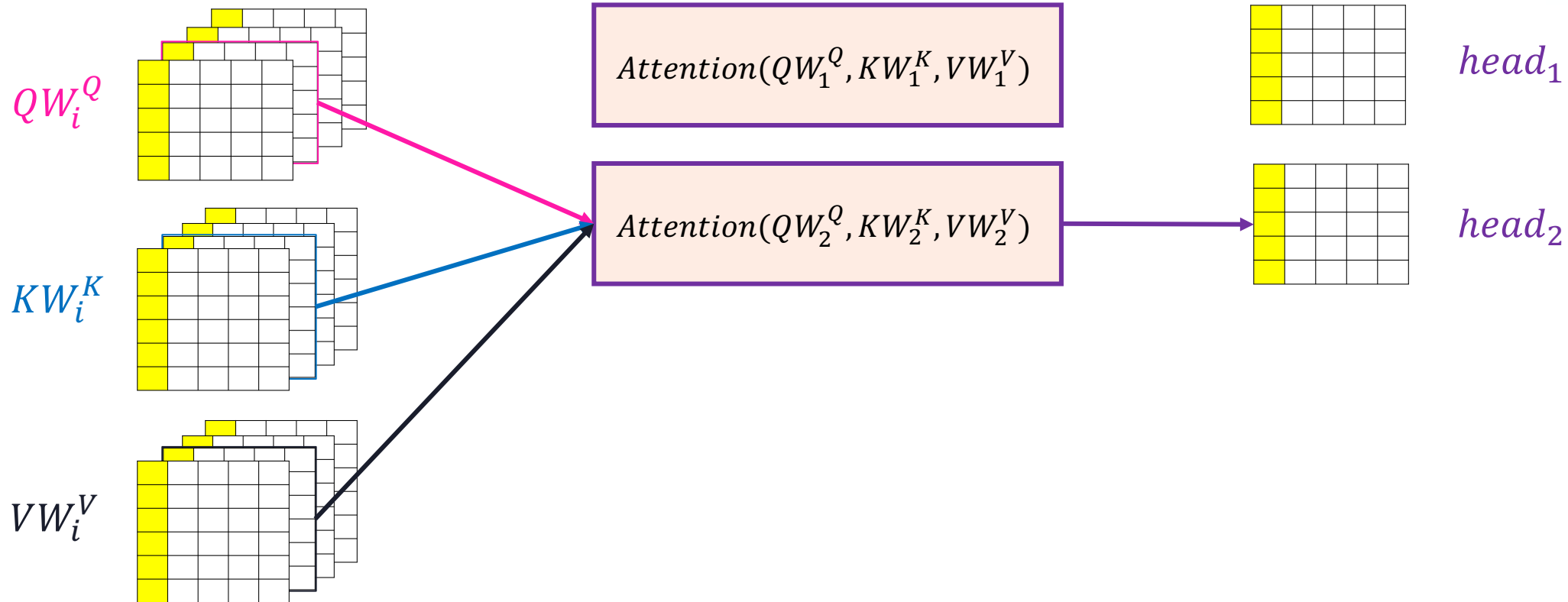
$$\text{where } \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = \text{concatenate}(\text{head}_1, \dots, \text{head}_h) W^O$$

