

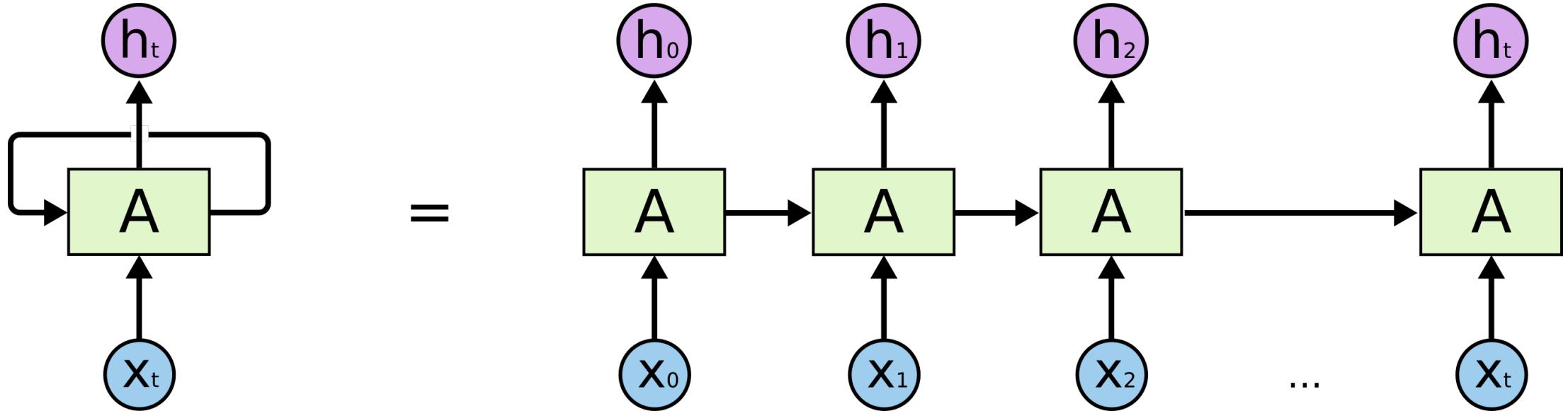
Attention is All You Need

한양대 HAI 지영채

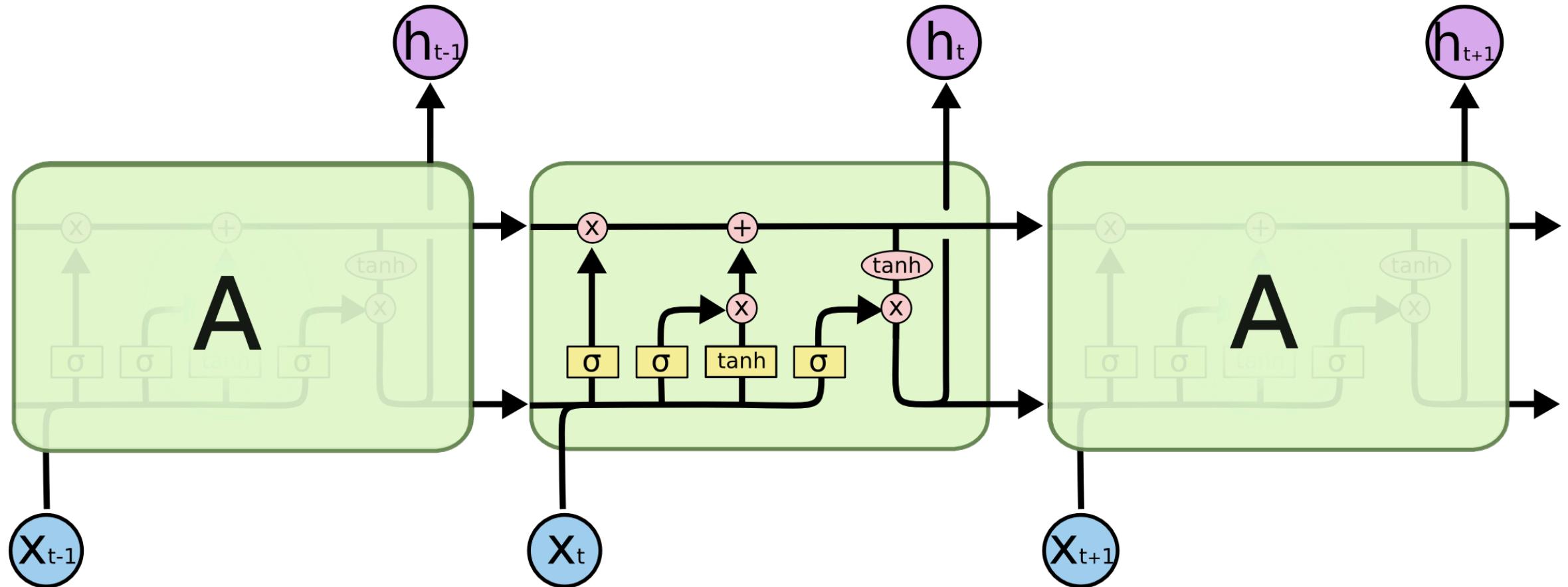
Claims

- ① Superior in **quality** 28.4 BLEU WMT English to German Translation
- ② More **Parallelizable**
- ③ Less **Train Time**

RNN Review

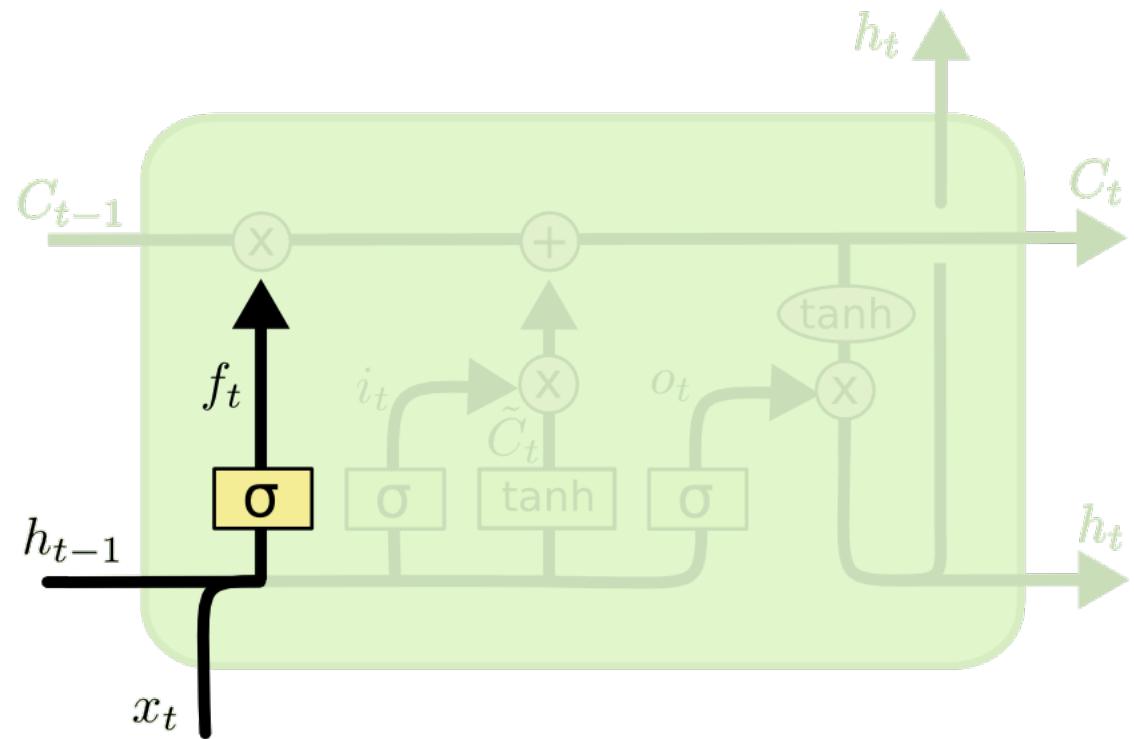


RNN Review – LSTM (Long and Short Term Memory)



RNN Review – LSTM (Long and Short Term Memory)

