Attention Is All You Need
(How to implement Transformer)

2020. 01. 06

Juhyuk Lee

sehkmg@gmail.com

Table of Contents

- **NMT Basics and Encoder-Decoder Model**

- **Abstract View of Transformer Architecture**

- **Dive into the Transformer Architecture**

Table of Contents

- **NMT Basics and Encoder-Decoder Model**

- Abstract View of Transformer Architecture

- Dive into the Transformer Architecture

NMT Basics and Encoder-Decoder Model

- Neural Machine Translation
  - Dataset만 이용해서 black box model을 end-to-end로 학습시켜 번역.
  - 기본적인 Deep Learning Framework.

NMT Basics – Dataset 준비

- Dataset 준비

Source (Train)	**Target (Train)**
Je suis etudiant.	I am a student.
Quel mois?	What month?

Source (Valid)	**Target (Valid)**
Je suis medecin.	I am a doctor.
Ce mois-ci?	This month?

Source (Test)	**Target (Test)**
Je suis enseignant.	I am a teacher.
Quel jour?	What day?

NMT Basics – Data preprocessing

- Make batches - tokenize

Source Batch (Train)	**Target Batch (Train)**
[Je, suis, etudiant, .]	[I, am, a, student, .]
[Quel, mois, ?]	[What, month, ?]

Source Batch (Valid)	**Target Batch (Valid)**
[Je, suis, medecin, .]	[I, am, a, doctor, .]
[Ce, mois, -, ci, ?]	[This, month, ?]

Source Batch (Test)

[Je, suis, enseignant, .]
[Quel, jour, ?]

NMT Basics – Data preprocessing

- Make batches – add <sos>, <eos> token

Source Batch (Train)	**Target Batch (Train)**
[Je, suis, etudiant, .]	[<sos>, I, am, a, student, ., <eos>]
[Quel, mois, ?]	[<sos>, What, month, ?, <eos>]

Source Batch (Valid)	**Target Batch (Valid)**
[Je, suis, medecin, .]	[<sos>, I, am, a, doctor, ., <eos>]
[Ce, mois, -, ci, ?]	[<sos>, This, month, ?, <eos>]

Source Batch (Test)

[Je, suis, enseignant, .]
[Quel, jour, ?]

NMT Basics – Data preprocessing

- Make batches – padding

Source Batch (Train)	**Target Batch (Train)**
[Je, suis, etudiant, .]	[<sos>, I, am, a, student, ., <eos>]
[Quel, mois, ?, <pad>]	[<sos>, What, month, ?, <eos>, <pad>, <pad>]

Source Batch (Valid)	**Target Batch (Valid)**
[Je, suis, medecin, ., <pad>]	[<sos>, I, am, a, doctor, ., <eos>]
[Ce, mois, -, ci, ?]	[<sos>, This, month, ?, <eos>, <pad>, <pad>]

Source Batch (Test)

[Je, suis, enseignant, .]
[Quel, jour, ?, <pad>]

NMT Basics – Data preprocessing

- Numericalize – make vocabulary from train dataset

Source (Train)	**Target (Train)**
Je suis etudiant.	I am a student.
Quel mois?	What month?

Source Vocab	**Target Vocab**
0: <sos>	0: <sos>
1: <eos>	1: <eos>
2: <pad>	2: <pad>
3: <unk>	3: <unk>
4: Je	4: I
5: suis	5: am
6: etudiant	6: a
7: .	7: student
8: Quel	8: .
9: mois	9: What
10: ?	10: month
11: …..	11: ?
	12: …..

NMT Basics – Data preprocessing

- Numericalize – handle out-of-vocabulary words

Source Batch (Train)

[Je, suis, etudiant, .]

[Quel, mois, ?, <pad>]

Target Batch (Train)

[<sos>, I, am, a, student, ., <eos>]

[<sos>, What, month, ?, <eos>, <pad>, <pad>]

Source Batch (Valid)

[Je, suis, <unk>, ., <pad>]

[<unk>, mois, <unk>, <unk>, ?]

Target Batch (Valid)

[<sos>, I, am, a, <unk>, ., <eos>]

[<sos>, <unk>, month, ?, <eos>, <pad>, <pad>]

Source Batch (Test)

[Je, suis, <unk>, .]

[Quel, <unk>, ?, <pad>]

NMT Basics – Data preprocessing

- Numericalize – numericalize

Source Batch (Train)

[4, 5, 6, 7]
[8, 9, 10, 2]

Target Batch (Train)

[0, 4, 5, 6, 7, 8, 1]
[0, 9, 10, 11, 1, 2, 2]

Source Batch (Valid)

[4, 5, 3, 7, 2]
[3, 9, 3, 3, 10]

Target Batch (Valid)

[0, 4, 5, 6, 3, 8, 1]
[0, 3, 10, 11, 1, 2, 2]

Source Batch (Test)

[4, 5, 3, 7]
[8, 3, 10, 2]

Encoder-Decoder Model - Training

Source Batch (Train)

4	5	6	7
8	9	10	2

Target Batch (Train)

0	4	5	6	7	8	1
0	9	10	11	1	2	2

Target Batch (Input)

0	4	5	6	7	8	1
0	9	10	11	1	2	2

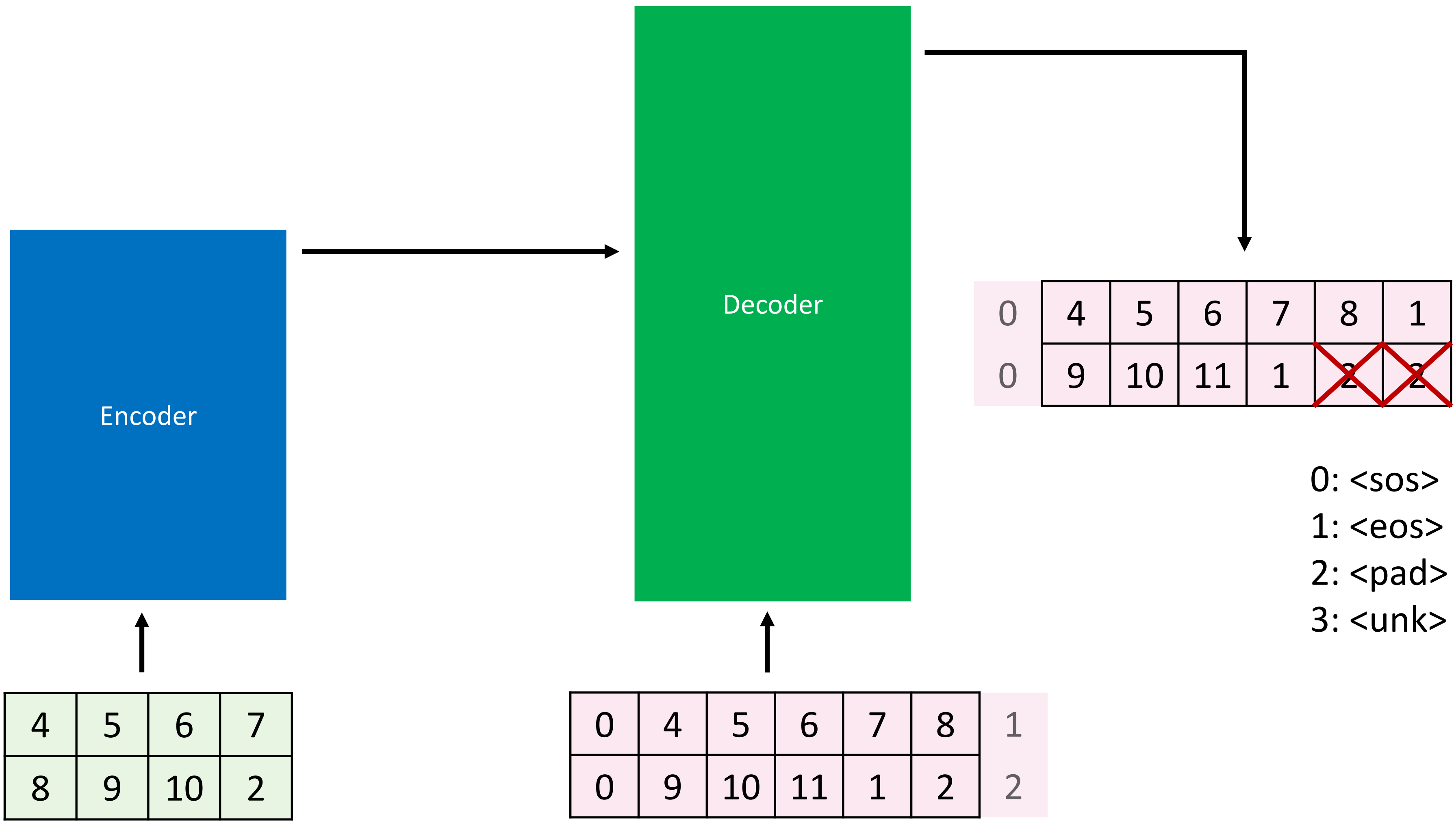
Target Batch (True Output)

0	4	5	6	7	8	1
0	9	10	11	1	2	2

Encoder

Decoder

Encoder-Decoder Model - Training



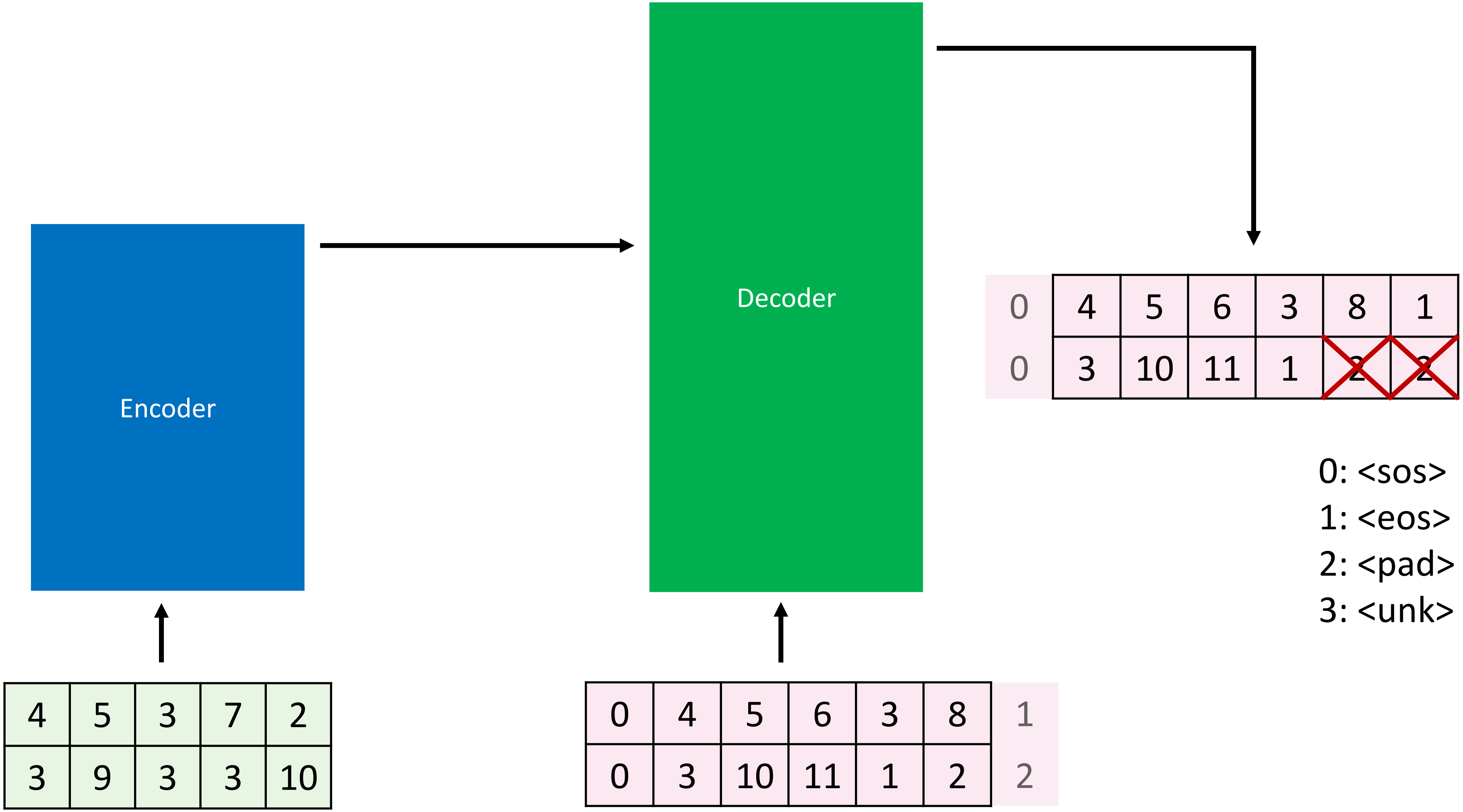
4	5	6	7
8	9	10	2

0	4	5	6	7	8	1
0	9	10	11	1	2	2

0	4	5	6	7	8	1
0	9	10	11	1	~~2~~	~~2~~

0: <sos>
1: <eos>
2: <pad>
3: <unk>

Encoder-Decoder Model - Validation



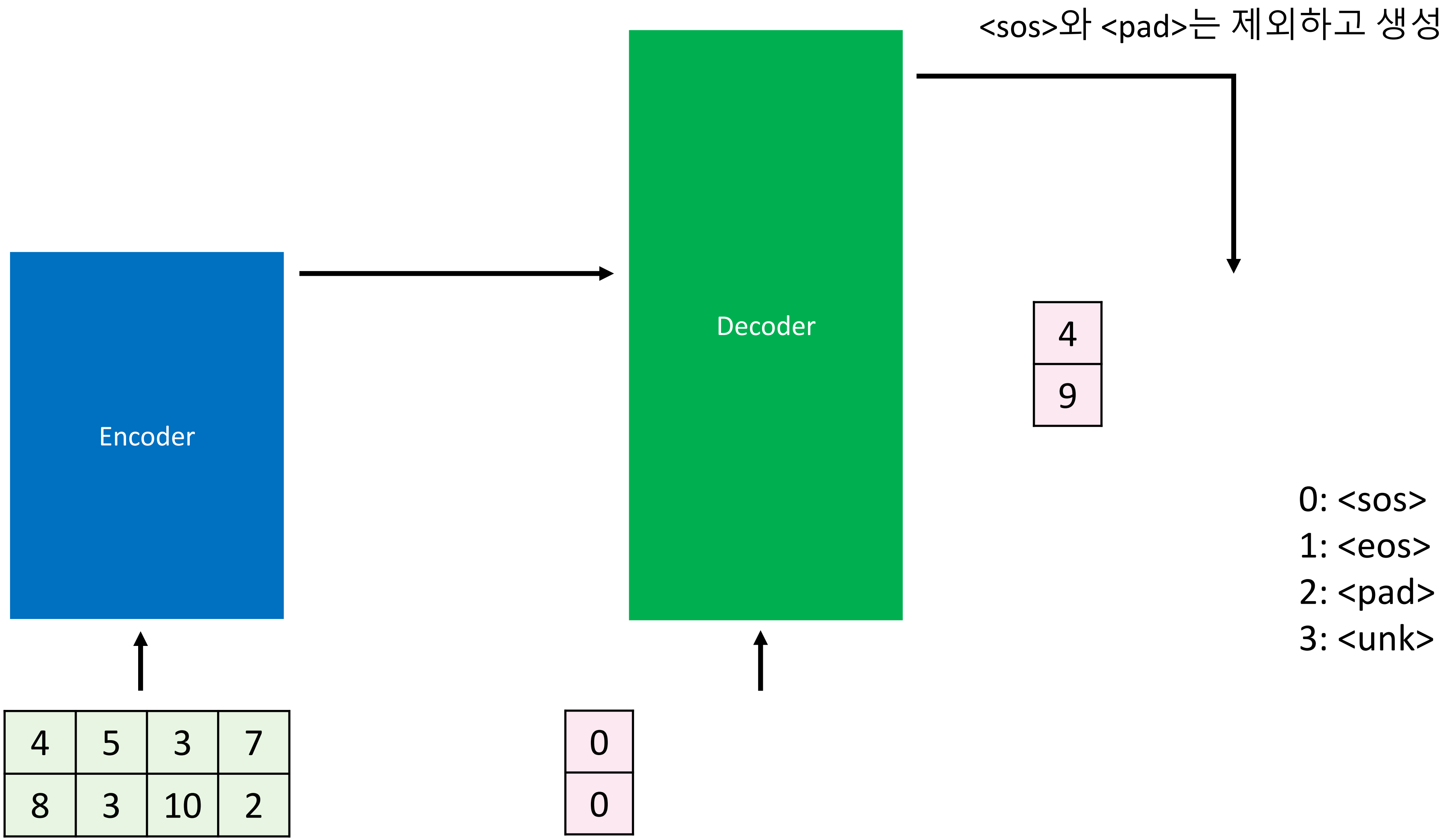
4	5	3	7	2
3	9	3	3	10

0	4	5	6	3	8	1
0	3	10	11	1	2	2

0	4	5	6	3	8	1
0	3	10	11	1	~~2~~	~~2~~

0: <sos>
1: <eos>
2: <pad>
3: <unk>

Encoder-Decoder Model – Test (Inference)