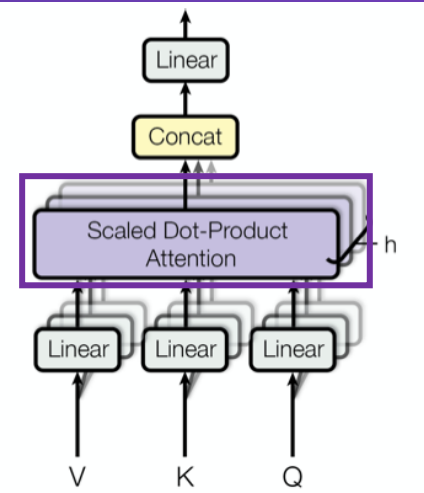
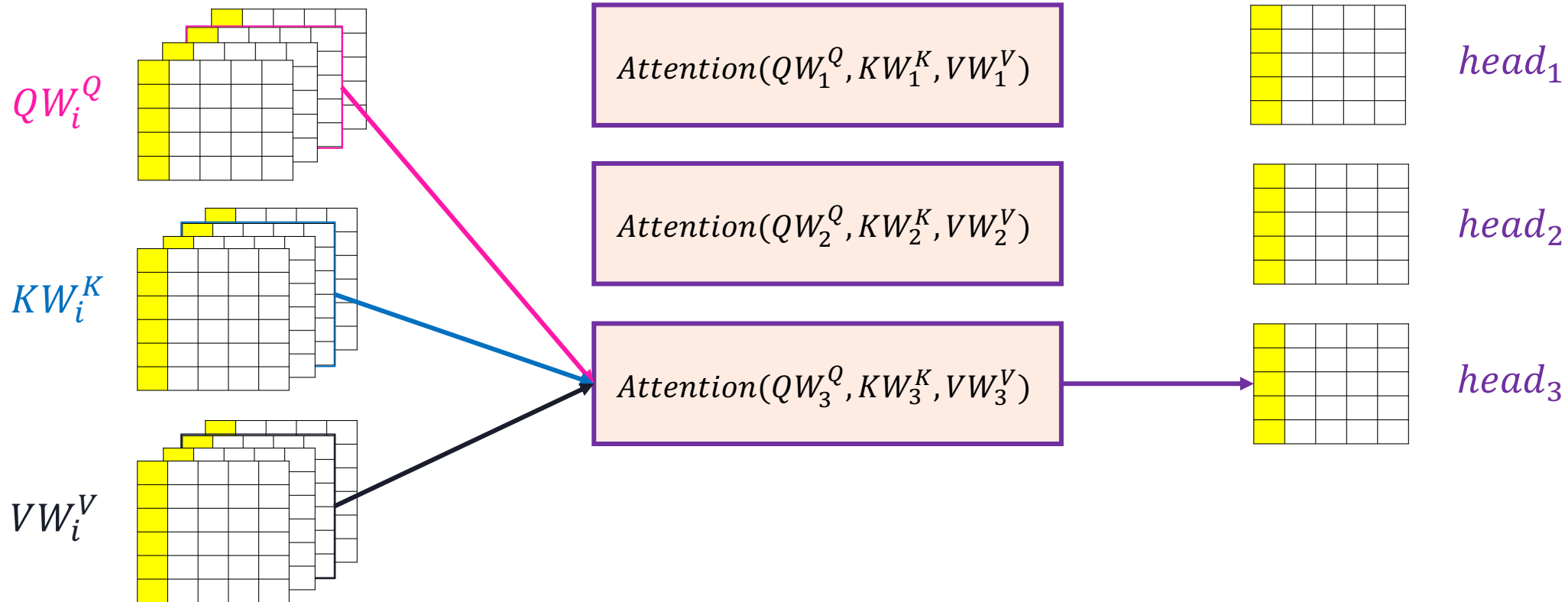
$$\text{where } \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = \text{concatenate}(\text{head}_1, \dots, \text{head}_h) W^O$$