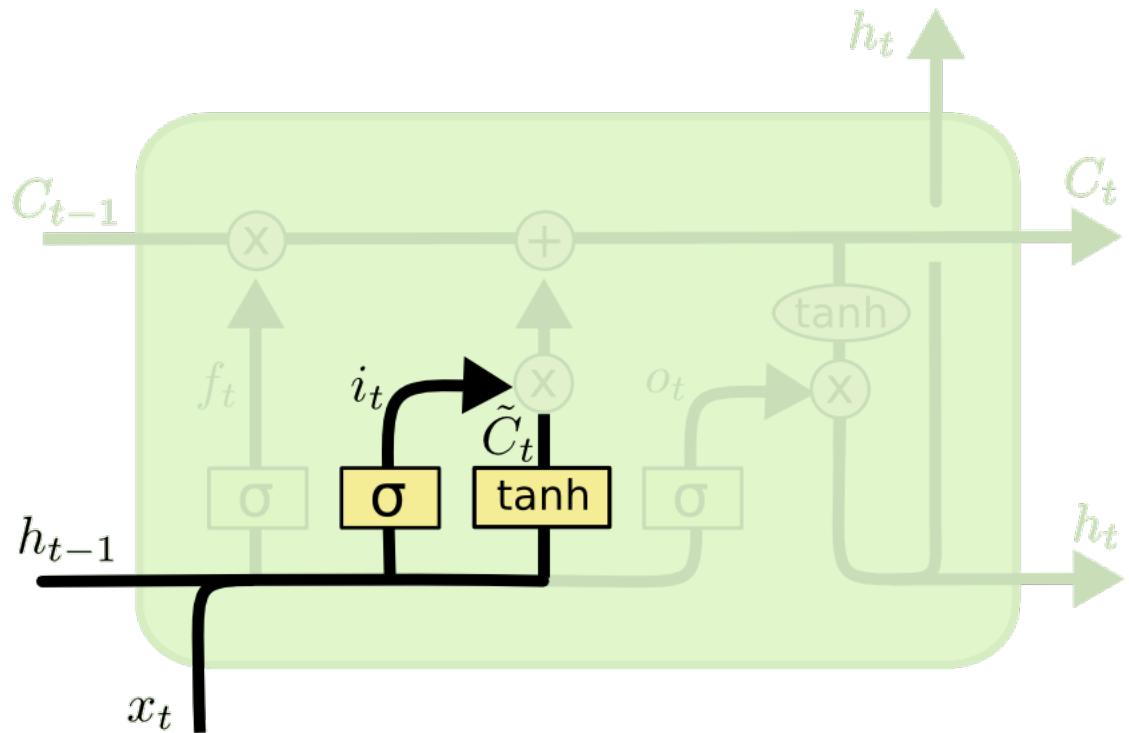
Forget Long Term Memory



$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$$

RNN Review – LSTM (Long and Short Term Memory)

Add Long Term Memory

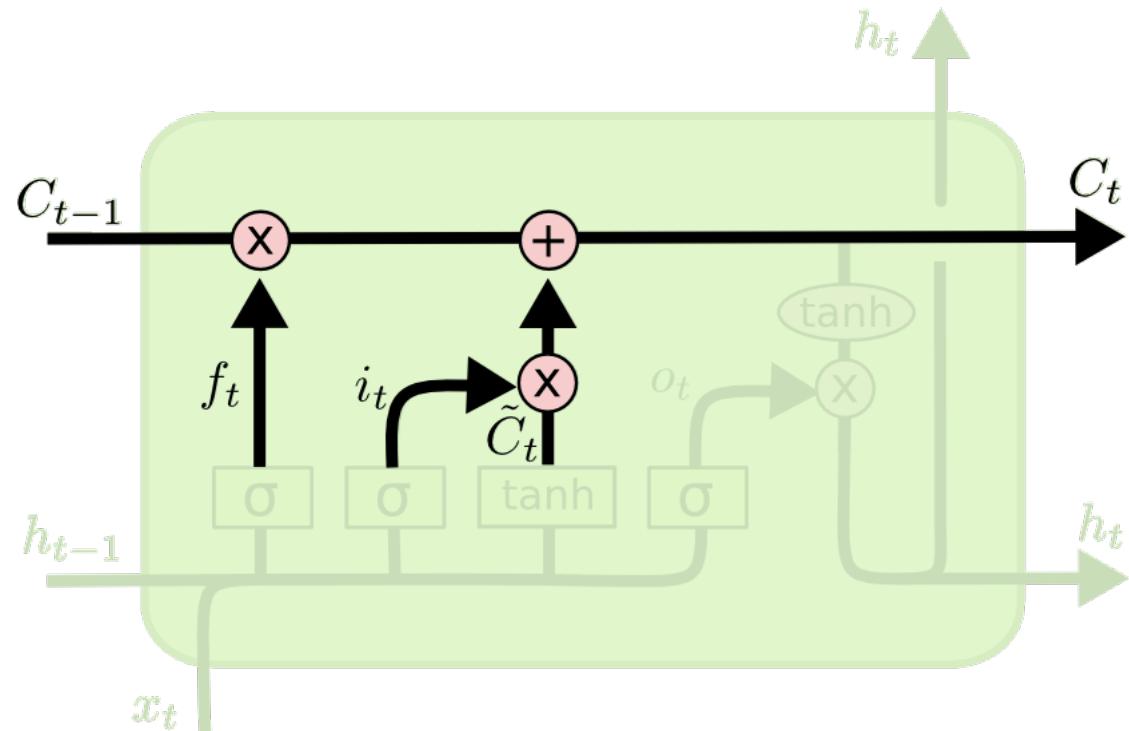


$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

RNN Review – LSTM (Long and Short Term Memory)

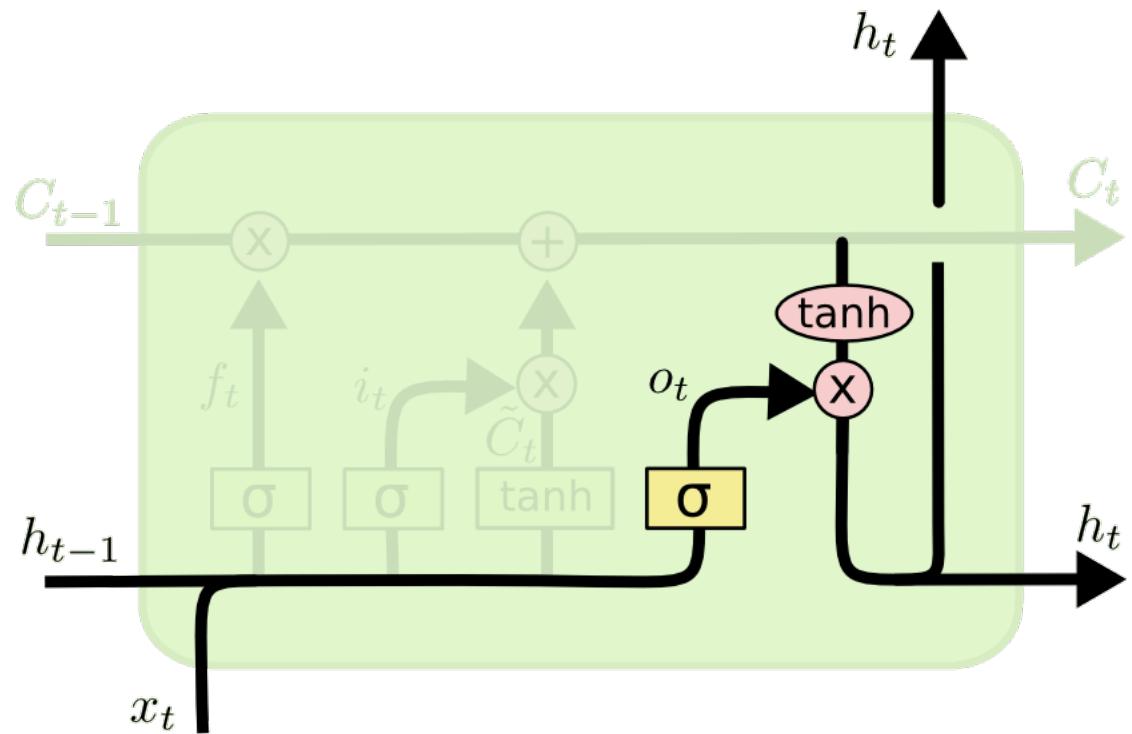
Update Long Term Memory



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

RNN Review – LSTM (Long and Short Term Memory)