<sos>와 <pad>는 제외하고 생성

Encoder

Decoder

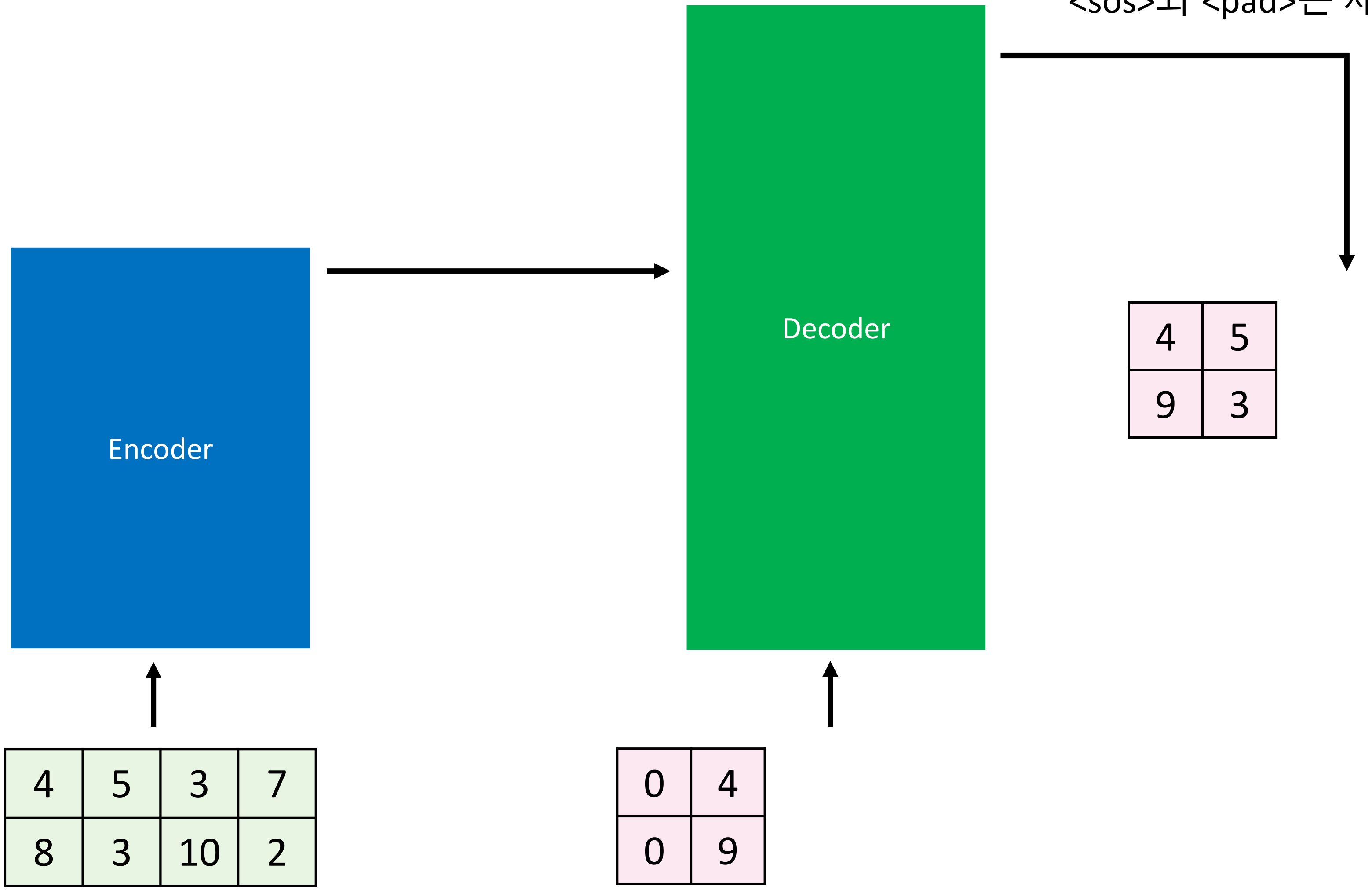
4	5	3	7
8	3	10	2

0
0

4
9

0: <sos>
1: <eos>
2: <pad>
3: <unk>

Encoder-Decoder Model – Test (Inference)



<sos>와 <pad>는 제외하고 생성

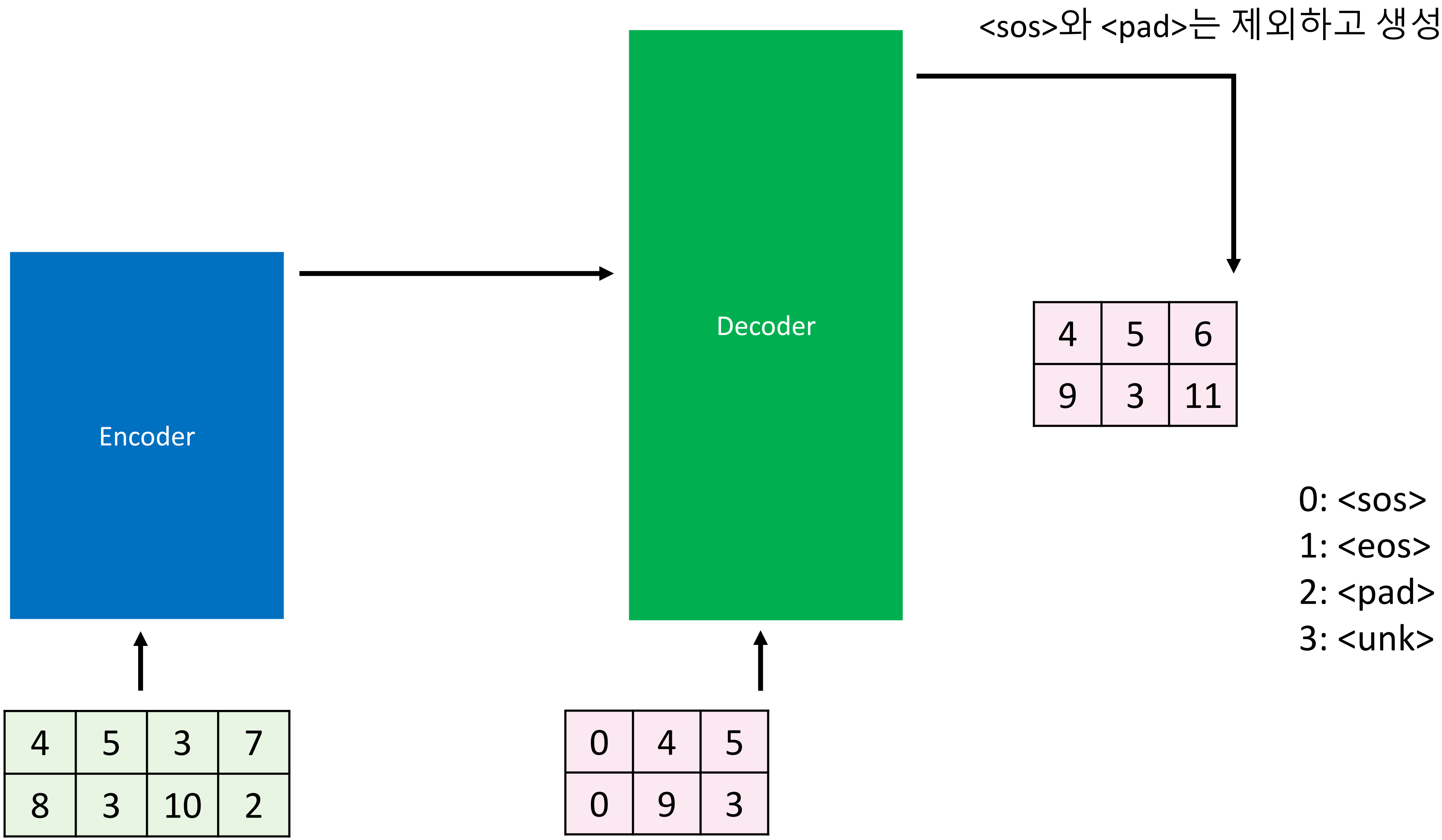
4	5	3	7
8	3	10	2

0	4
0	9

4	5
9	3

0:	<sos>
1:	<eos>
2:	<pad>
3:	<unk>

Encoder-Decoder Model – Test (Inference)



4	5	3	7
8	3	10	2

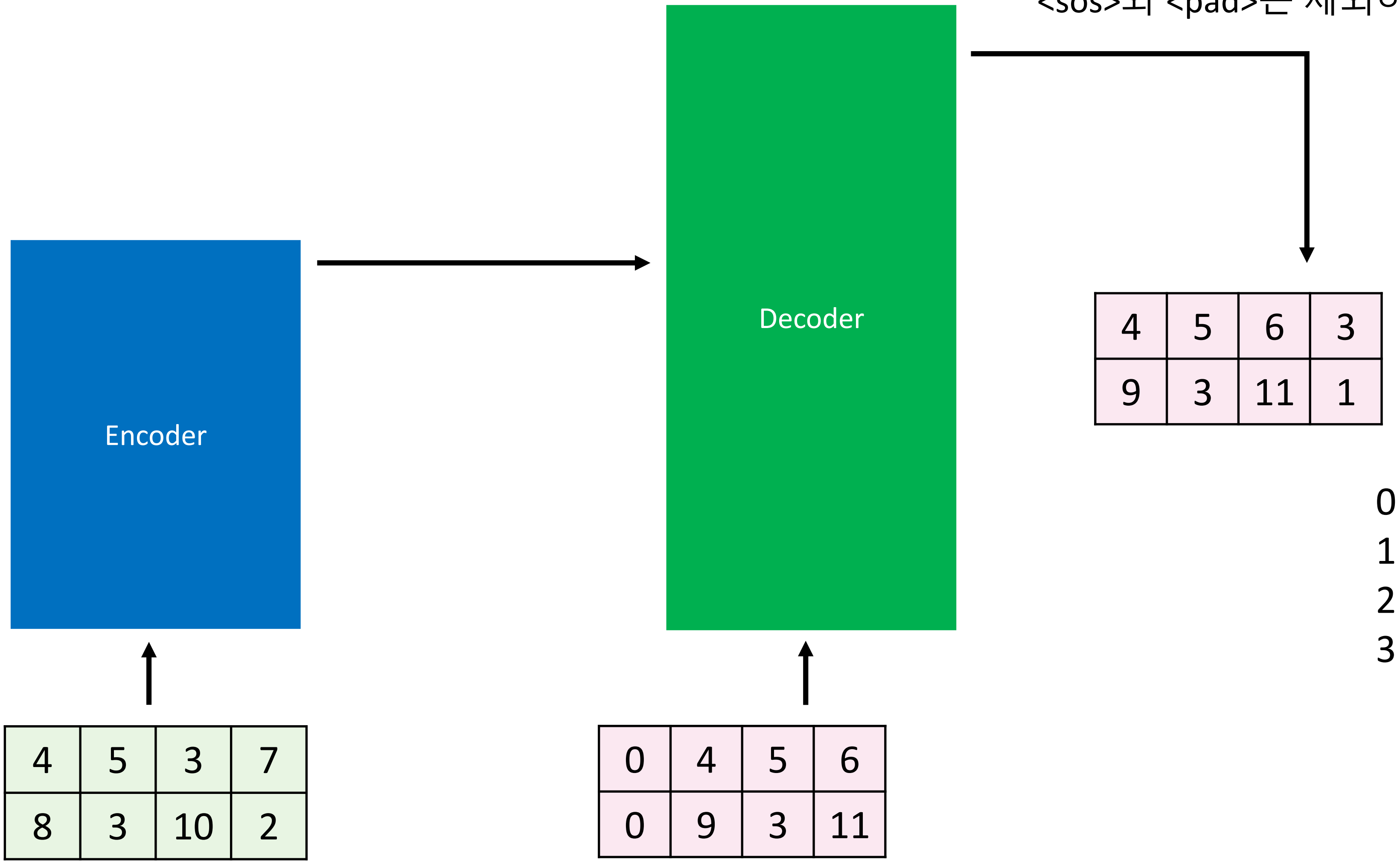
0	4	5
0	9	3

<sos>와 <pad>는 제외하고 생성

4	5	6
9	3	11

0: <sos>
1: <eos>
2: <pad>
3: <unk>

Encoder-Decoder Model – Test (Inference)



<sos>와 <pad>는 제외하고 생성

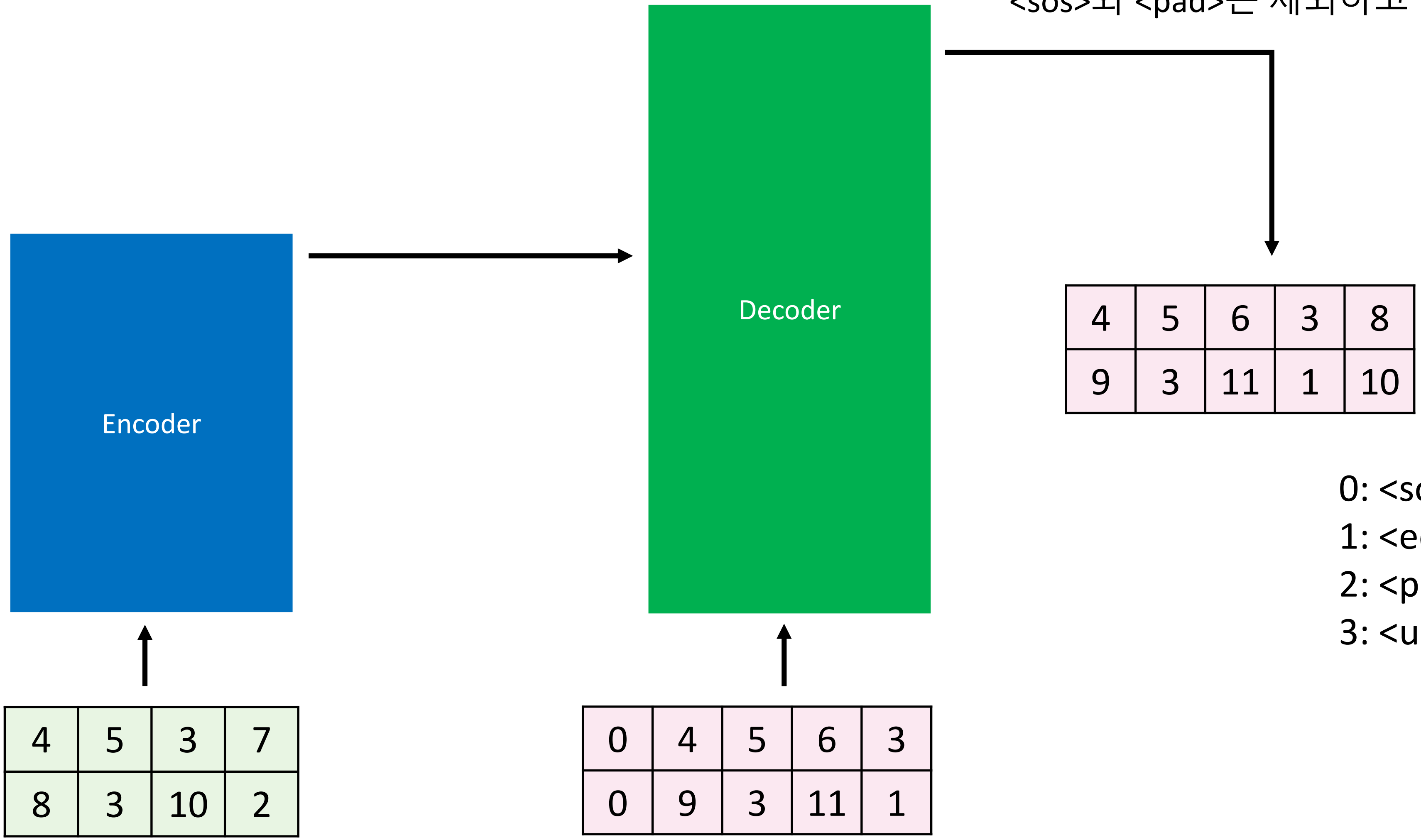
4	5	3	7
8	3	10	2

0	4	5	6
0	9	3	11

4	5	6	3
9	3	11	1

0: <sos>
1: <eos>
2: <pad>
3: <unk>

Encoder-Decoder Model – Test (Inference)