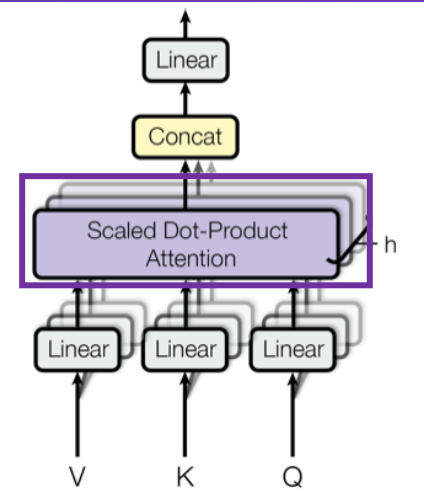
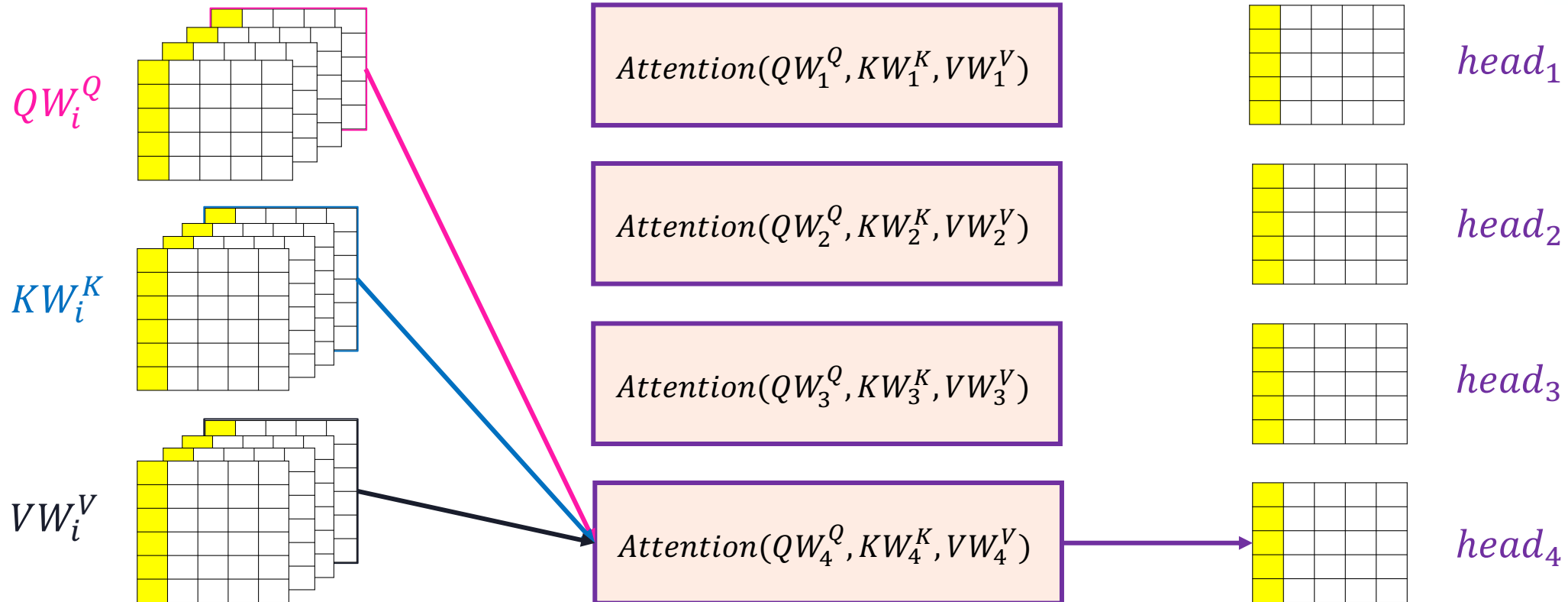
$$\text{where } \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = \text{concatenate}(\text{head}_1, \dots, \text{head}_h) W^O$$