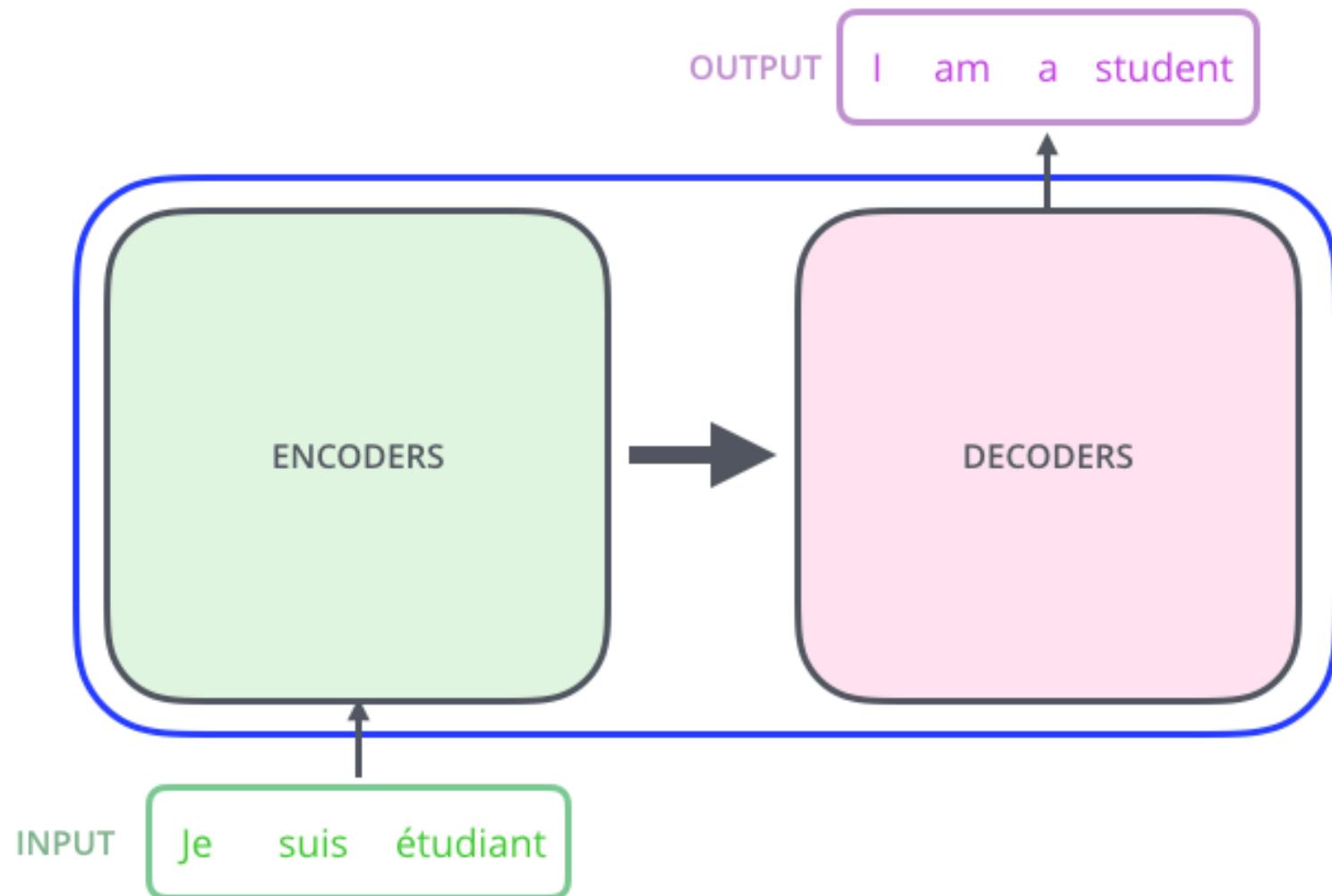
Update Short Term Memory



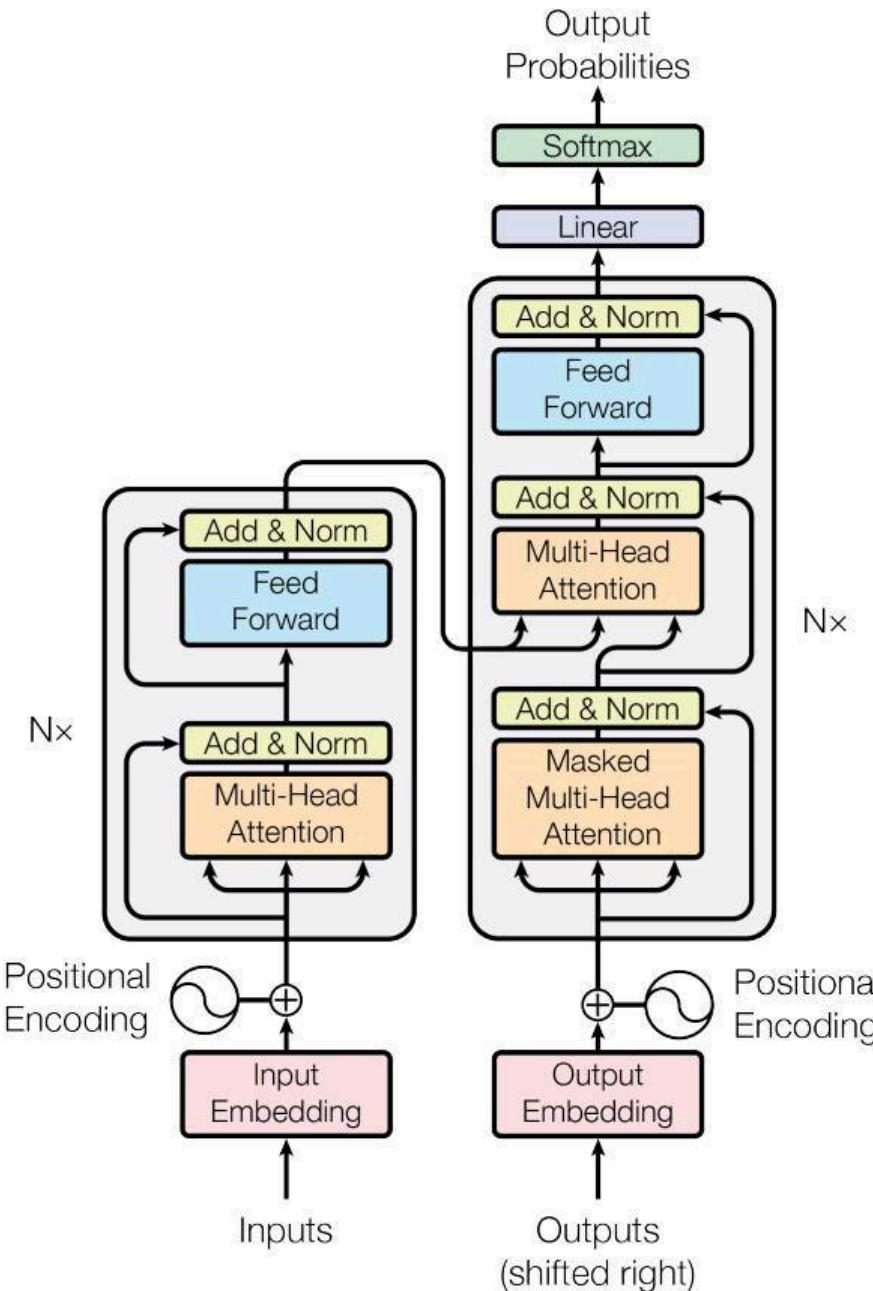
$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh (C_t)$$