<sos>와 <pad>는 제외하고 생성

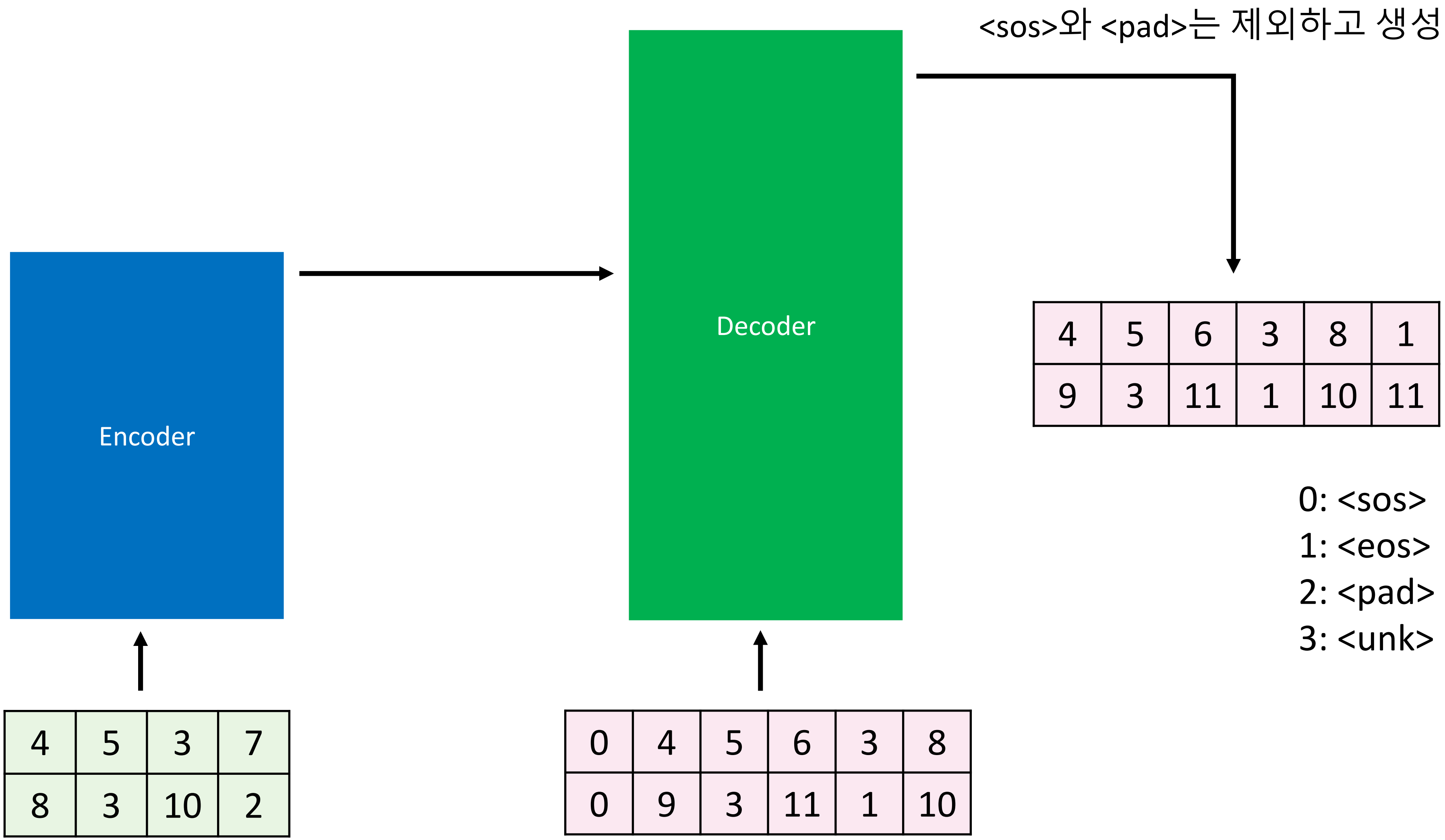
4	5	6	3	8
9	3	11	1	10

0: <sos>
1: <eos>
2: <pad>
3: <unk>

4	5	3	7
8	3	10	2

0	4	5	6	3
0	9	3	11	1

Encoder-Decoder Model – Test (Inference)



<sos>와 <pad>는 제외하고 생성

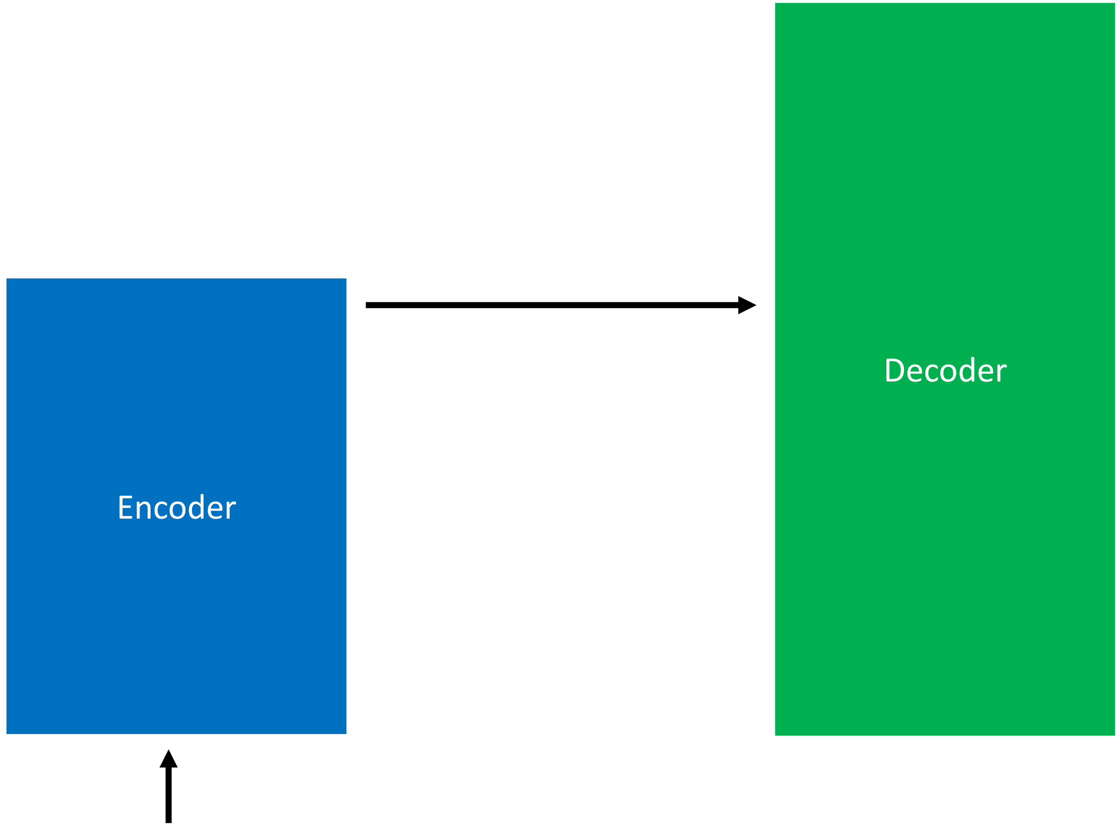
4	5	6	3	8	1
9	3	11	1	10	11

4	5	3	7
8	3	10	2

0	4	5	6	3	8
0	9	3	11	1	10

0: <sos>
1: <eos>
2: <pad>
3: <unk>

Encoder-Decoder Model – Test (Inference)



4	5	3	7
8	3	10	2

0	4	5	6	3	8	1
0	9	3	11	1	10	11

종료 조건

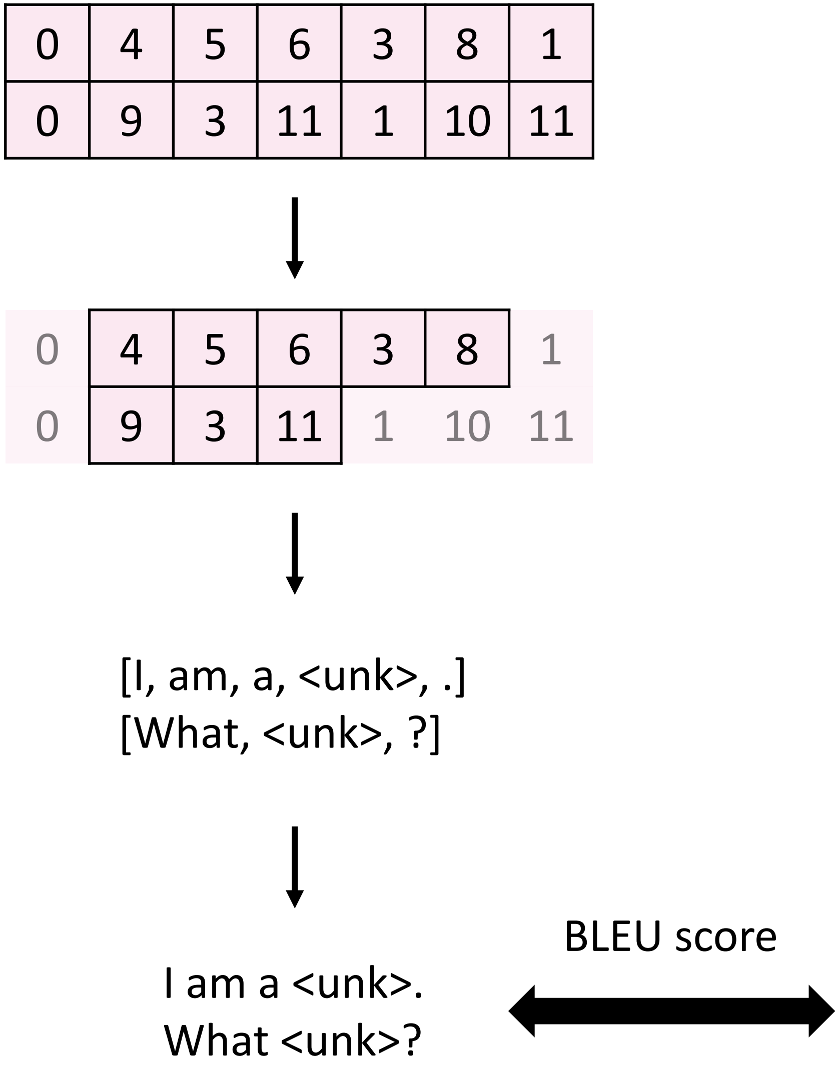
1. <eos>가 모든 문장에 나타났을 때.
2. 정해둔 max length에 도달했을 때.

0: <sos>
1: <eos>
2: <pad>
3: <unk>

NMT Basics – Compute BLEU Score

Target Vocab

0: <sos>
1: <eos>
2: <pad>
3: <unk>
4: I
5: am
6: a
7: student
8: .
9: What
10: month
11: ?
12: …..



0	4	5	6	3	8	1
0	9	3	11	1	10	11

0	4	5	6	3	8	1
0	9	3	11	1	10	11

[I, am, a, <unk>, .]
[What, <unk>, ?]

I am a <unk>.
What <unk>?

BLEU score

Target (Test)

I am a teacher.
What day?

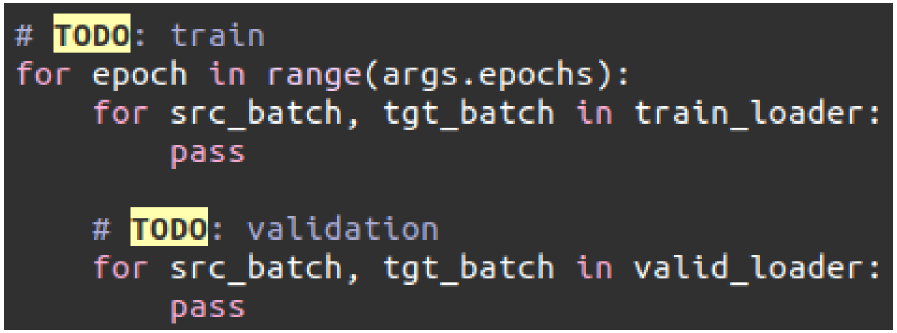
NMT Basics and Encoder-Decoder Model

- **NMT Basics**
  - Dataset 준비
  - Data preprocessing
    - Make batches
      - tokenize
      - add <sos>, <eos> token
      - padding
    - Numericalize
      - make vocabulary from train dataset
      - handle out-of-vocabulary words
      - numericalize
  - Compute BLEU Score
- **Encoder-Decoder Model**
  - Training
  - Validation
  - Test (Inference)

NMT Basics and Encoder-Decoder Model

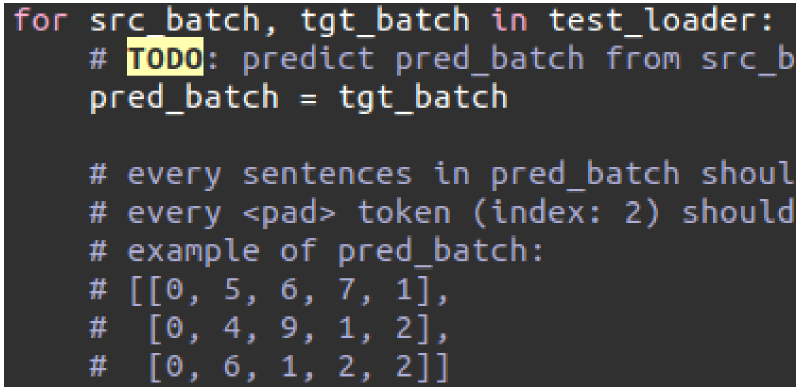
- **NMT Basics (Done!)**
  - Dataset 준비: Multi30k English to German Translation Dataset
  - Data preprocessing
    - Make batches
      - tokenize
      - add <sos>, <eos> token
      - padding
    - Numericalize
      - make vocabulary from train dataset
      - handle out-of-vocabulary words
      - numericalize
  - Compute BLEU Score
- **Encoder-Decoder Model**
  - Training
  - Validation
  - Test (Inference)

https://github.com/sehkmg/NMT_practice



```
# TODO: train
for epoch in range(args.epochs):
    for src_batch, tgt_batch in train_loader:
        pass

    # TODO: validation
    for src_batch, tgt_batch in valid_loader:
        pass
```



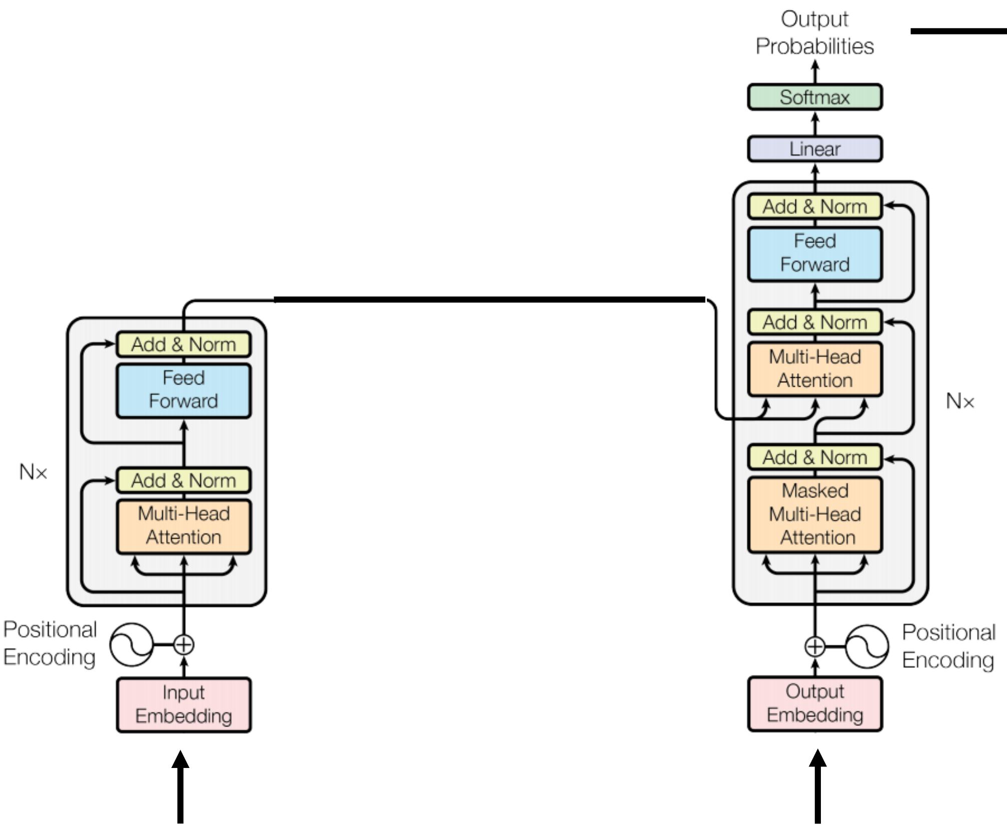
```
for src_batch, tgt_batch in test_loader:
    # TODO: predict pred_batch from src_b
    pred_batch = tgt_batch

    # every sentences in pred_batch shoul
    # every <pad> token (index: 2) should
    # example of pred_batch:
    # [[0, 5, 6, 7, 1],
    #  [0, 4, 9, 1, 2],
    #  [0, 6, 1, 2, 2]]
```