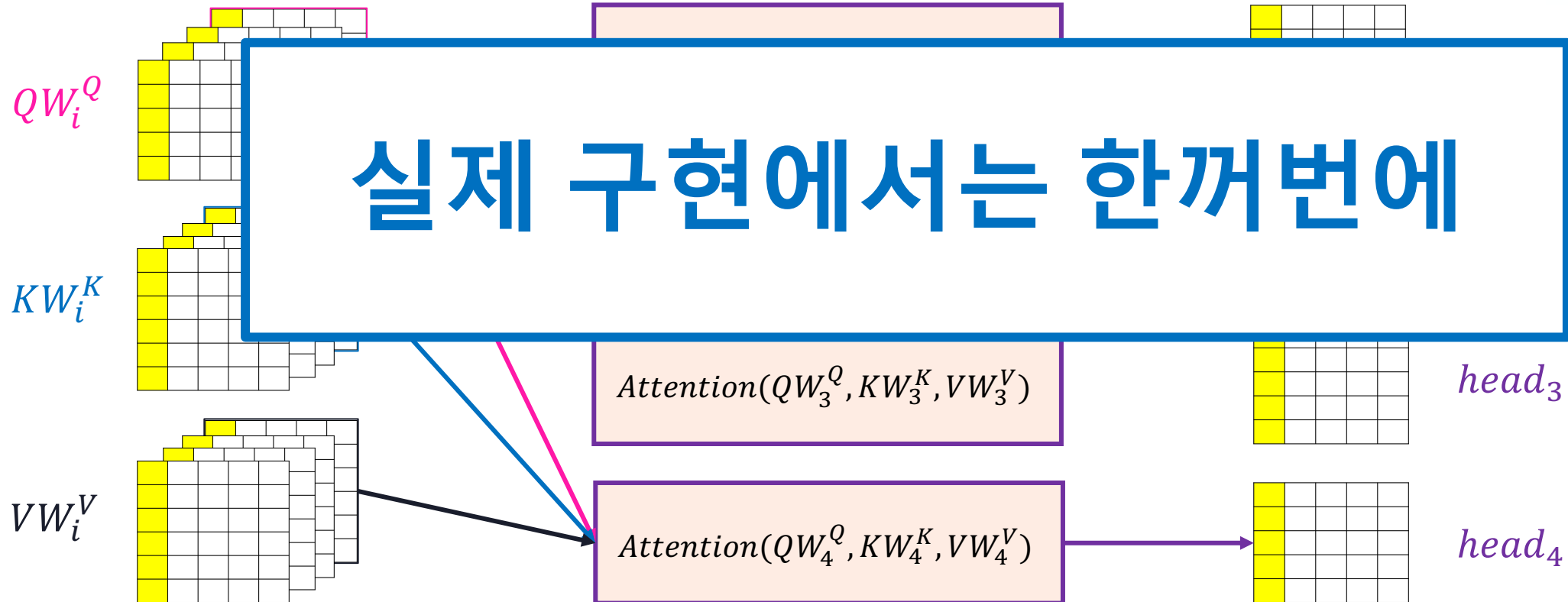
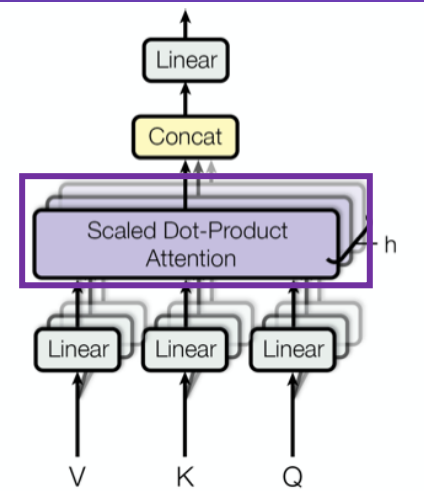
$$\text{where } \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = \text{concatenate}(\text{head}_1, \dots, \text{head}_h)W^O$$