Transformer

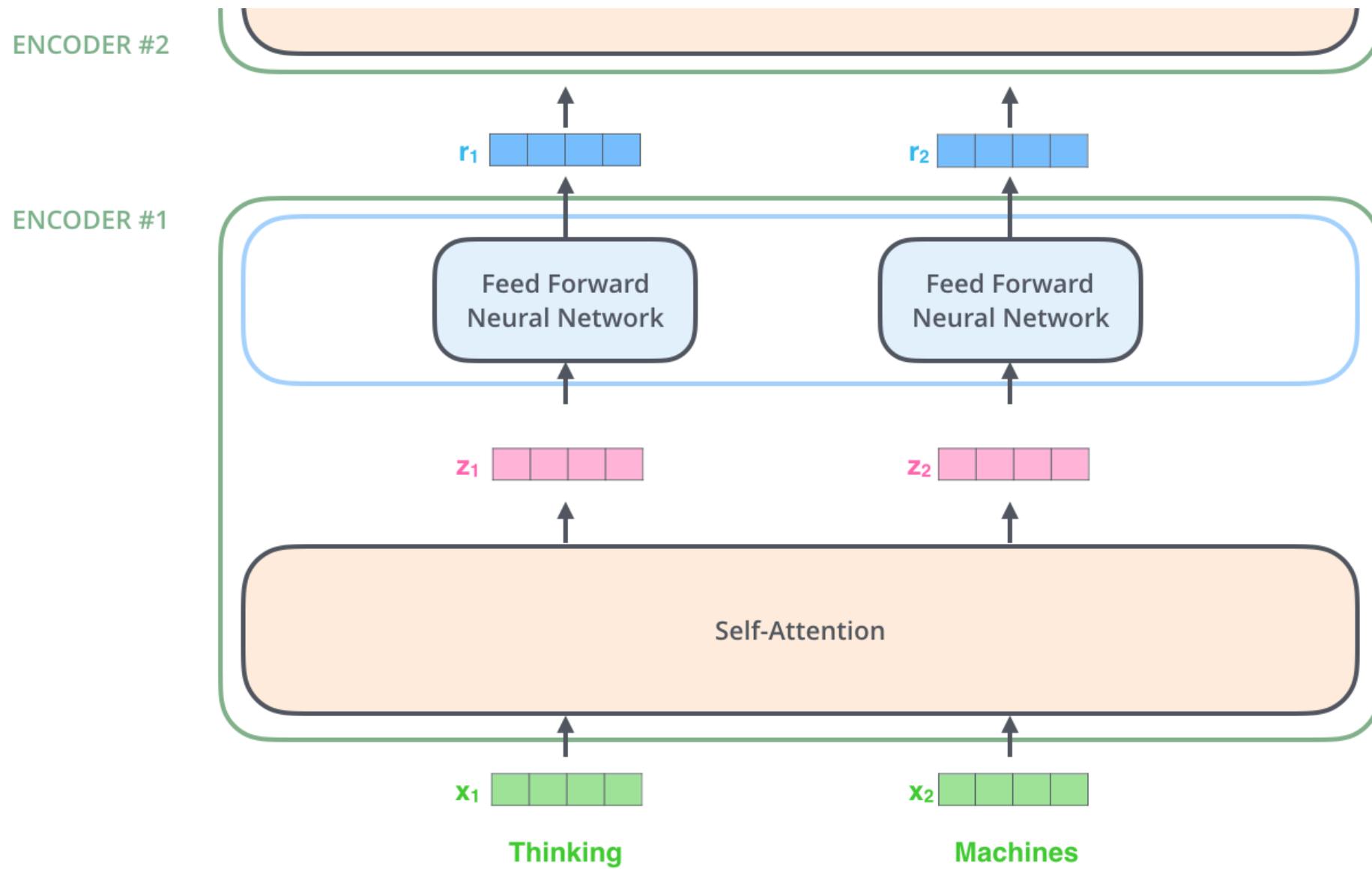
High-Level Look



High-Level Look



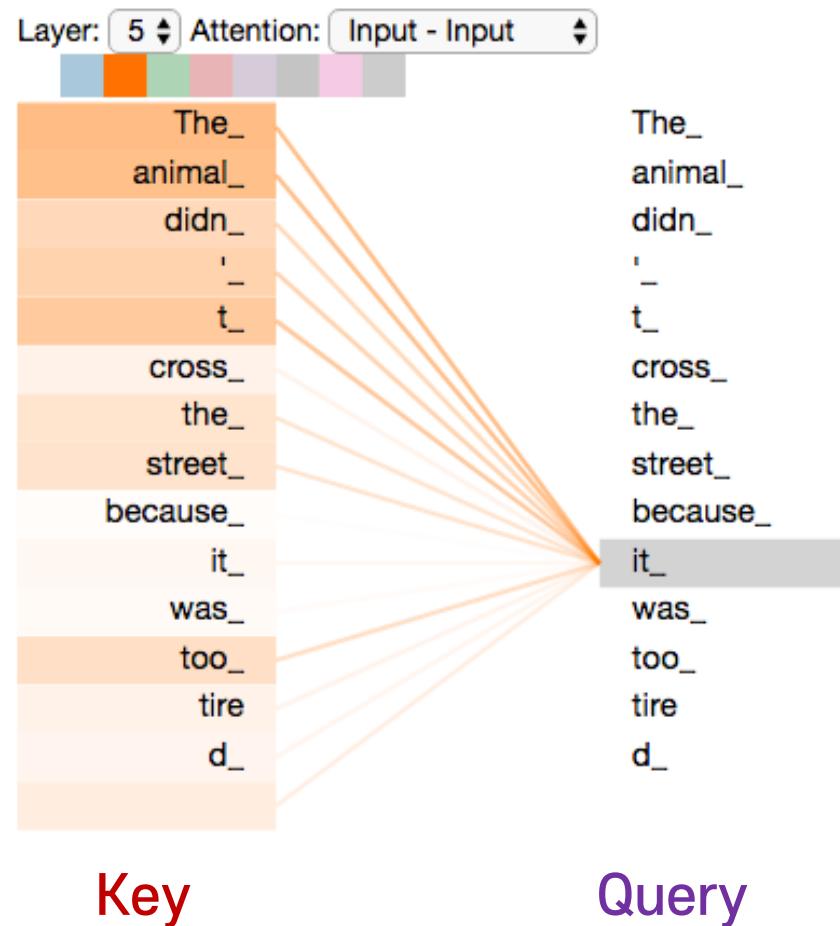
Encoder



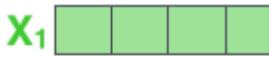
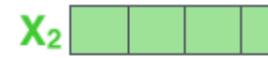
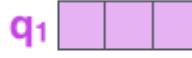
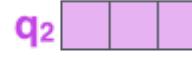
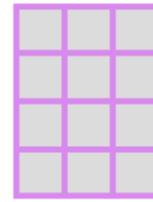
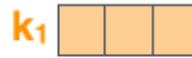
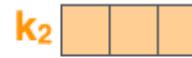
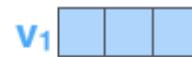
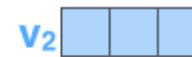
Attention

Attention

Query 가 들어왔을 때 어떤 key에 집중할 것인가?