Table of Contents

- NMT Basics and Encoder-Decoder Model
- **Abstract View of Transformer Architecture**
- Dive into the Transformer Architecture

Abstract View of Transformer Architecture



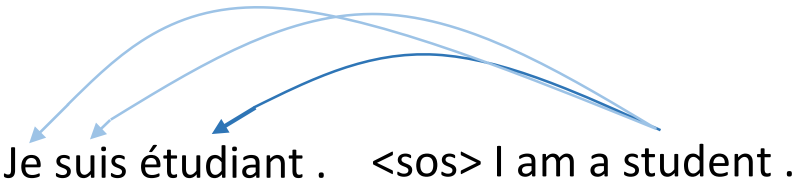
4	5	6	7
8	9	10	2

0	4	5	6	7	8
0	9	10	11	1	2

p_1^1	p_2^1	p_3^1	p_4^1	p_5^1	p_6^1
p_1^2	p_2^2	p_3^2	p_4^2	p_5^2	p_6^2

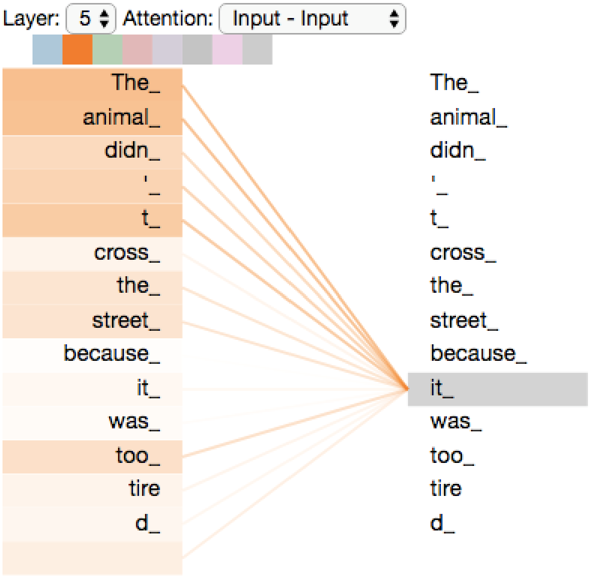
Self Attention

Original Attention:


Je suis étudiant . <sos> I am a student .

Q: target
K: source
V: source

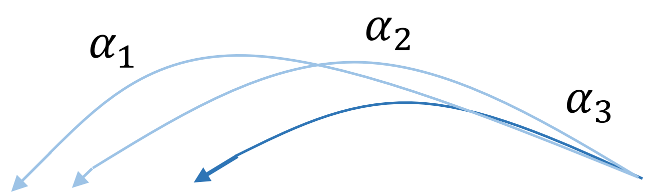
Self Attention:


Layer: 5 Attention: Input - Input

Q: source
K: source
V: source

Q: target
K: target
V: target

Multi-Head Attention

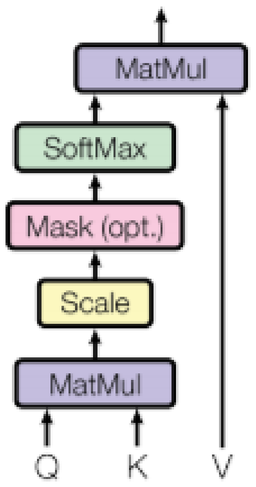


Original Attention: Je suis étudiant . <sos> I am a student .

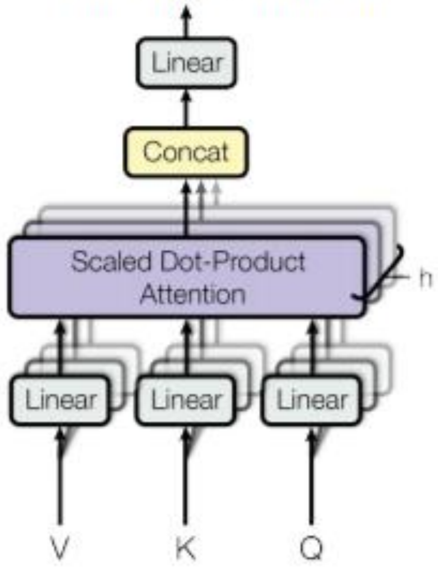
Q: target
K: source
V: source

Attention for student: α_1 × Je + α_2 × suis + α_3 × étudiant

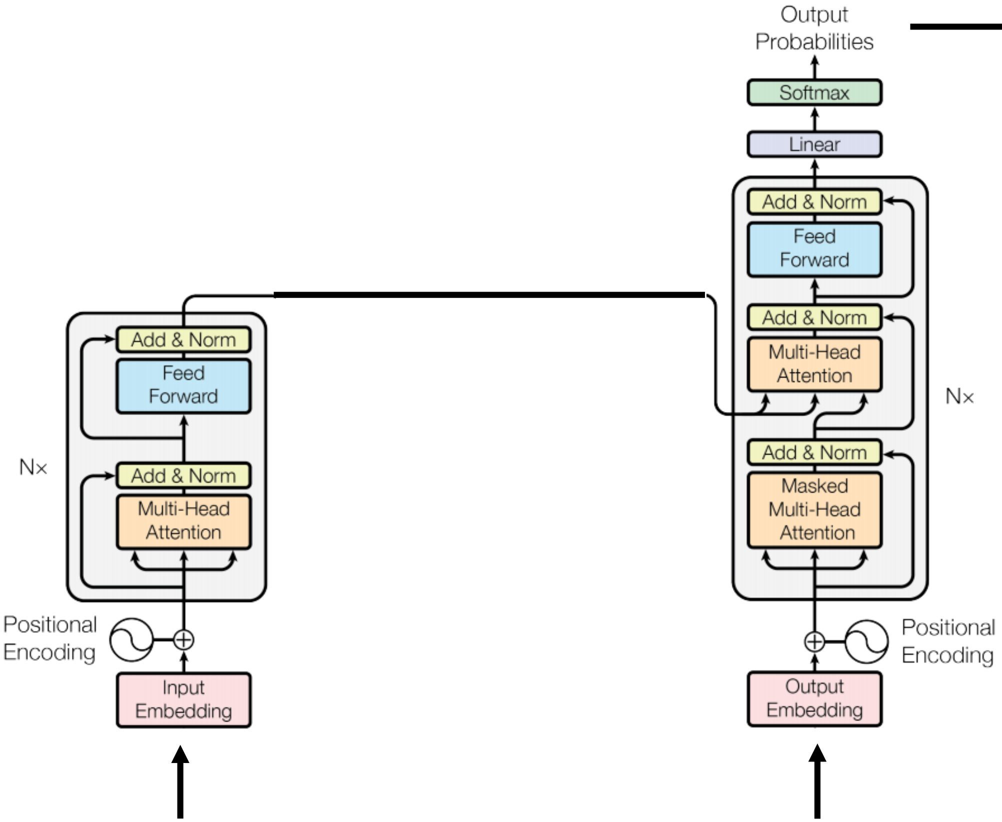
Scaled Dot-Product Attention



Multi-Head Attention



Abstract View of Transformer Architecture



4	5	6	7
8	9	10	2

0	4	5	6	7	8
0	9	10	11	1	2

p_1^1	p_2^1	p_3^1	p_4^1	p_5^1	p_6^1
p_1^2	p_2^2	p_3^2	p_4^2	p_5^2	p_6^2

Table of Contents

- NMT Basics and Encoder-Decoder Model

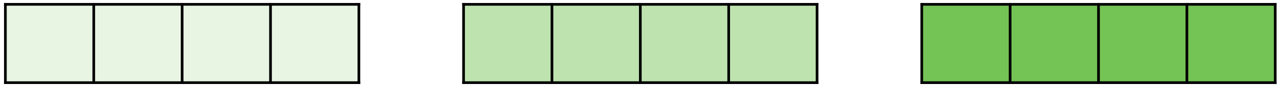
- Abstract View of Transformer Architecture

- **Dive into the Transformer Architecture**

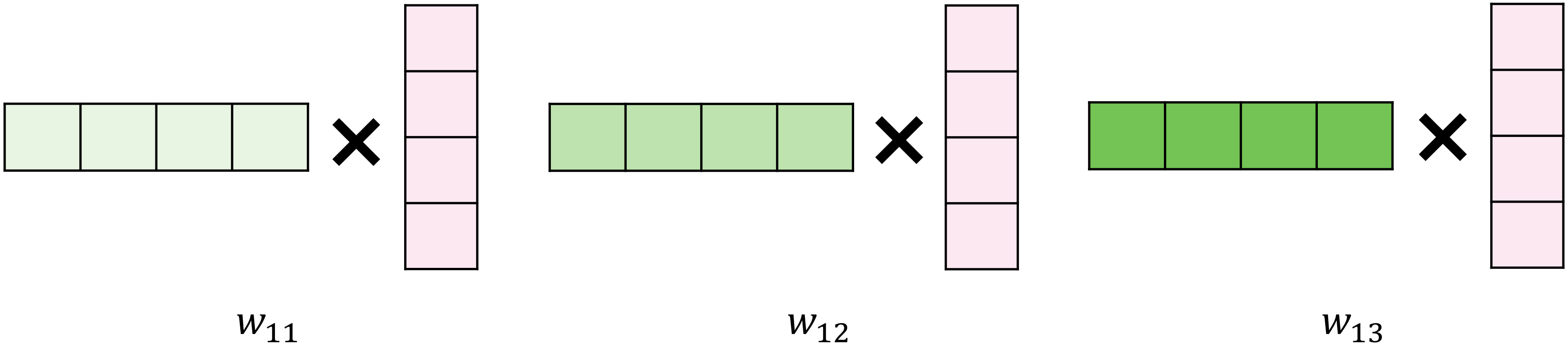
Attention is a Weighted Sum

- Query: 주인공 문장.

- Key, Value: 주인공 문장이 보는 문장.



- Query는 Key와 연산하여 weight를 구한다.



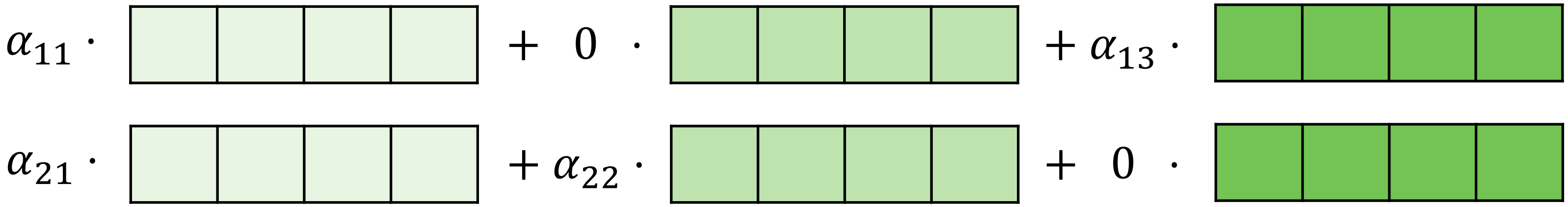
Attention is a Weighted Sum

- d_k로 나눠주고 원치 않는 정보를 masking을 통해 지운 후 Softmax를 취한다.

$$\text{Softmax}\left(\frac{w_{11}}{d_k}, \xcancel{\frac{w_{12}}{d_k}}, \frac{w_{13}}{d_k}\right) = (\alpha_{11}, 0, \alpha_{13})$$

$$\text{Softmax}\left(\frac{w_{21}}{d_k}, \frac{w_{22}}{d_k}, \xcancel{\frac{w_{23}}{d_k}}\right) = (\alpha_{21}, \alpha_{22}, 0)$$

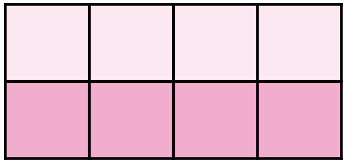
- Value를 대상으로 Weighted Sum을 한다.



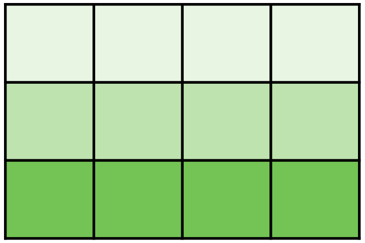
- 최종 output.

Attention in Matrix Form

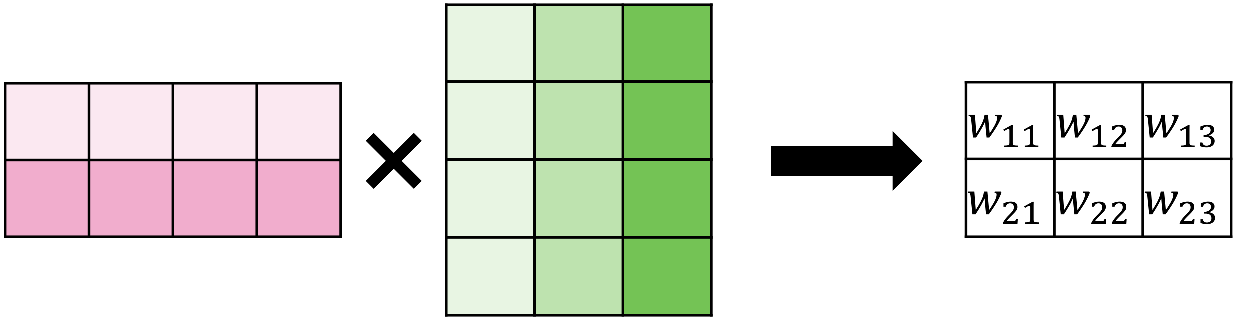
- Query: 주인공 문장.



- Key, Value: 주인공 문장이 보는 문장.



- Query는 Key와 연산하여 weight를 구한다.



$$\begin{array}{|c|c|c|}\hline w_{11} & w_{12} & w_{13} \\ \hline w_{21} & w_{22} & w_{23} \\ \hline \end{array}$$

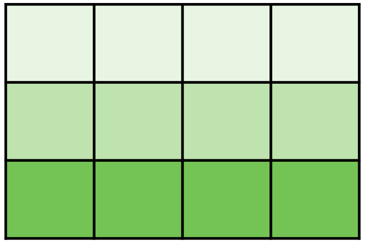
Attention in Matrix Form

- d_k로 나눠주고 원치 않는 정보를 masking을 통해 지운 후 Softmax를 취한다.

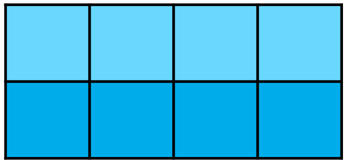
$$\text{Softmax}\left(\frac{\begin{array}{|c|c|c|}\hline w_{11} & \cancel{w_{12}} & w_{13} \\ \hline w_{21} & w_{22} & \cancel{w_{23}} \\ \hline\end{array}}{d_k}\right) \rightarrow \begin{array}{|c|c|c|}\hline \alpha_{11} & 0 & \alpha_{13} \\ \hline \alpha_{21} & \alpha_{22} & 0 \\ \hline\end{array}$$

- Value를 대상으로 Weighted Sum을 한다.

$$\begin{array}{|c|c|c|}\hline \alpha_{11} & 0 & \alpha_{13} \\ \hline \alpha_{21} & \alpha_{22} & 0 \\ \hline\end{array} \times$$

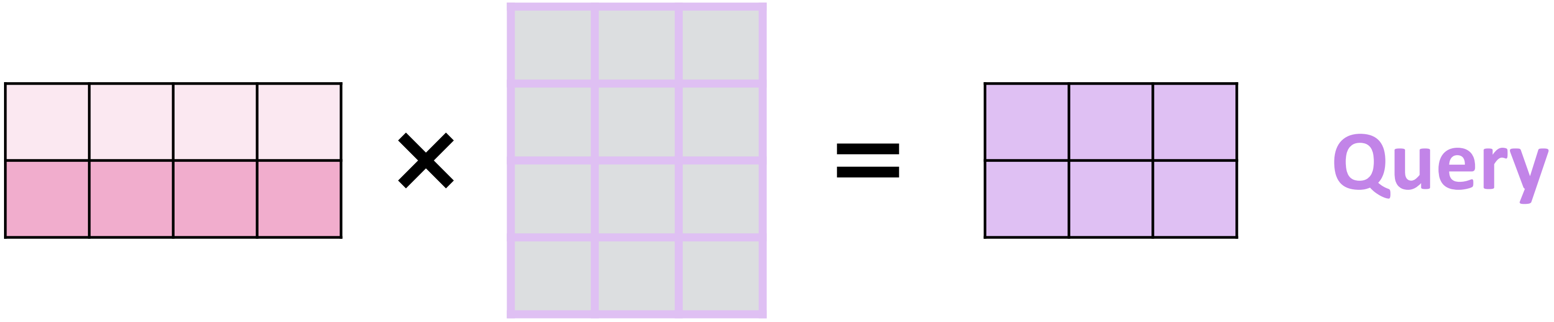


- 최종 output.