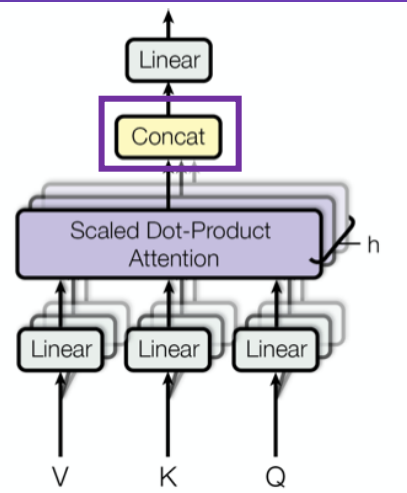
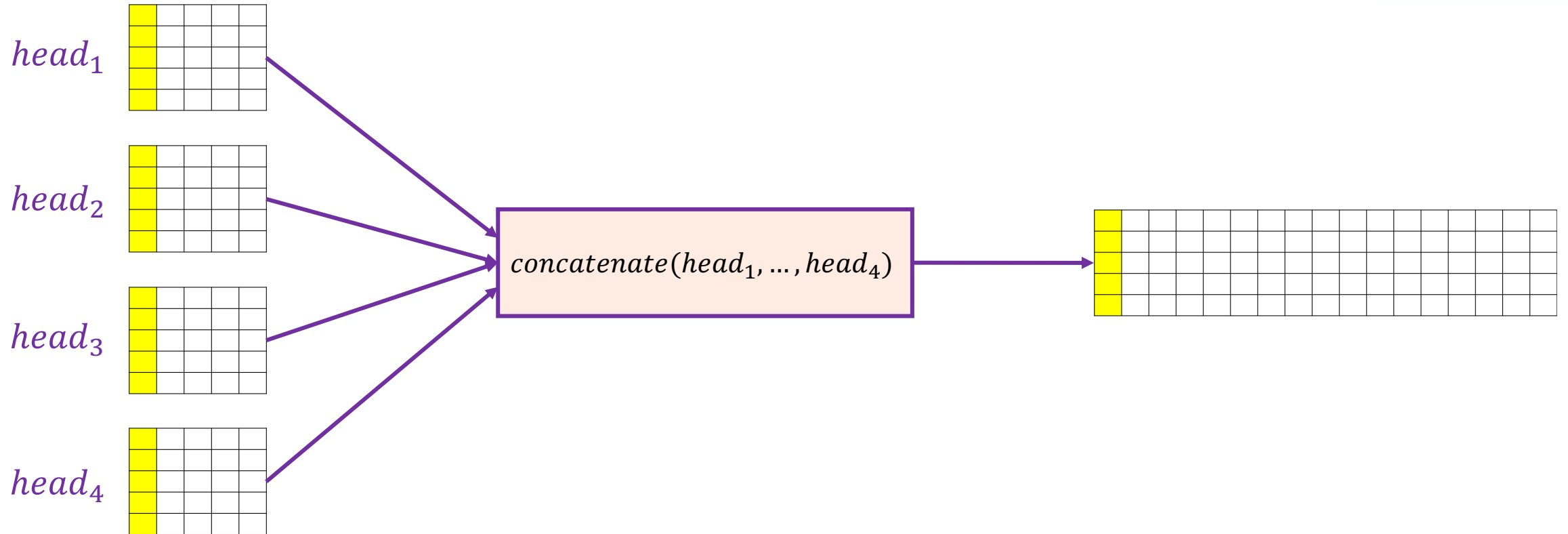
$$\text{where } \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = \text{concatenate}(\text{head}_1, \dots, \text{head}_h) W^O$$

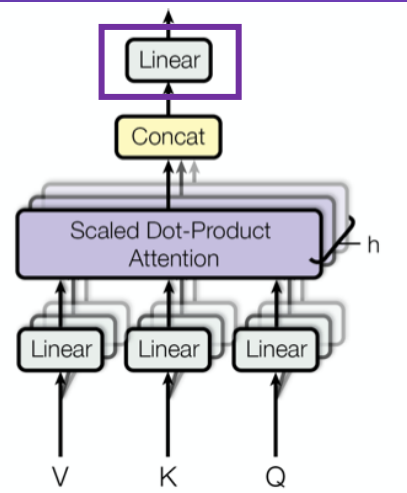
where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$



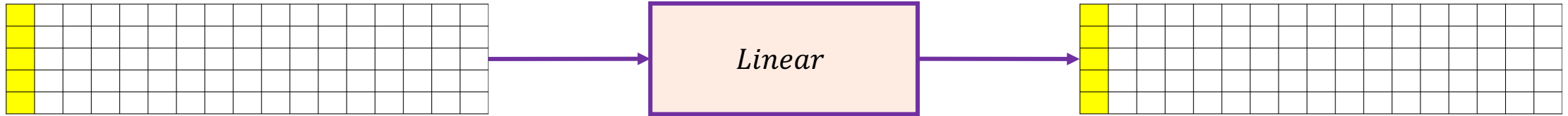
Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = \text{concatenate}(\text{head}_1, \dots, \text{head}_h) W^O$$