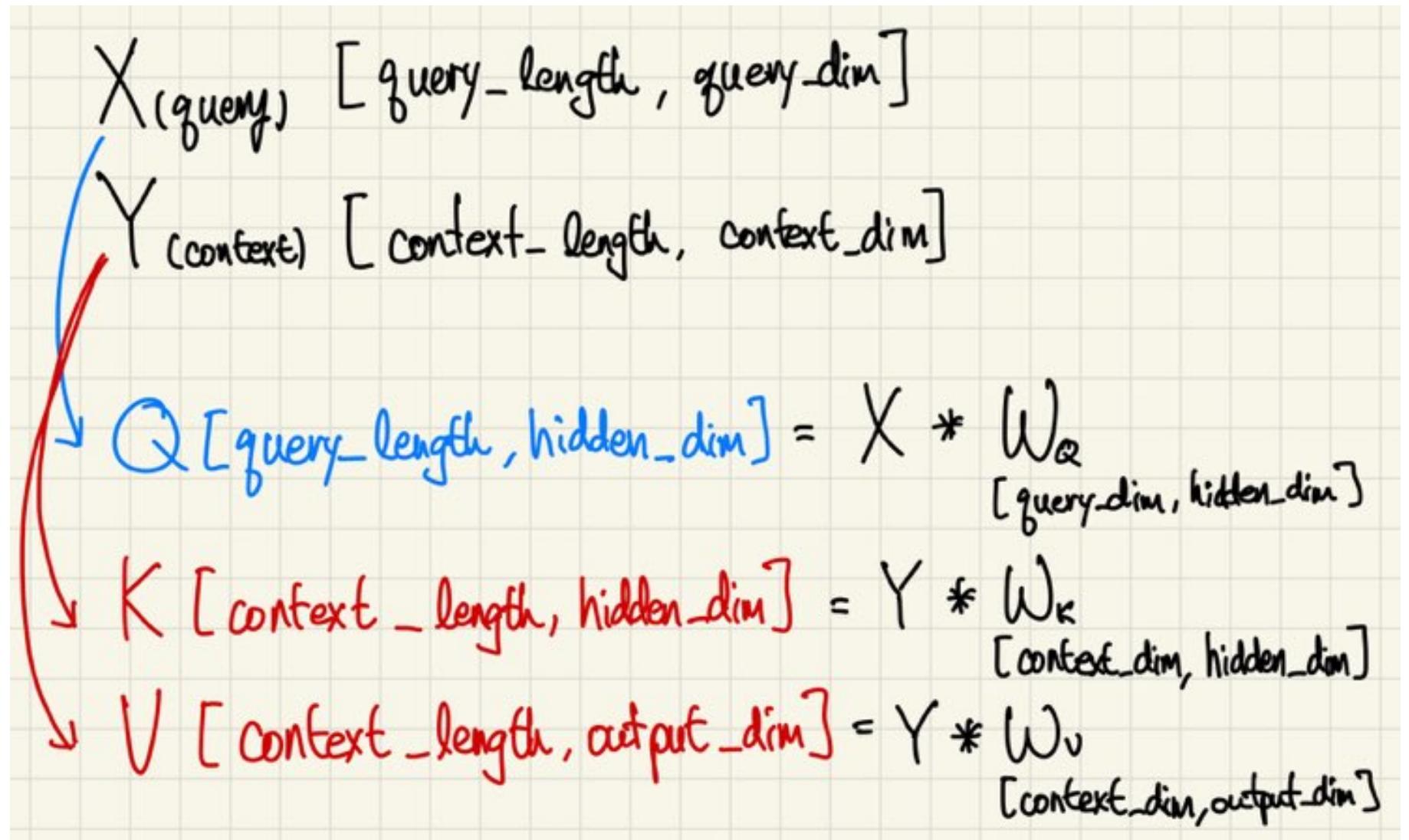


Attention

Input	Thinking	Machines	
Embedding	x_1 	x_2 	Trainable
Queries	q_1 	q_2 	 W^Q
Keys	k_1 	k_2 	 W^K
Values	v_1 	v_2 	 W^V

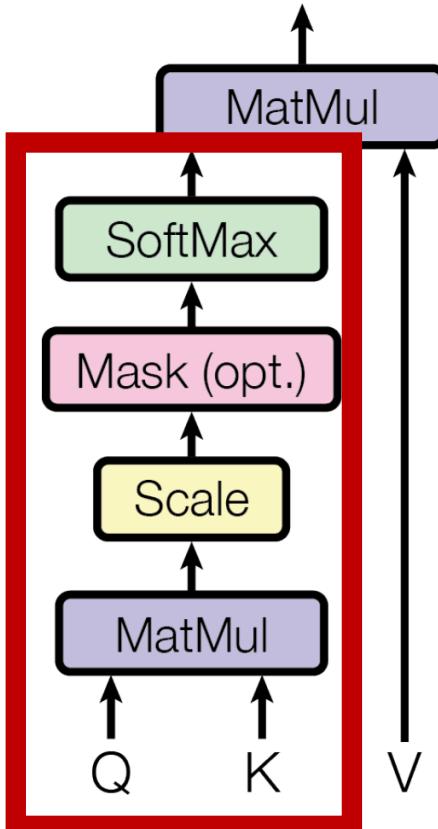
Attention – Scaled Dot Product Attention

Input	Thinking	Machines
Embedding	X_1 [green 4x4]	X_2 [green 4x4]
Queries	q_1 [purple 2x2]	q_2 [purple 2x2]
Keys	k_1 [orange 2x2]	k_2 [orange 2x2]
Values	v_1 [blue 2x2]	v_2 [blue 2x2]



Attention – Scaled Dot Product Attention

Scaled Dot-Product Attention



$$Q\text{-KMap} = \text{softmax}\left(\frac{1}{\sqrt{h.\text{dim}}} Q \cdot K^T\right)$$

$[q\text{-len}, c\text{-len}]$

$Q_1 \quad Q_2 \quad \dots \quad Q_{c\text{-len}}$

$Q_1 \quad Q_2 \quad \dots \quad Q_{c\text{-len}}$

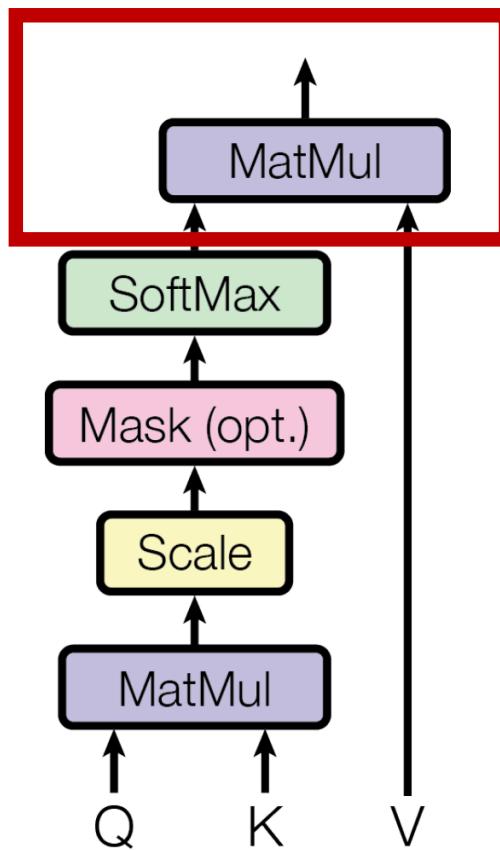
합 1.
(Q_2 가 $K_1 \sim K_{c\text{-len}}$ 과
집중되는 바운)

$Q_{q\text{-len}}$

This block contains handwritten notes explaining the computation of the Query-Key Map. It shows two vectors, Q_1 and Q_2 , and a sequence of keys $K_1, K_2, \dots, K_{c\text{-len}}$. A red box highlights the first few elements of Q_2 and $K_1, K_2, \dots, K_{c\text{-len}}$, with the text "합 1." (sum 1) and the note "(Q_2 가 $K_1 \sim K_{c\text{-len}}$ 과 집중되는 바운)".

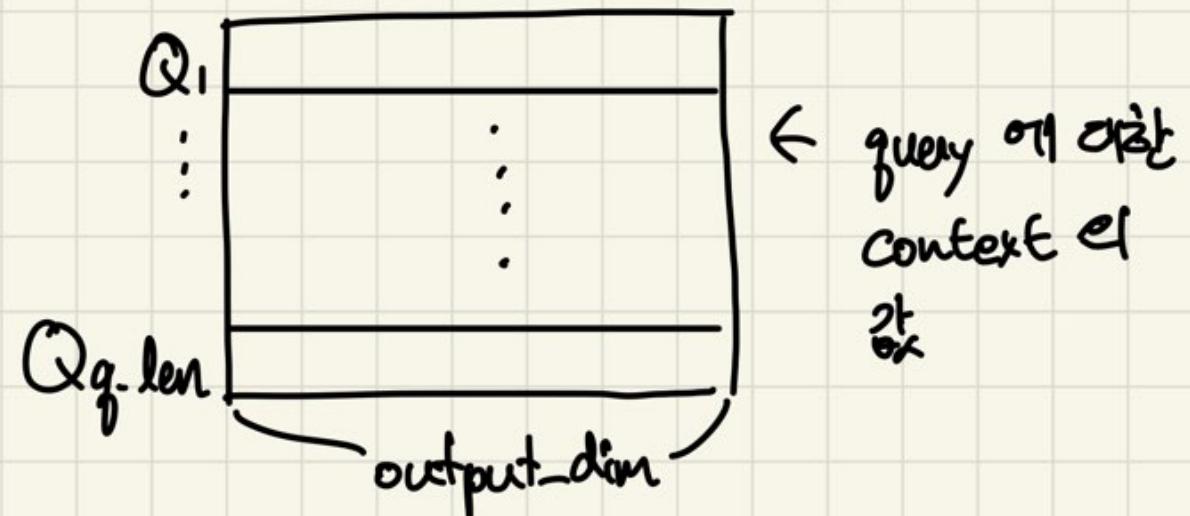
Attention – Scaled Dot Product Attention