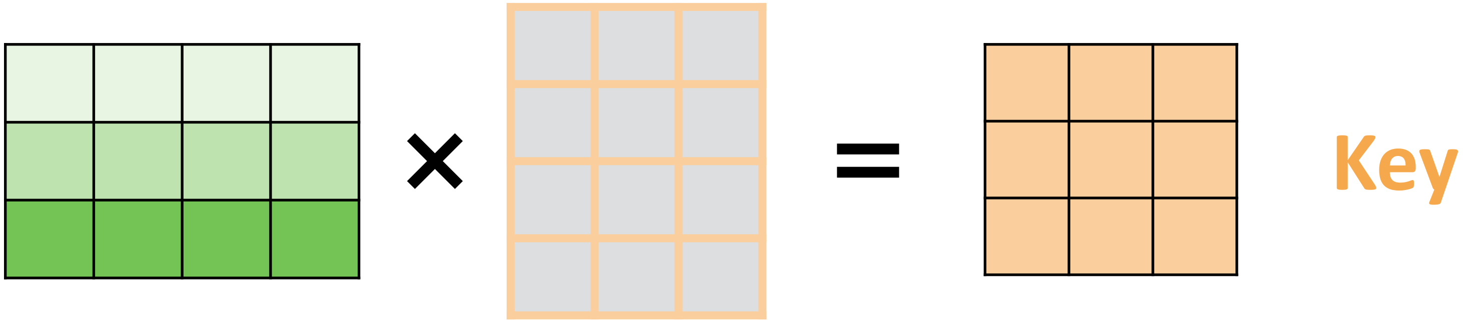


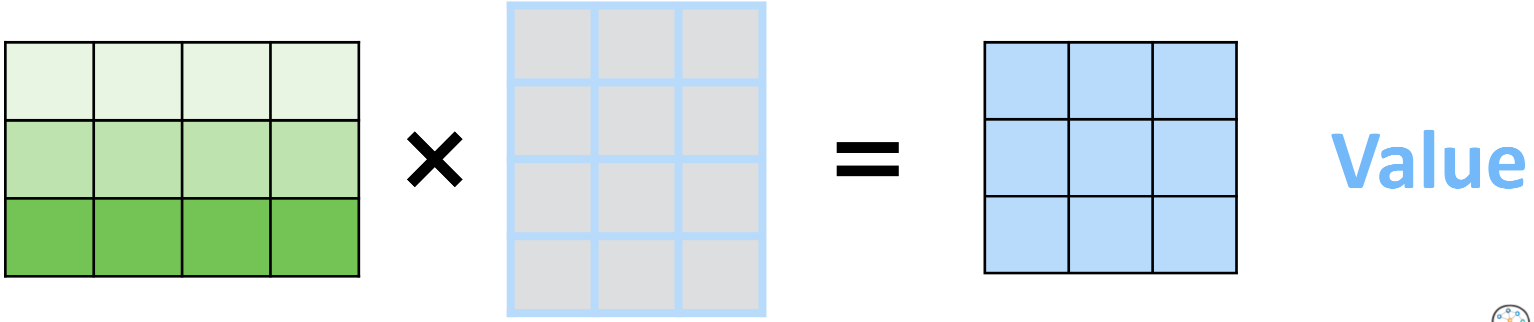
Attention in Implementation

- Query: 주인공 문장.



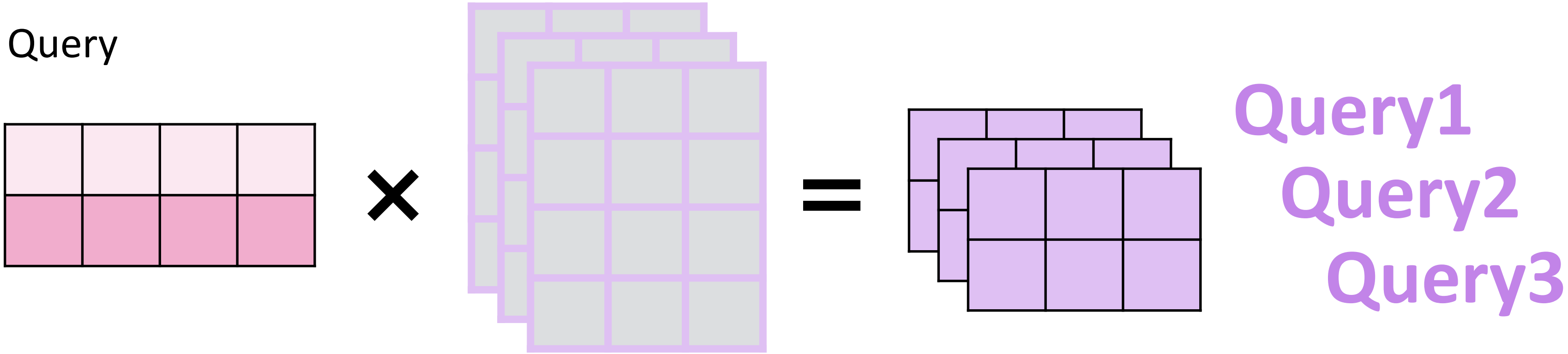
- Key, Value: 주인공 문장이 보는 문장.



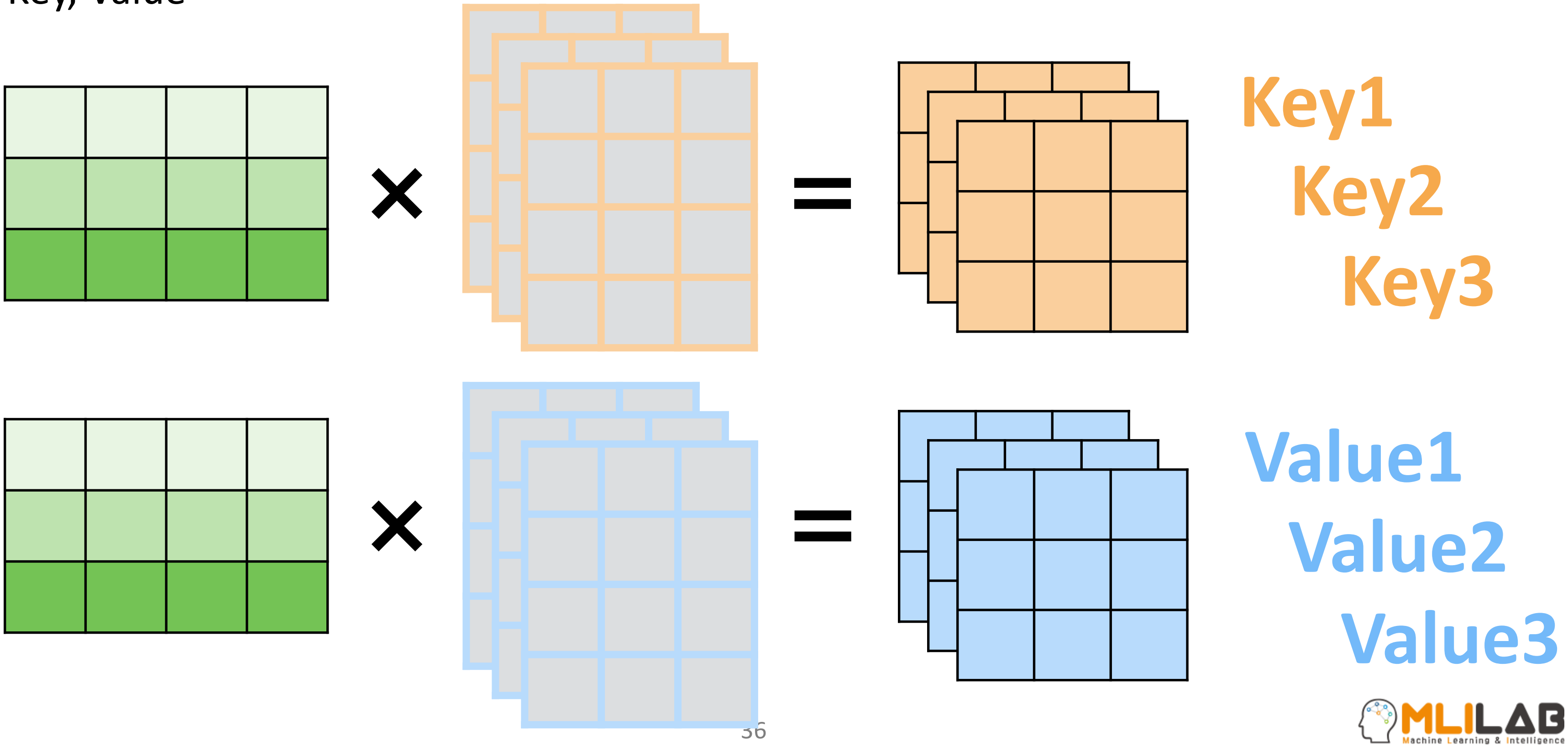


Multi-Head Attention

- Query

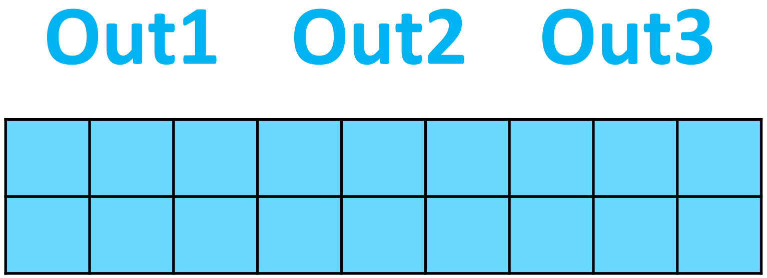


- Key, Value

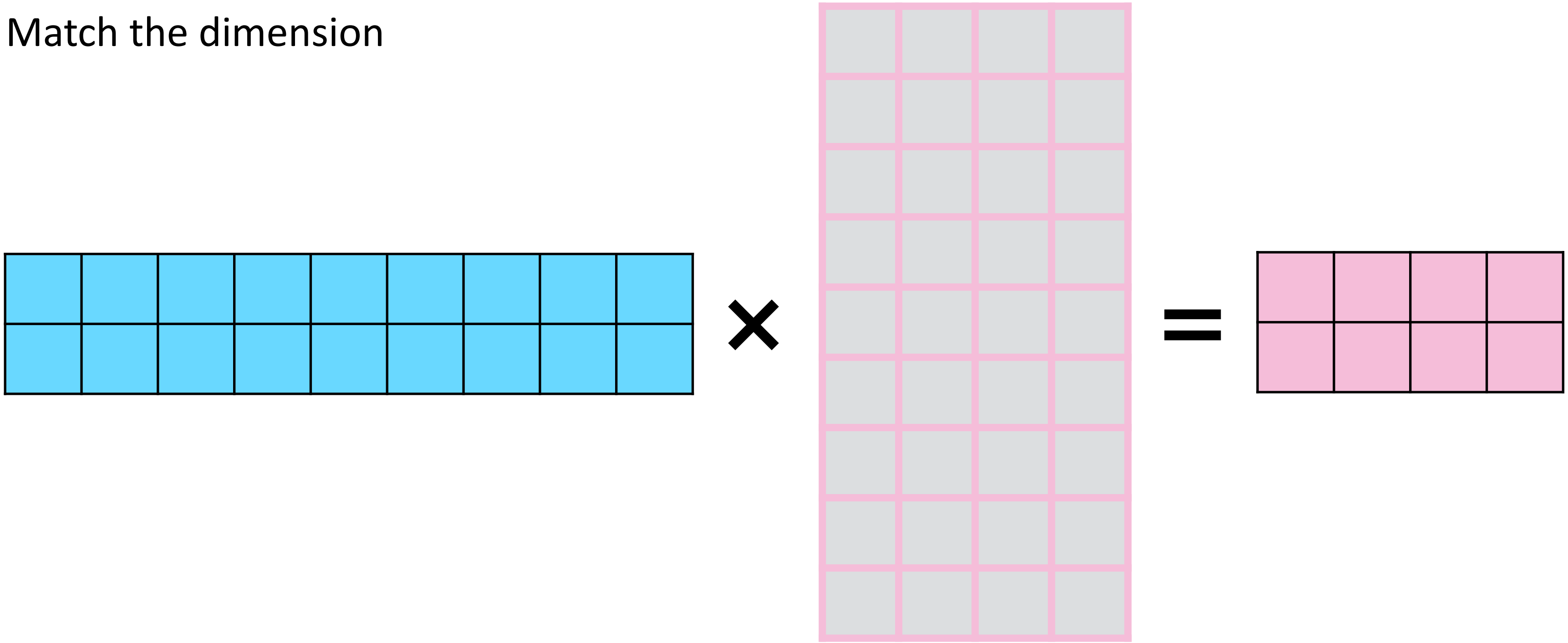


Multi-Head Attention

- Concatenate Outputs



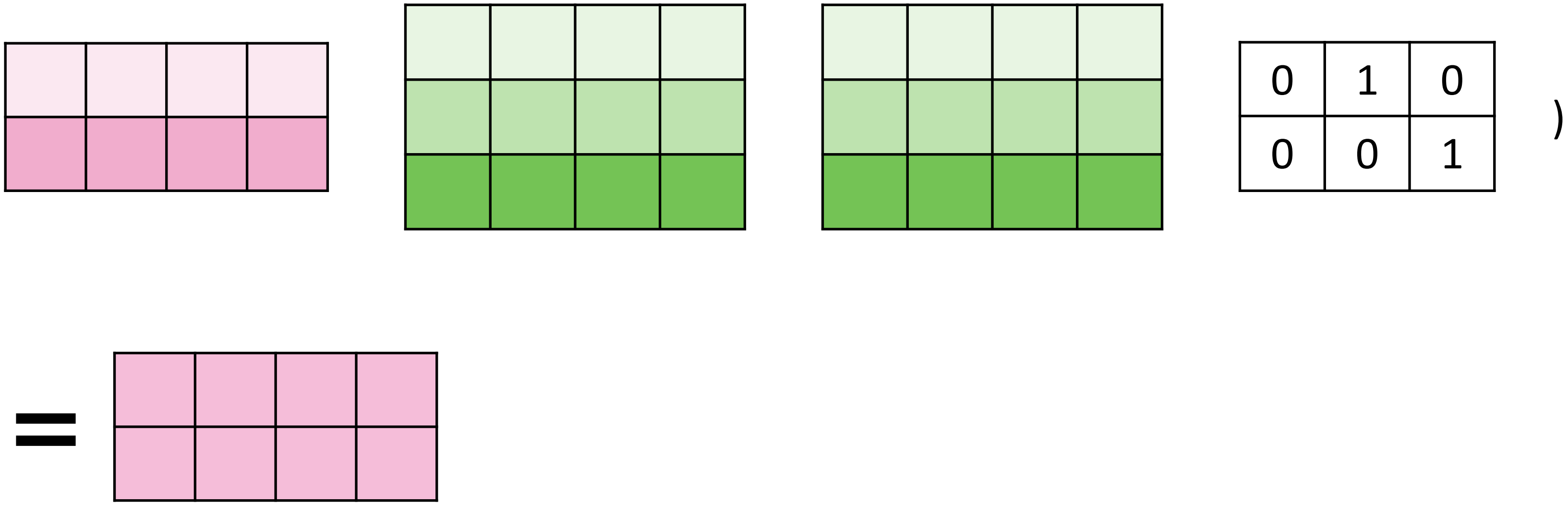
- Match the dimension



Multi-Head Attention

- Multi_Head_Attention(Query, Key, Value, Mask)