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



$\text{concatenate}(\text{head}_1, \dots, \text{head}_4)$

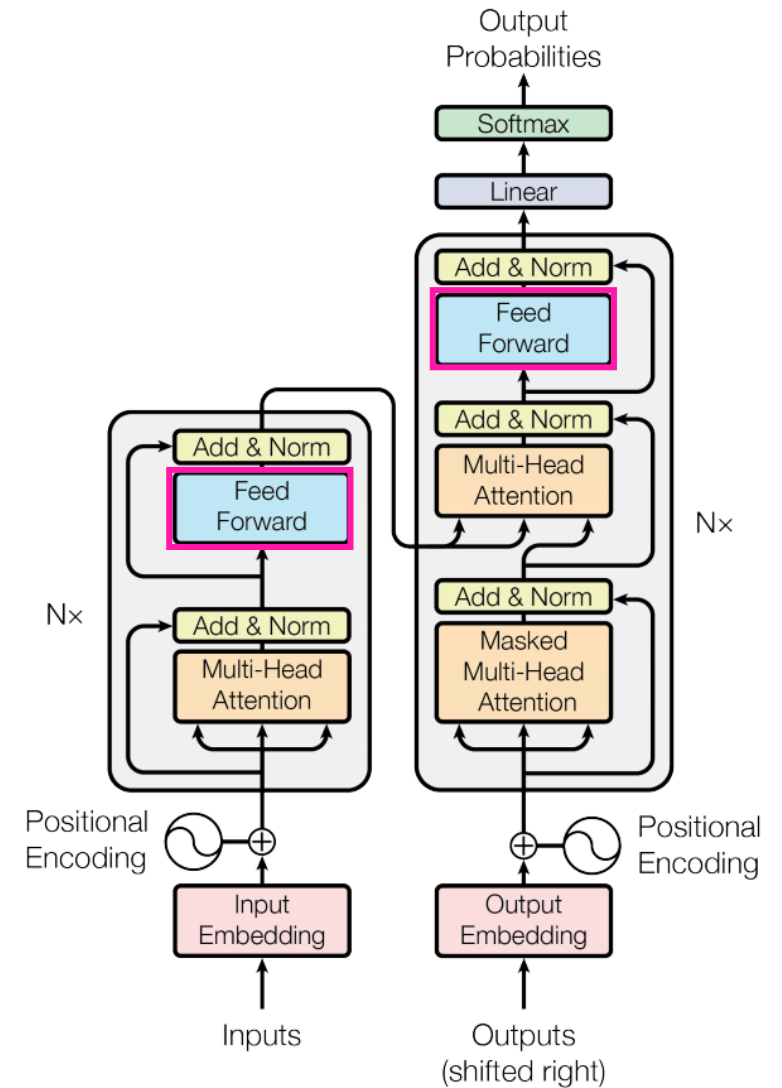




Position-wise Feed-Forward Network

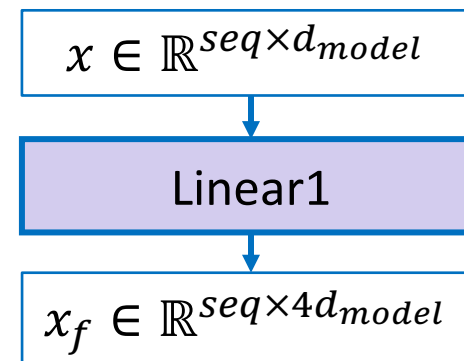
Position-wise Feed-Forward Network

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



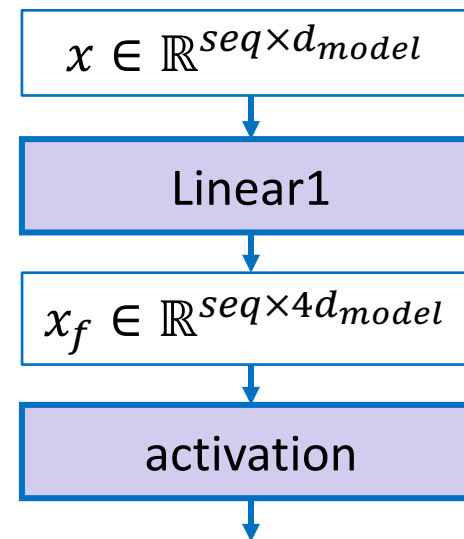
$$FFN(x) = \max(0, xW_1 + b_1) W_2 + b_2$$