Scaled Dot-Product Attention



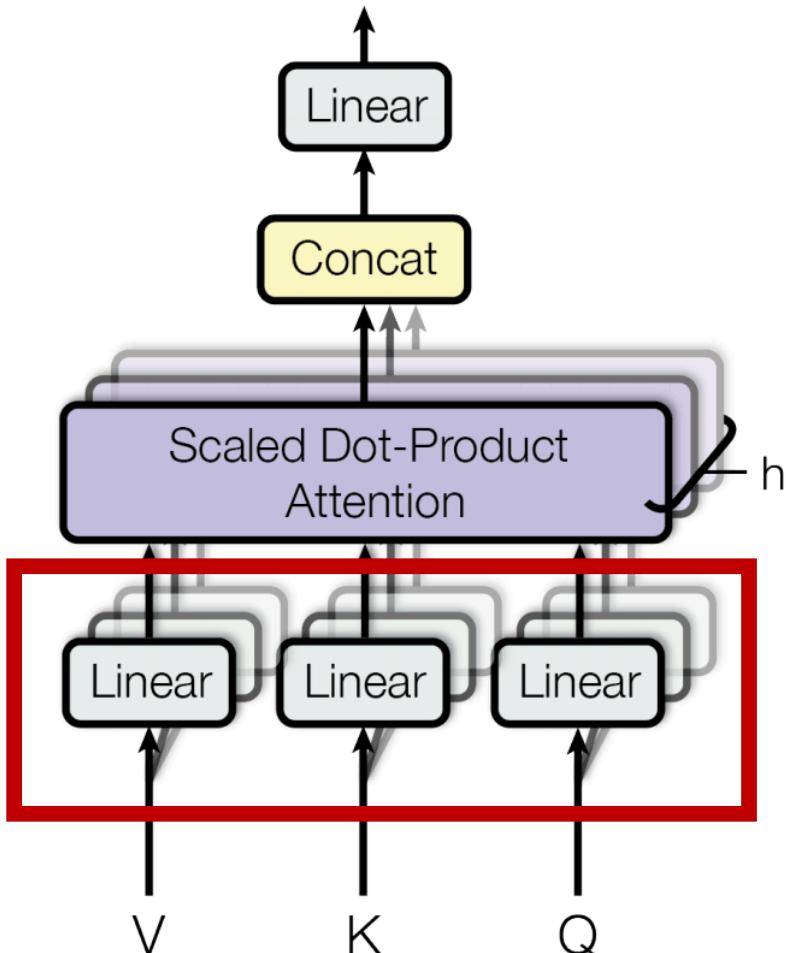
$$\text{Output} = \text{Q-K Map} * \text{Value}$$

[query_len, output_dim] [context_len, output_dim]



Attention – Multi-Head Attention

Multi-Head Attention



$X_{(\text{query})}$ [query_length, query_dim]

$Y_{(\text{context})}$ [context_length, context_dim]

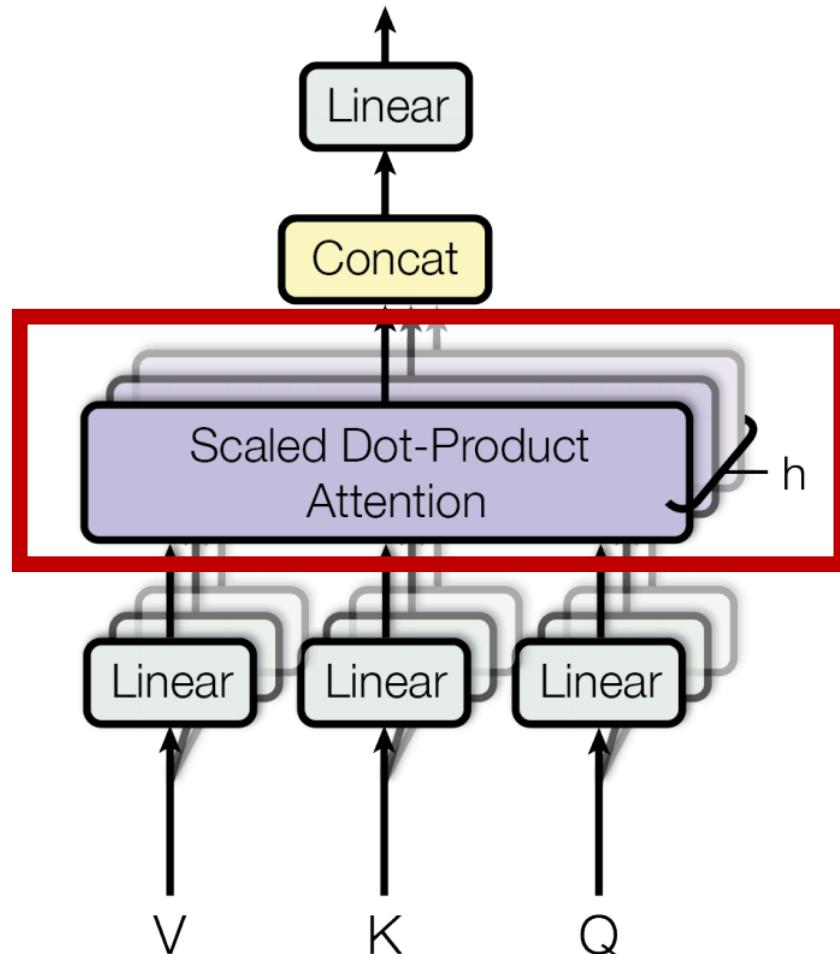
Q [query_length, hidden_dim] = $X * W_Q$
[query_dim, hidden_dim] *
 $*h$

K [context_length, hidden_dim] = $Y * W_K$
[context_dim, hidden_dim] *
 $*h$

V [context_length, output_dim] = $Y * W_V$
[context_dim, output_dim] *
 $*h$

Attention – Multi-Head Attention

Multi-Head Attention



$$Q\text{-KMap} = \text{softmax}\left(\frac{1}{\sqrt{h \cdot \text{dim}}} Q \cdot K^\top\right)$$

$[h, q\text{-len}, c\text{-len}]$

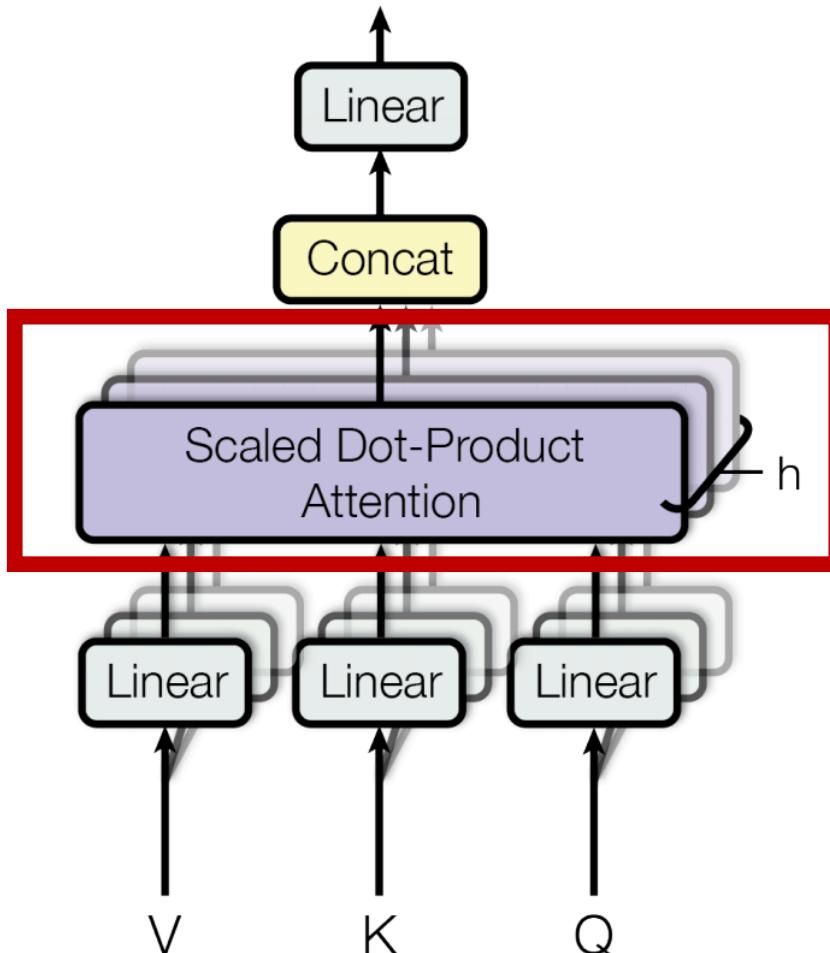
$Q_1 \quad Q_2 \quad \dots \quad Q_{q\text{-len}}$