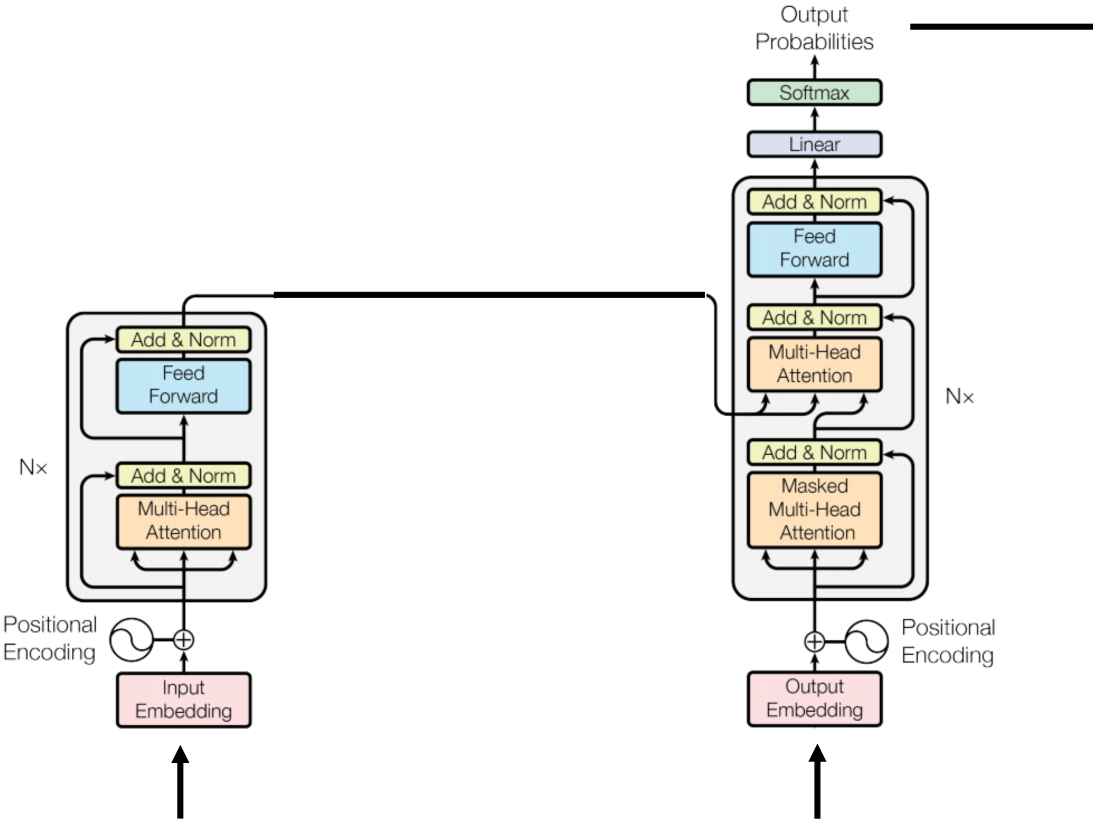
Multi_Head_Attention (



0	1	0
0	0	1

- 위 example에서는 단어 벡터의 차원이 4, Head가 3개, 각 Head의 차원이 3.
- 보통은 단어 벡터의 차원이 512, Head가 8개, 각 Head의 차원이 64.

Dive into the Transformer Architecture

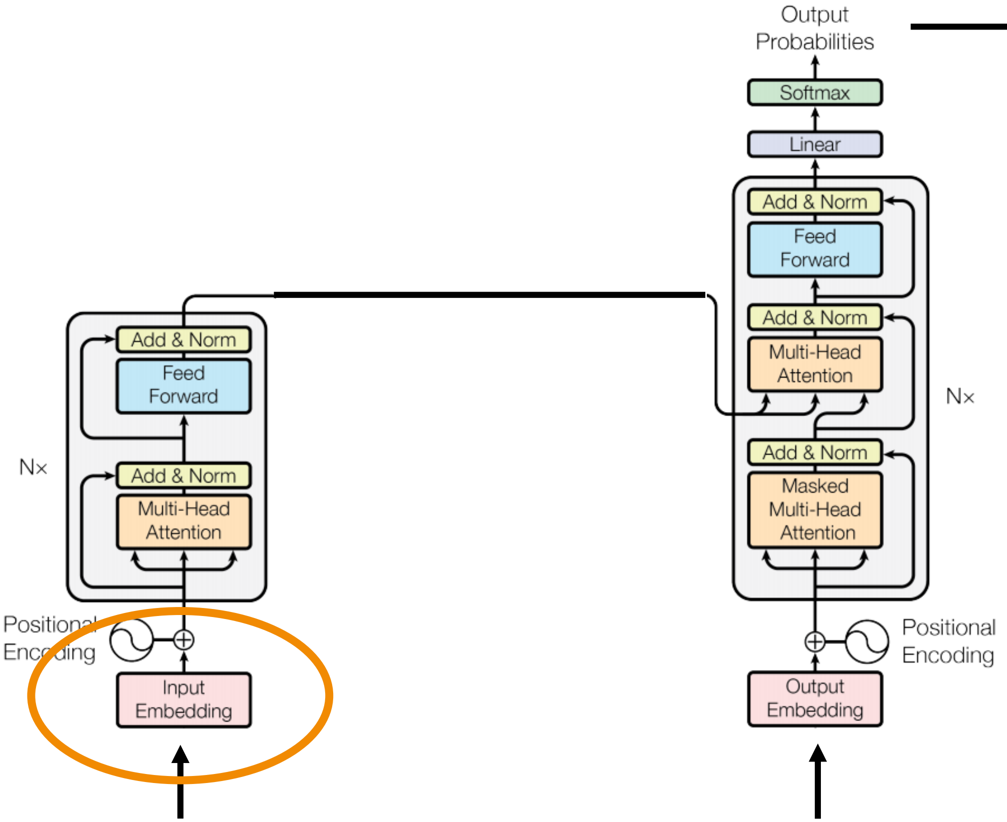


p_1^1	p_2^1	p_3^1	p_4^1	p_5^1	p_6^1
p_1^2	p_2^2	p_3^2	p_4^2	p_5^2	p_6^2

4	5	6	7
8	9	10	2

0	4	5	6	7	8
0	9	10	11	1	2

Input Embedding



4	5	6	7
8	9	10	2

0	4	5	6	7	8
0	9	10	11	1	2

p_1^1	p_2^1	p_3^1	p_4^1	p_5^1	p_6^1
p_1^2	p_2^2	p_3^2	p_4^2	p_5^2	p_6^2

Input Embedding

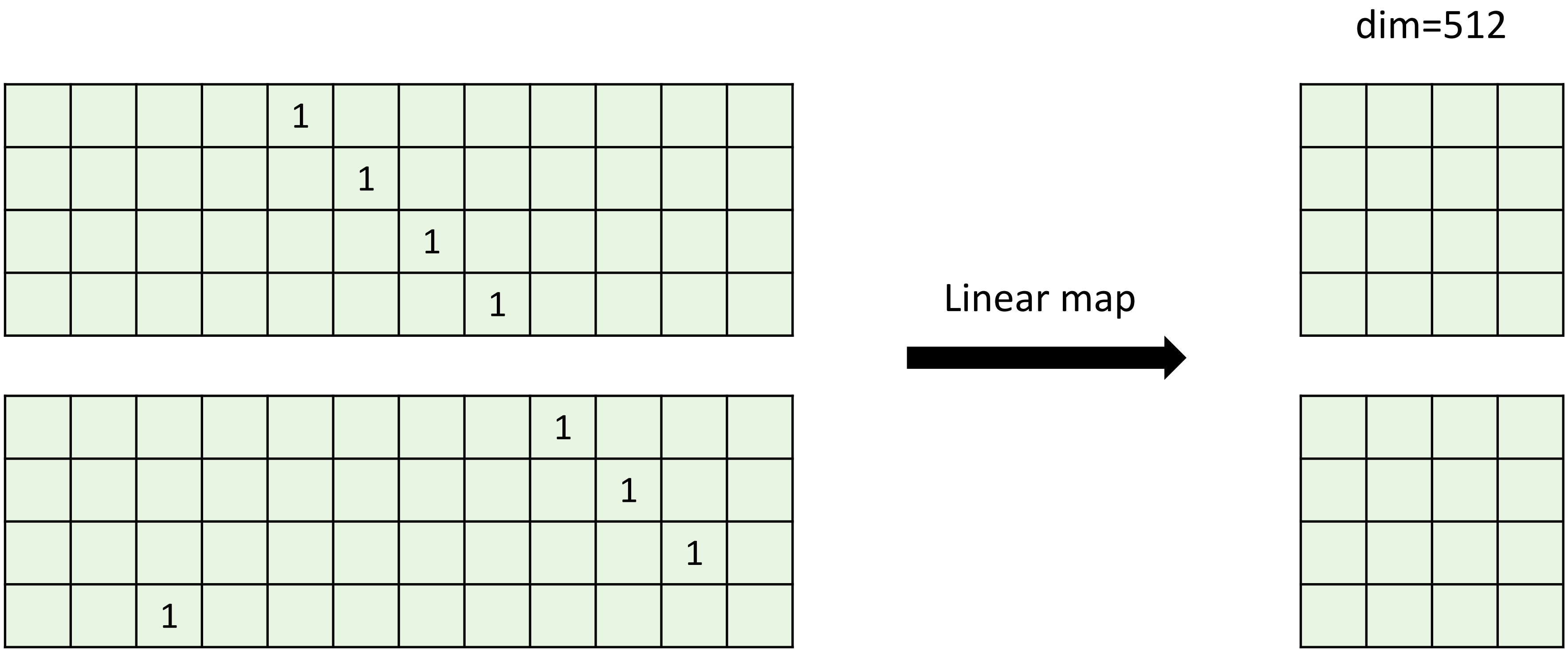
4	5	6	7
8	9	10	2

one-hot vector

				1							
					1						
						1					
							1				

								1			
									1		
										1	
		1									

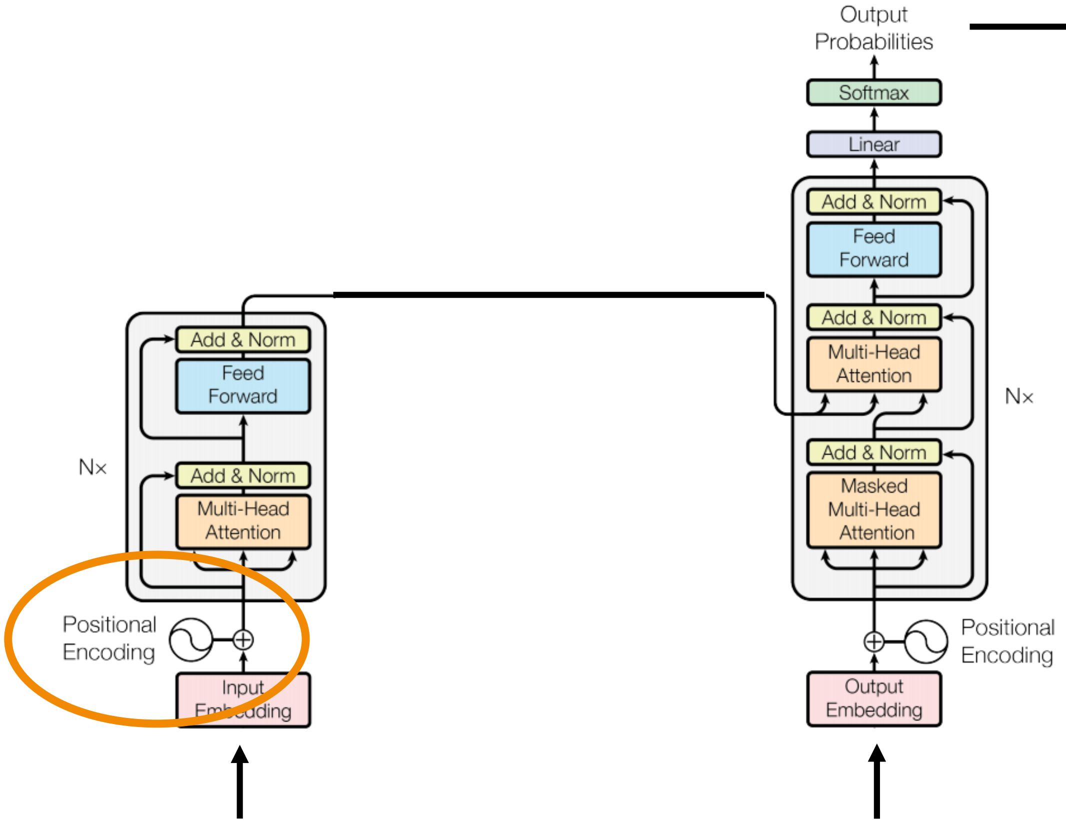
Input Embedding



				1							
					1						
						1					
							1				

								1			
									1		
										1	
		1									

Positional Encoding



p_1^1	p_2^1	p_3^1	p_4^1	p_5^1	p_6^1
p_1^2	p_2^2	p_3^2	p_4^2	p_5^2	p_6^2

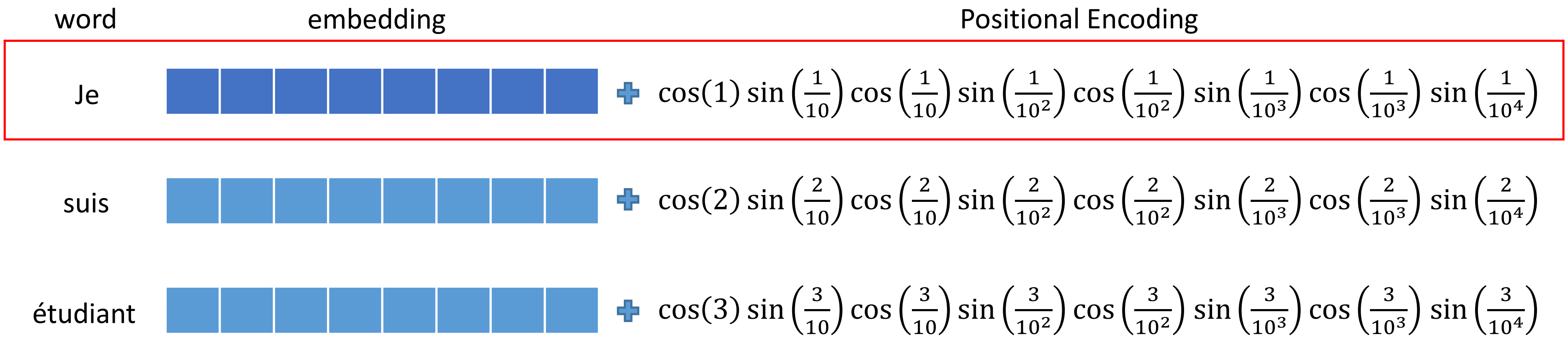
4	5	6	7
8	9	10	2

0	4	5	6	7	8
0	9	10	11	1	2

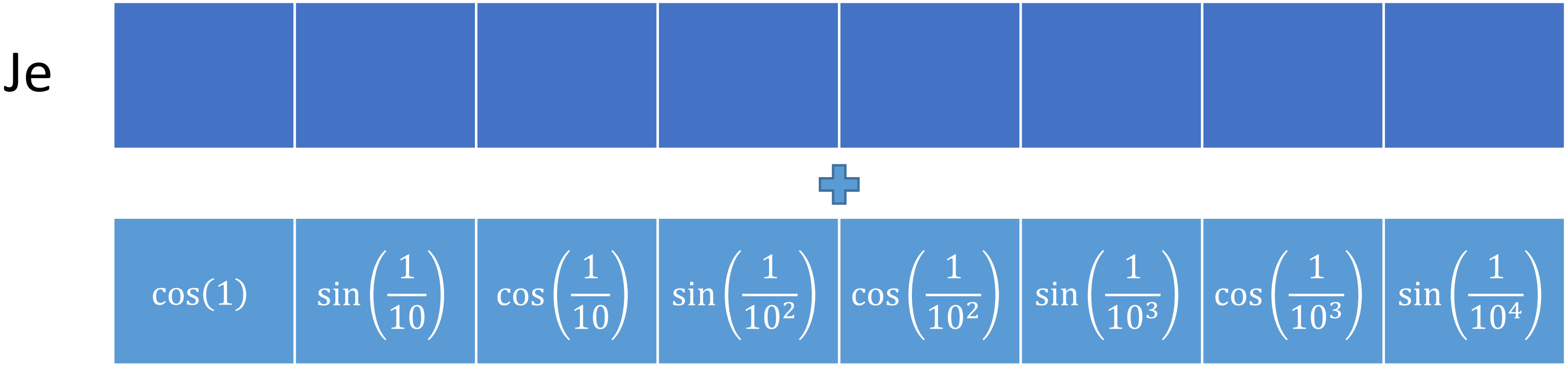
Positional Encoding

word	embedding		Positional Encoding
Je	🟦🟦🟦🟦🟦🟦🟦🟦	+	$\cos(1)\ \sin\left(\frac{1}{10}\right) \cos\left(\frac{1}{10}\right) \sin\left(\frac{1}{10^2}\right) \cos\left(\frac{1}{10^2}\right) \sin\left(\frac{1}{10^3}\right) \cos\left(\frac{1}{10^3}\right) \sin\left(\frac{1}{10^4}\right)$
suis	🟦🟦🟦🟦🟦🟦🟦🟦	+	$\cos(2)\ \sin\left(\frac{2}{10}\right) \cos\left(\frac{2}{10}\right) \sin\left(\frac{2}{10^2}\right) \cos\left(\frac{2}{10^2}\right) \sin\left(\frac{2}{10^3}\right) \cos\left(\frac{2}{10^3}\right) \sin\left(\frac{2}{10^4}\right)$
étudiant	🟦🟦🟦🟦🟦🟦🟦🟦	+	$\cos(3)\ \sin\left(\frac{3}{10}\right) \cos\left(\frac{3}{10}\right) \sin\left(\frac{3}{10^2}\right) \cos\left(\frac{3}{10^2}\right) \sin\left(\frac{3}{10^3}\right) \cos\left(\frac{3}{10^3}\right) \sin\left(\frac{3}{10^4}\right)$