$$xW_1 + b_1 \longrightarrow$$

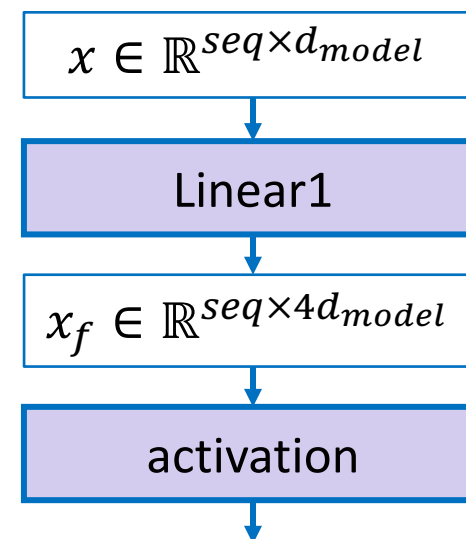
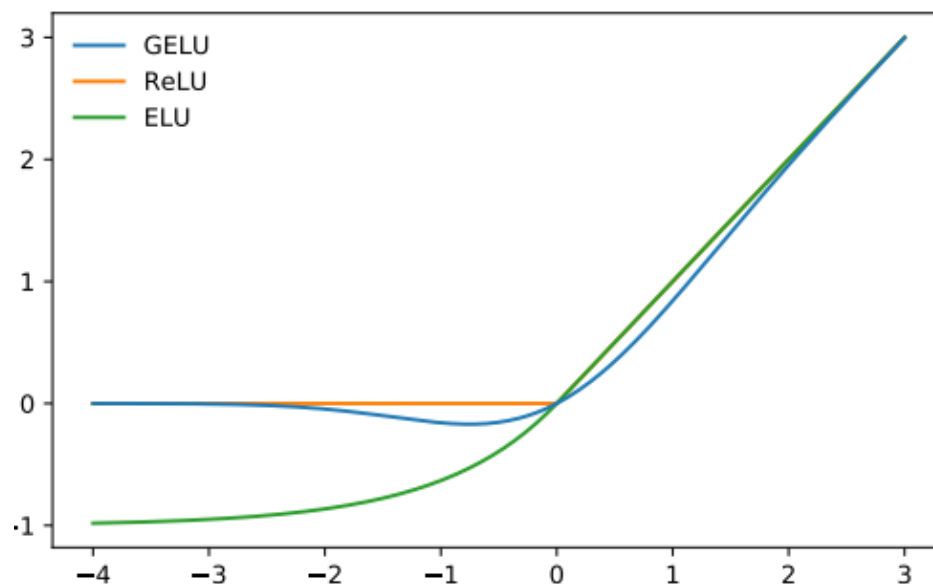


$$FFN(x) = \max(0, xW_1 + b_1) W_2 + b_2$$

$\max(0, xW_1 + b_1) \longrightarrow$

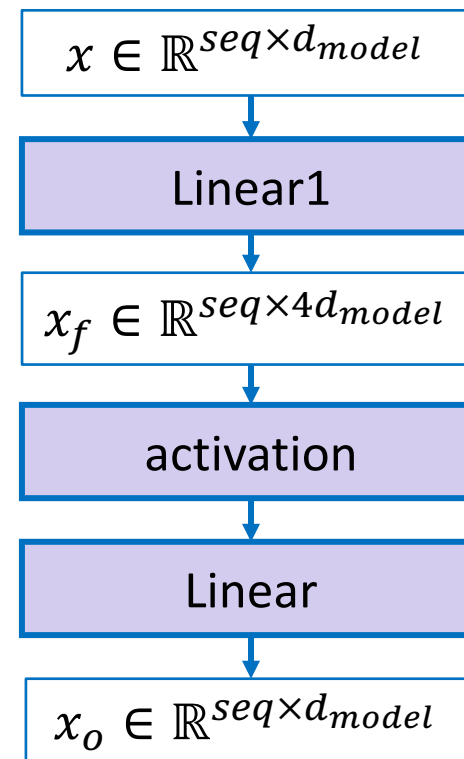


$$FFN(x) = \max(0, xW_1 + b_1) W_2 + b_2$$



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

$$\max(0, xW_1 + b_1) W_2 + b_2 \longrightarrow$$





Reference

- Attention is All You Need: <https://arxiv.org/abs/1706.03762>
- The Illustrated Transformer: <https://nlpinkorean.github.io/illustrated-transformer/>
- CS224n - Machine Translation, Seq2Seq and Attention:
<http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf>
- Yandex School of Data Analysis - Seq2seq and Attention :
https://github.com/yandexdataschool/nlp_course/blob/2019/resources/slides/nlp19_04_seq2seq_attention.pdf
- Learning Spoons - Attention :
<https://github.com/changwookjun/learningspoons/blob/master/Slide/Lecture6.pdf>