$K_1 \quad K_2 \quad \dots \quad K_{c\text{-len}}$

합 1.
 $(Q_2 \text{가 } K_1 \sim K_{c\text{-len}} \text{ 까지 } \text{집중되는 } \text{부분})$

Attention – Multi-Head Attention

Multi-Head Attention



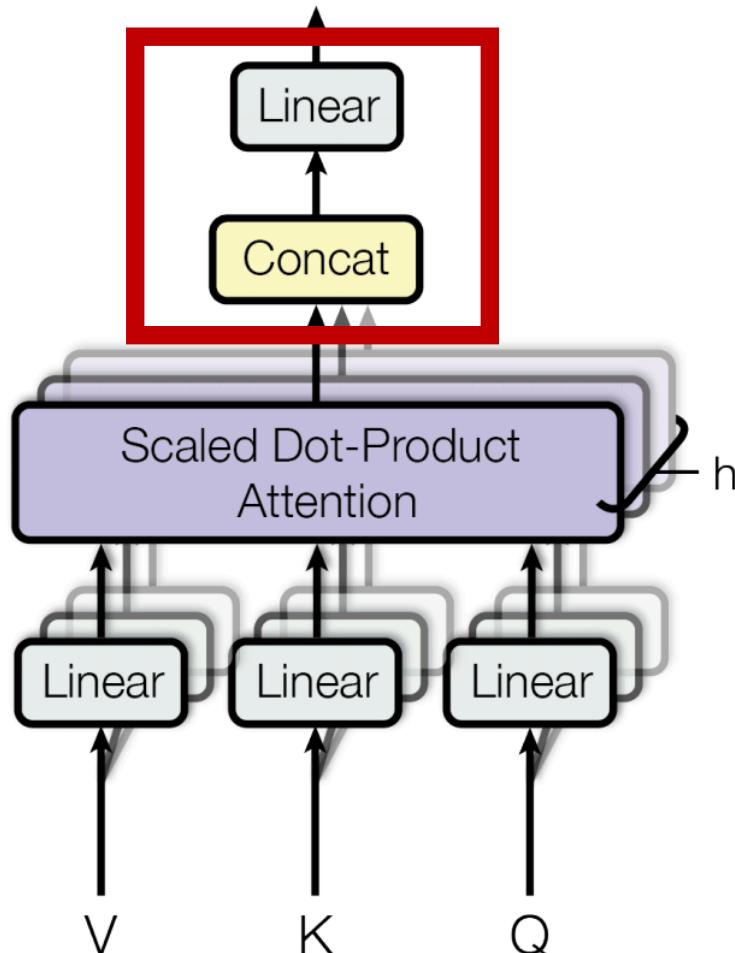
Output = $\text{Q-K Map} * \text{Value}$
[$h, \text{query_len}, \text{output_dim}$] [$h, \text{context_len}, \text{output_dim}$]

Q_1 \vdots \vdots \vdots
 $Q_{q.\text{len}}$ $\underbrace{\hspace{10em}}_{\text{output_dim}}$

← query 와 context
의 차이

Attention – Multi-Head Attention

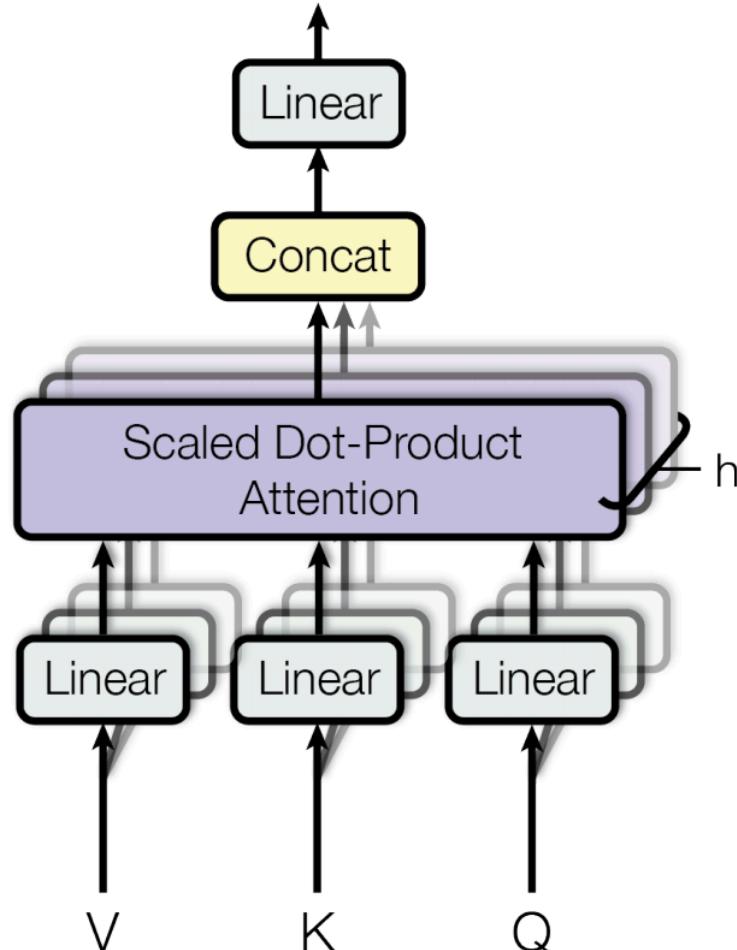
Multi-Head Attention



Output $[query_len, output_dim * h]$
↓
Linear ($output_dim * h, real_output_dim$)
 $[query_len, \underbrace{real_output_dim}_{\approx query_dim}]$