Positional Encoding



word	embedding		Positional Encoding
Je		+	$\cos(1)\ \sin\left(\frac{1}{10}\right)\ \cos\left(\frac{1}{10}\right)\ \sin\left(\frac{1}{10^2}\right)\ \cos\left(\frac{1}{10^2}\right)\ \sin\left(\frac{1}{10^3}\right)\ \cos\left(\frac{1}{10^3}\right)\ \sin\left(\frac{1}{10^4}\right)$
suis		+	$\cos(2)\ \sin\left(\frac{2}{10}\right)\ \cos\left(\frac{2}{10}\right)\ \sin\left(\frac{2}{10^2}\right)\ \cos\left(\frac{2}{10^2}\right)\ \sin\left(\frac{2}{10^3}\right)\ \cos\left(\frac{2}{10^3}\right)\ \sin\left(\frac{2}{10^4}\right)$
étudiant		+	$\cos(3)\ \sin\left(\frac{3}{10}\right)\ \cos\left(\frac{3}{10}\right)\ \sin\left(\frac{3}{10^2}\right)\ \cos\left(\frac{3}{10^2}\right)\ \sin\left(\frac{3}{10^3}\right)\ \cos\left(\frac{3}{10^3}\right)\ \sin\left(\frac{3}{10^4}\right)$



Je

+

$\cos(1)$	$\sin\left(\frac{1}{10}\right)$	$\cos\left(\frac{1}{10}\right)$	$\sin\left(\frac{1}{10^2}\right)$	$\cos\left(\frac{1}{10^2}\right)$	$\sin\left(\frac{1}{10^3}\right)$	$\cos\left(\frac{1}{10^3}\right)$	$\sin\left(\frac{1}{10^4}\right)$

Positional Encoding

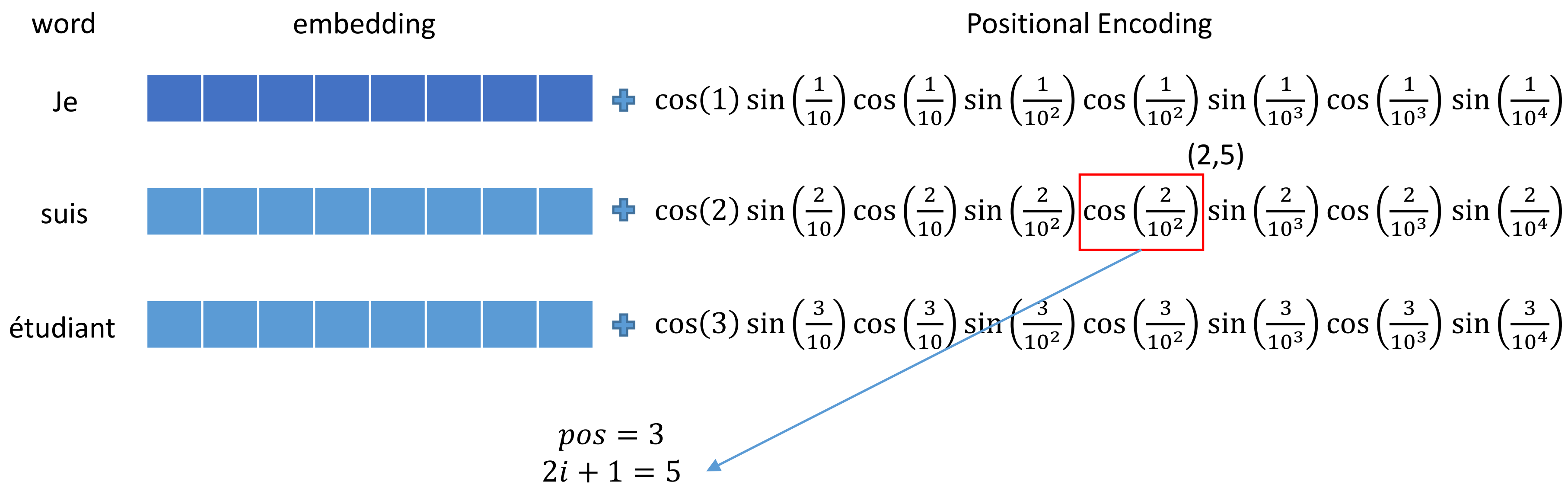
word	embedding	Positional Encoding
Je		+ $\cos(1)\ \sin\left(\frac{1}{10}\right)\ \cos\left(\frac{1}{10}\right)\ \sin\left(\frac{1}{10^2}\right)\ \cos\left(\frac{1}{10^2}\right)\ \sin\left(\frac{1}{10^3}\right)\ \cos\left(\frac{1}{10^3}\right)\ \sin\left(\frac{1}{10^4}\right)$
suis		+ $\cos(2)\ \sin\left(\frac{2}{10}\right)\ \cos\left(\frac{2}{10}\right)\ \sin\left(\frac{2}{10^2}\right)\ \cos\left(\frac{2}{10^2}\right)\ \sin\left(\frac{2}{10^3}\right)\ \cos\left(\frac{2}{10^3}\right)\ \sin\left(\frac{2}{10^4}\right)$
étudiant		+ $\cos(3)\ \boxed{\sin\left(\frac{3}{10}\right)}\ \cos\left(\frac{3}{10}\right)\ \sin\left(\frac{3}{10^2}\right)\ \cos\left(\frac{3}{10^2}\right)\ \sin\left(\frac{3}{10^3}\right)\ \cos\left(\frac{3}{10^3}\right)\ \sin\left(\frac{3}{10^4}\right)$

(3,2)

$$pos = 3$$
$$2i = 2$$
$$d_{model} = 8$$

$$PE_{pos,2i} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \Rightarrow PE_{3,2} = \sin\left(\frac{3}{10000^{\frac{2}{8}}}\right) = \sin\left(\frac{3}{10}\right)$$

Positional Encoding



word | embedding | Positional Encoding

Je + $\cos(1)\ \sin\left(\frac{1}{10}\right)\ \cos\left(\frac{1}{10}\right)\ \sin\left(\frac{1}{10^2}\right)\ \cos\left(\frac{1}{10^2}\right)\ \sin\left(\frac{1}{10^3}\right)\ \cos\left(\frac{1}{10^3}\right)\ \sin\left(\frac{1}{10^4}\right)$

(2,5)

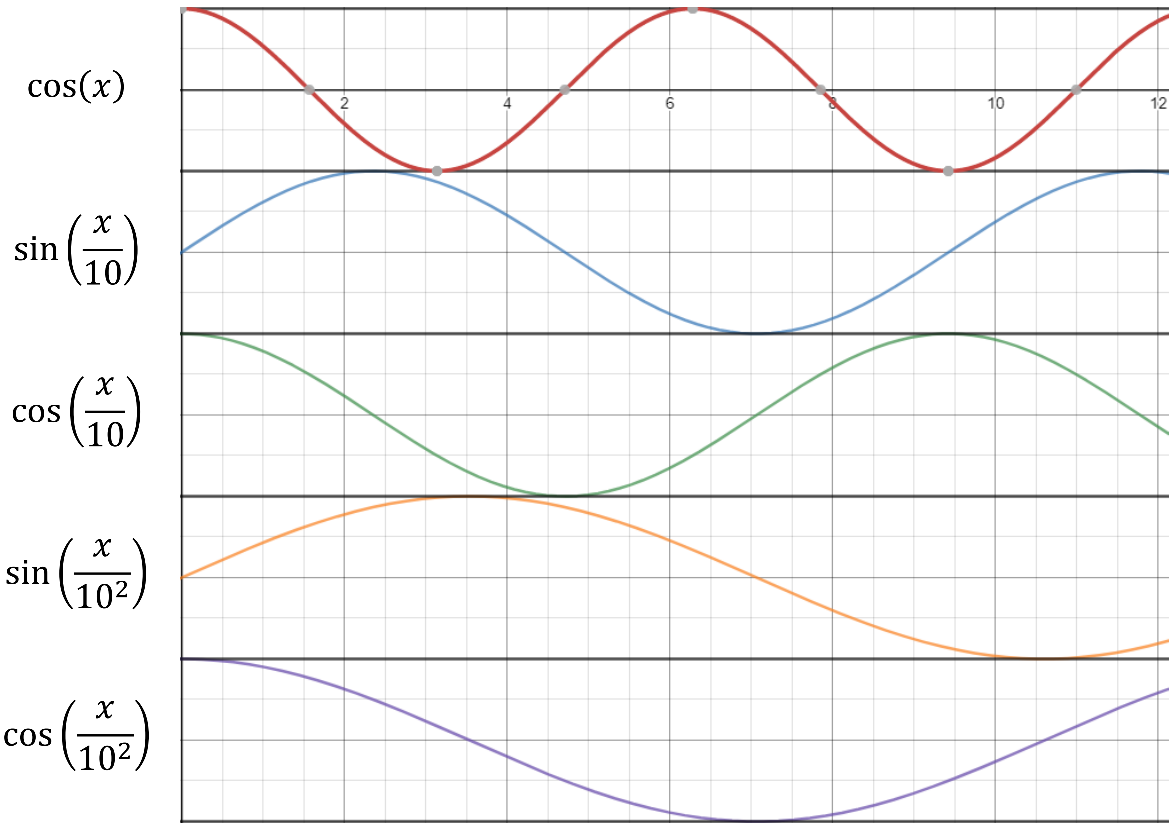
suis + $\cos(2)\ \sin\left(\frac{2}{10}\right)\ \cos\left(\frac{2}{10}\right)\ \sin\left(\frac{2}{10^2}\right)\ \boxed{\cos\left(\frac{2}{10^2}\right)}\ \sin\left(\frac{2}{10^3}\right)\ \cos\left(\frac{2}{10^3}\right)\ \sin\left(\frac{2}{10^4}\right)$

étudiant + $\cos(3)\ \sin\left(\frac{3}{10}\right)\ \cos\left(\frac{3}{10}\right)\ \sin\left(\frac{3}{10^2}\right)\ \cos\left(\frac{3}{10^2}\right)\ \sin\left(\frac{3}{10^3}\right)\ \cos\left(\frac{3}{10^3}\right)\ \sin\left(\frac{3}{10^4}\right)$

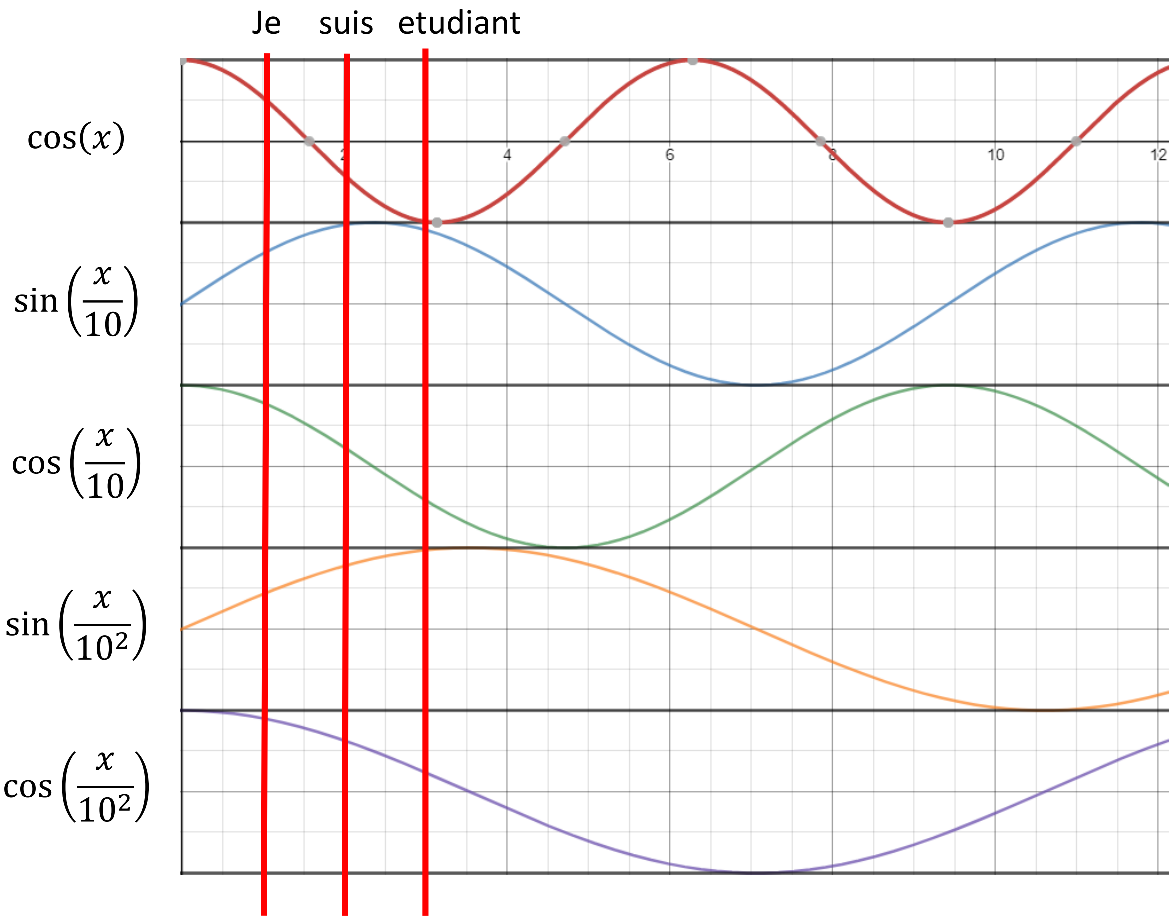
$$pos = 3$$
$$2i + 1 = 5$$
$$d_{model} = 8$$

$$PE_{pos,2i+1} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \Rightarrow PE_{2,5} = \cos\left(\frac{3}{10000^{\frac{4}{8}}}\right) = \cos\left(\frac{2}{10^2}\right)$$