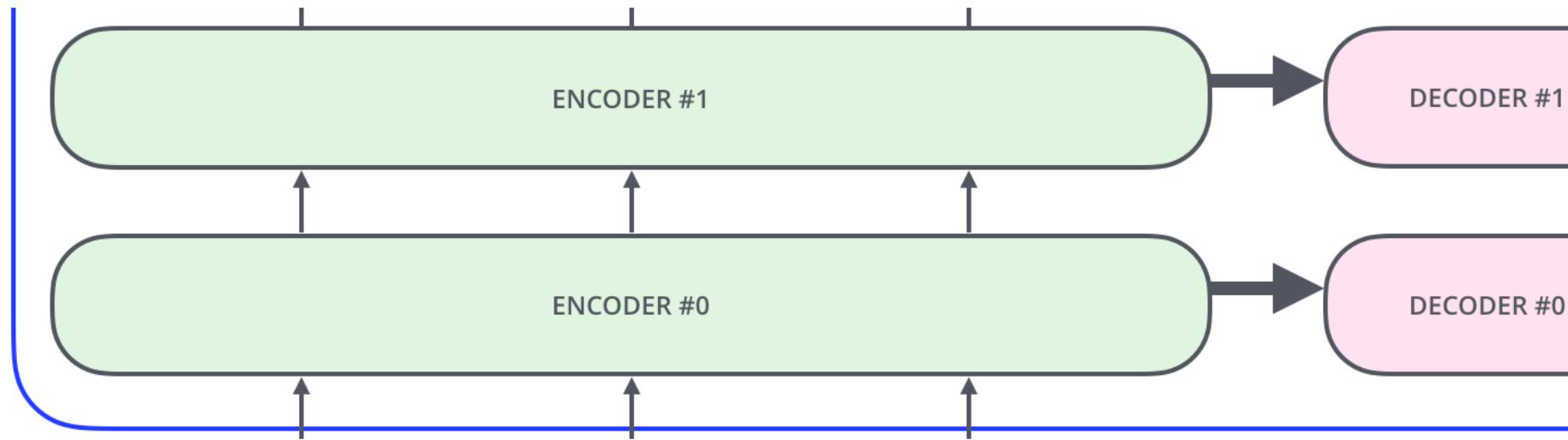
Attention – Multi-Head Attention

Multi-Head Attention



$$h = 8$$
$$d_{\text{model}} = 512$$

Positional Encoding



EMBEDDING
WITH TIME
SIGNAL

x_1

X₂

X₃

POSITIONAL ENCODING

t_1 

t₂ |

t₃

EMBEDDINGS

x_1 |

x_2

x_3

INPUT

Je

suis

étudiant

Je

suis

étudiant

INP

S

étudiant

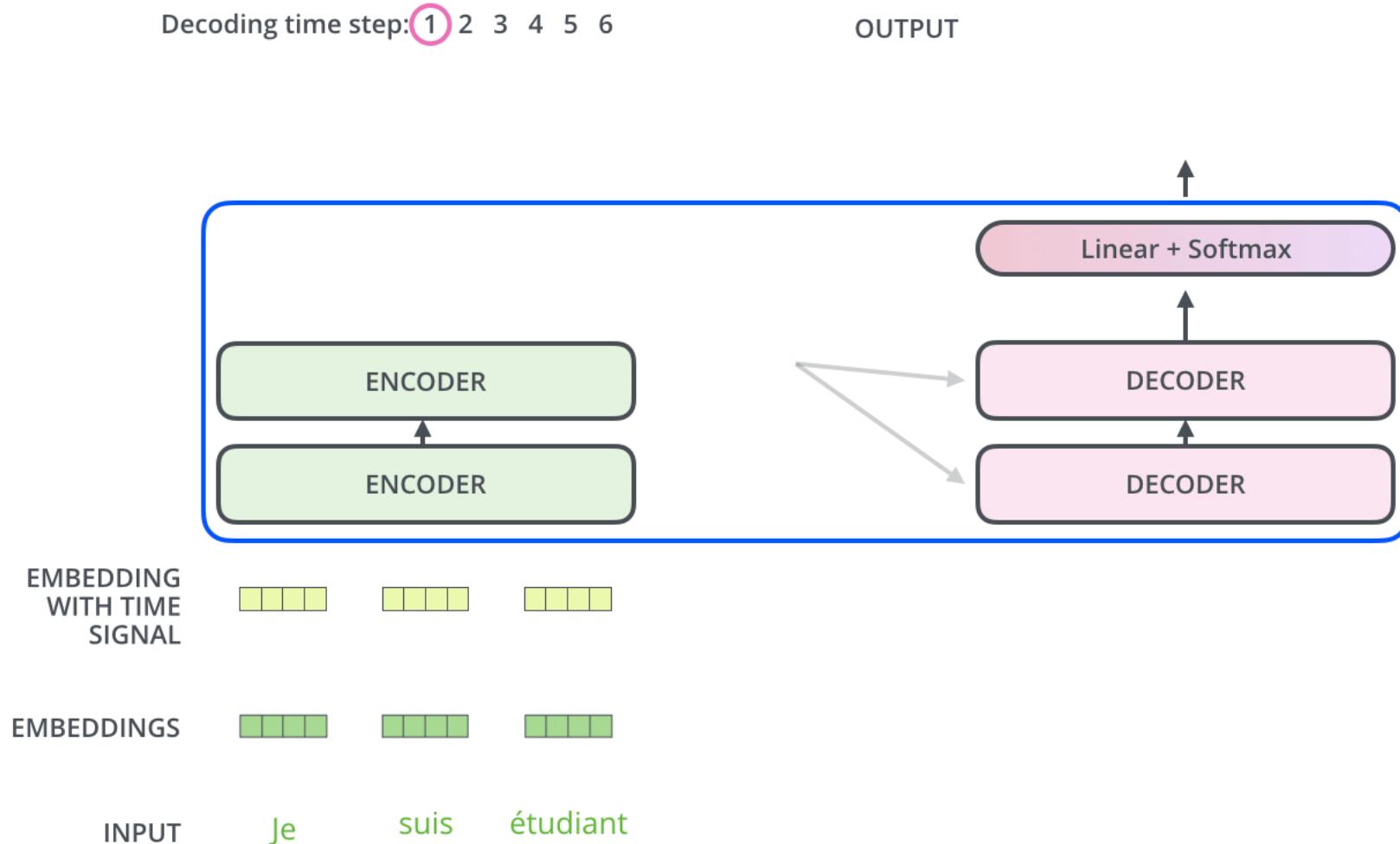
POSITION ENCODI

0	0	1
---	---	---

0.84 0.0001 0.54 1

0.91 0.0002 -0.42 1

Decoder



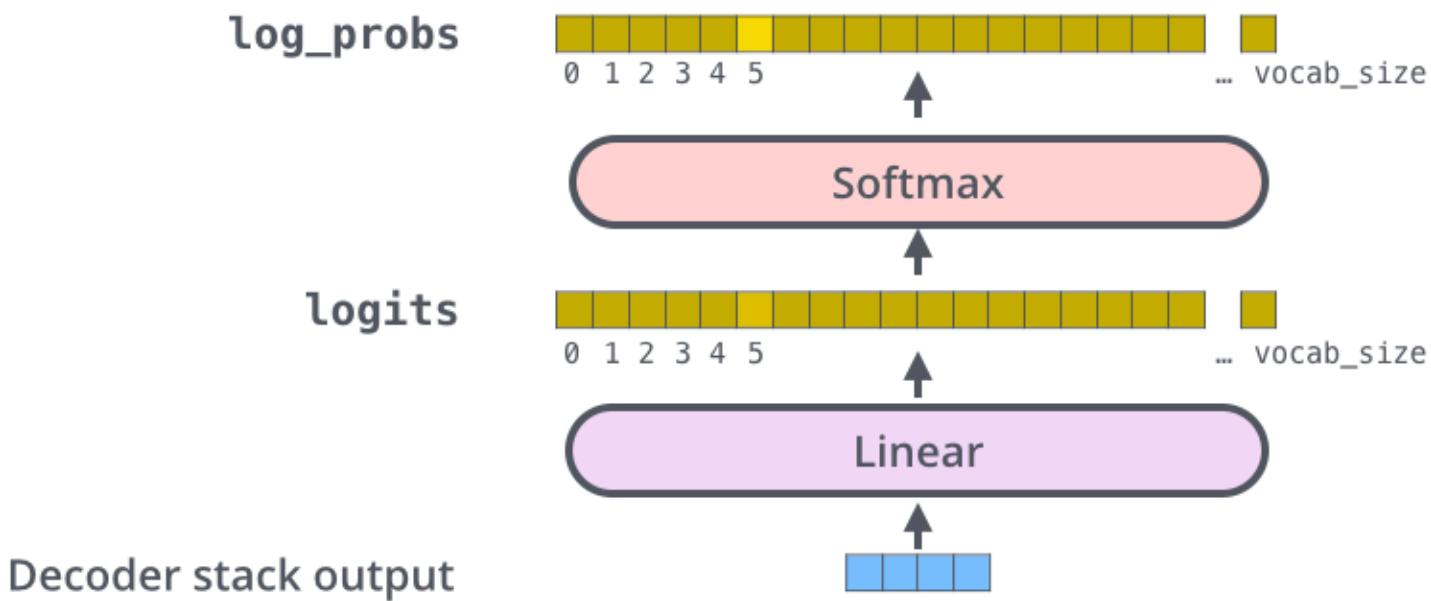
Decoder

Which word in our vocabulary
is associated with this index?

am

Get the index of the cell
with the highest value
(**argmax**)

5



Decoder

Target Model Outputs

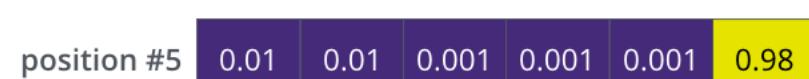
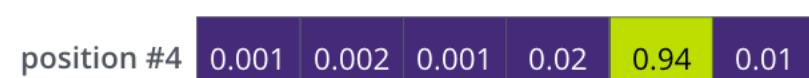
Output Vocabulary: a am I thanks student <eos>



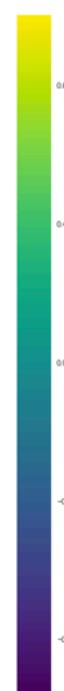
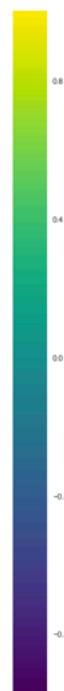
a am I thanks student <eos>

Trained Model Outputs

Output Vocabulary: a am I thanks student <eos>



a am I thanks student <eos>



≡