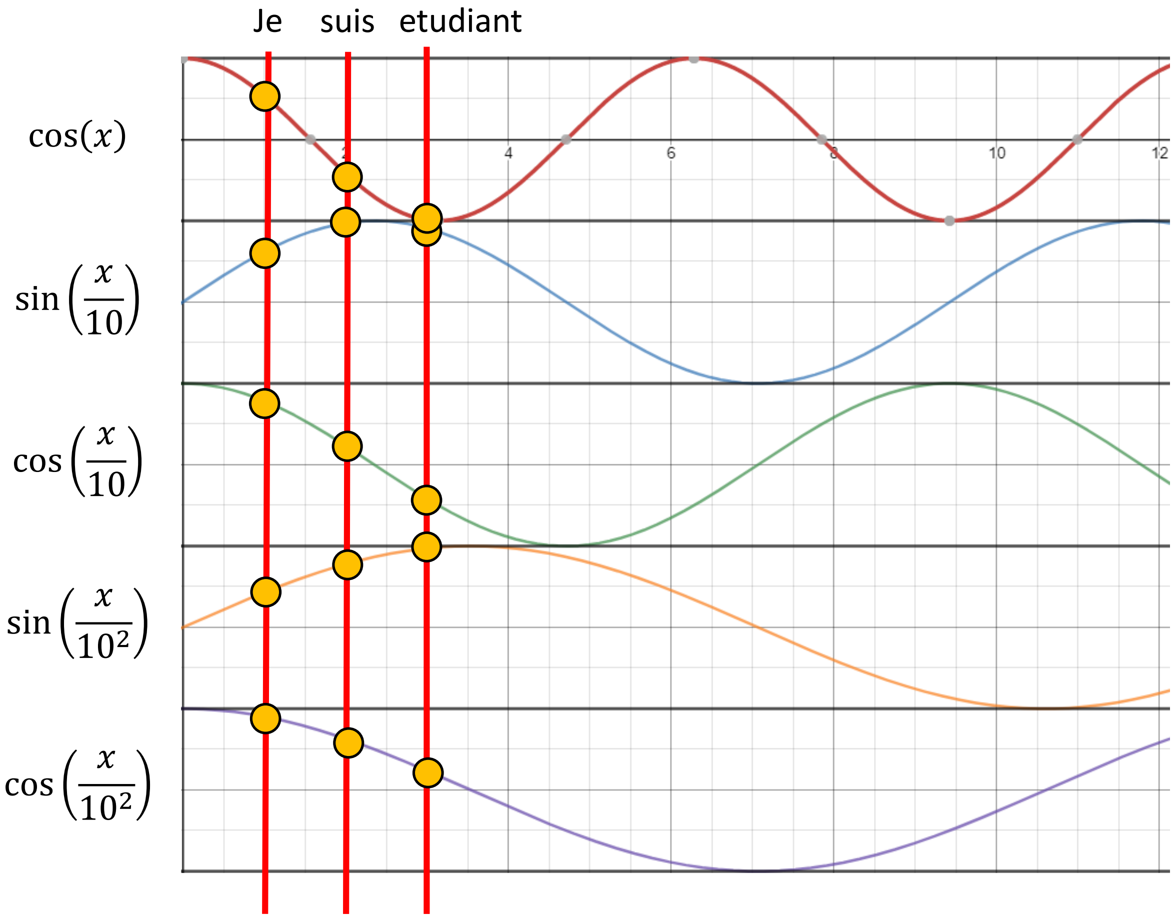
Positional Encoding



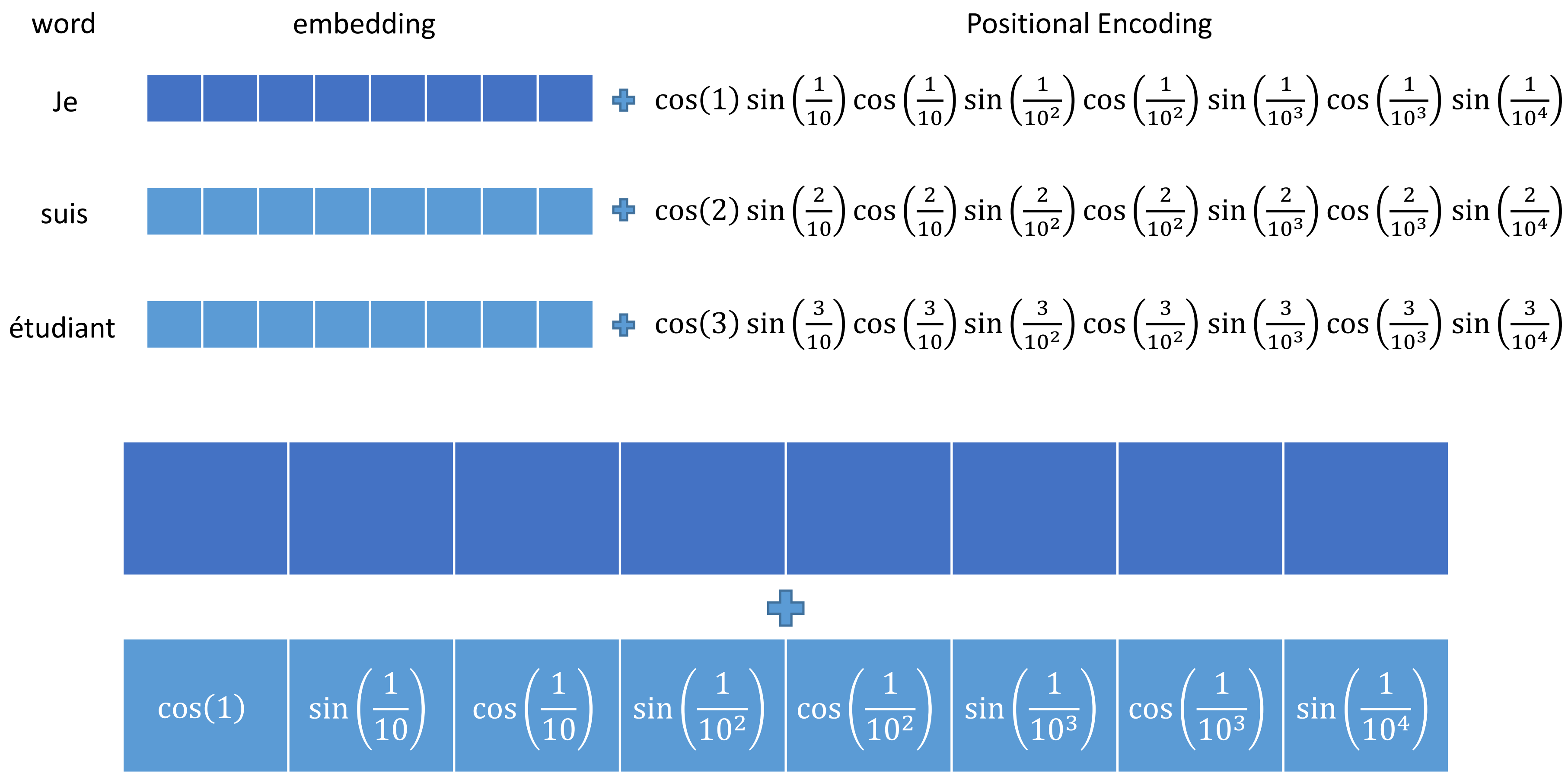
Positional Encoding



Positional Encoding



Positional Encoding

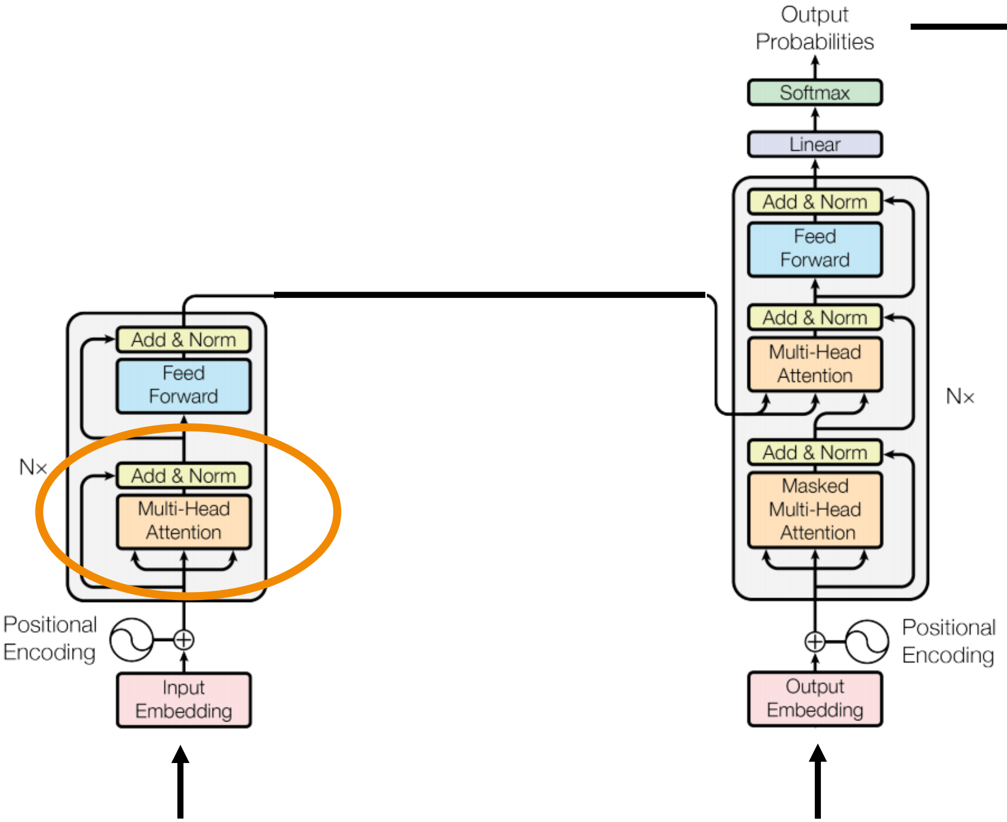


word	embedding		Positional Encoding
Je		+	$\cos(1)\ \sin\left(\frac{1}{10}\right)\ \cos\left(\frac{1}{10}\right)\ \sin\left(\frac{1}{10^2}\right)\ \cos\left(\frac{1}{10^2}\right)\ \sin\left(\frac{1}{10^3}\right)\ \cos\left(\frac{1}{10^3}\right)\ \sin\left(\frac{1}{10^4}\right)$
suis		+	$\cos(2)\ \sin\left(\frac{2}{10}\right)\ \cos\left(\frac{2}{10}\right)\ \sin\left(\frac{2}{10^2}\right)\ \cos\left(\frac{2}{10^2}\right)\ \sin\left(\frac{2}{10^3}\right)\ \cos\left(\frac{2}{10^3}\right)\ \sin\left(\frac{2}{10^4}\right)$
étudiant		+	$\cos(3)\ \sin\left(\frac{3}{10}\right)\ \cos\left(\frac{3}{10}\right)\ \sin\left(\frac{3}{10^2}\right)\ \cos\left(\frac{3}{10^2}\right)\ \sin\left(\frac{3}{10^3}\right)\ \cos\left(\frac{3}{10^3}\right)\ \sin\left(\frac{3}{10^4}\right)$

+

$\cos(1)$	$\sin\left(\frac{1}{10}\right)$	$\cos\left(\frac{1}{10}\right)$	$\sin\left(\frac{1}{10^2}\right)$	$\cos\left(\frac{1}{10^2}\right)$	$\sin\left(\frac{1}{10^3}\right)$	$\cos\left(\frac{1}{10^3}\right)$	$\sin\left(\frac{1}{10^4}\right)$

Multi-Head Attention (Self Attention)



4	5	6	7
8	9	10	2

0	4	5	6	7	8
0	9	10	11	1	2

p_1^1	p_2^1	p_3^1	p_4^1	p_5^1	p_6^1
p_1^2	p_2^2	p_3^2	p_4^2	p_5^2	p_6^2

Multi-Head Attention (Self Attention)

Multi_Head_Attention (

🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩

🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩

🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

)

Multi_Head_Attention (

🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩

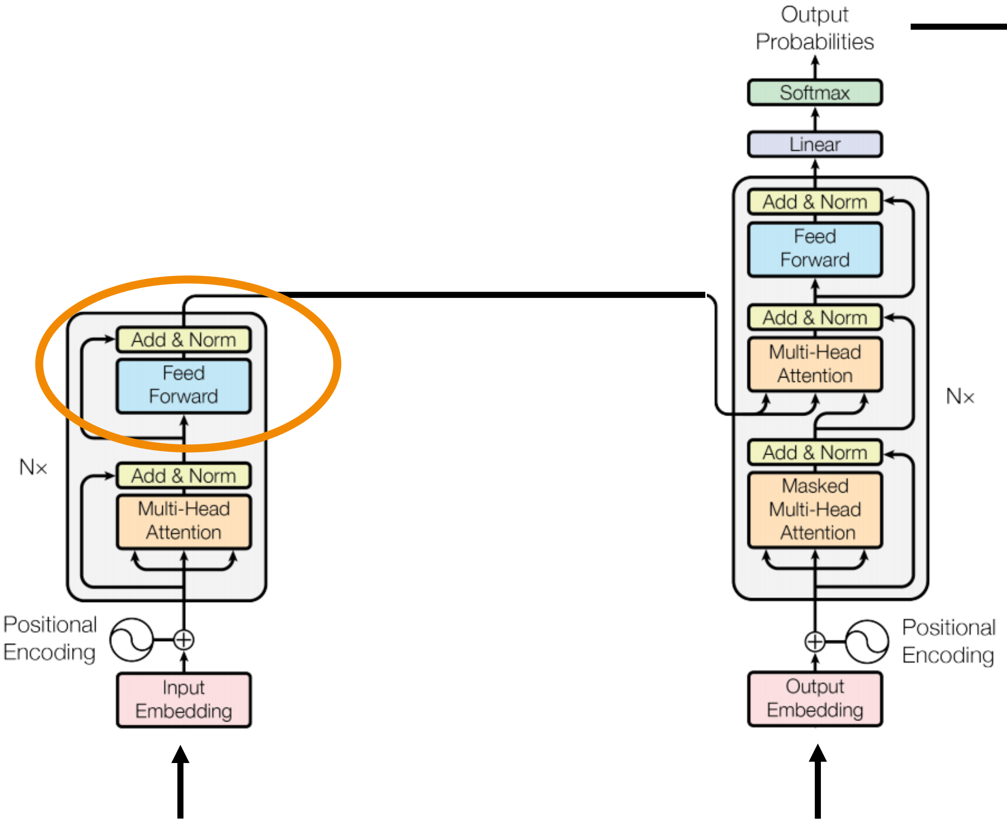
🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩

🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩

0	0	0	1
0	0	0	1
0	0	0	1
0	0	0	1

)

Position-wise Feed-Forward Networks

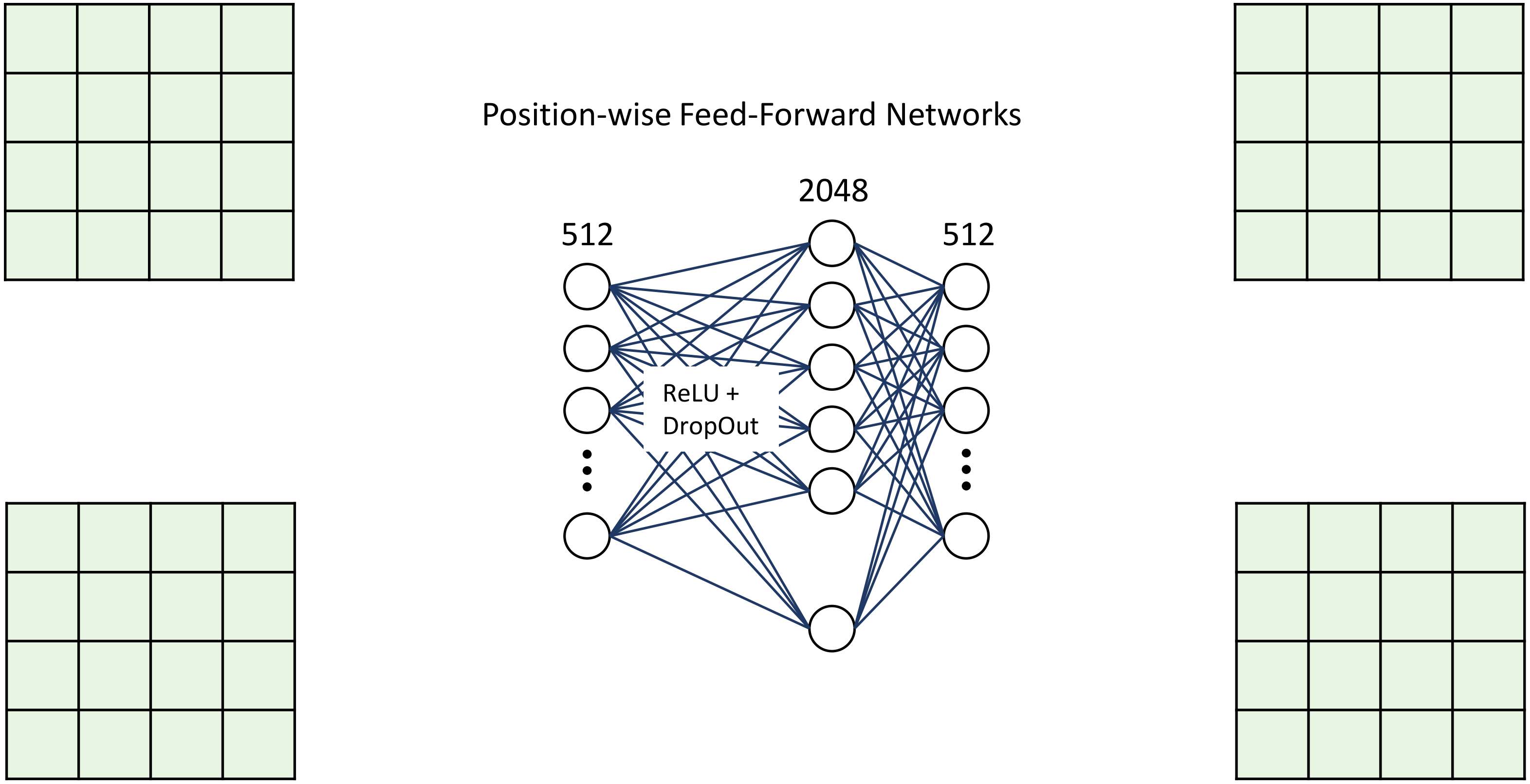


4	5	6	7
8	9	10	2

0	4	5	6	7	8
0	9	10	11	1	2

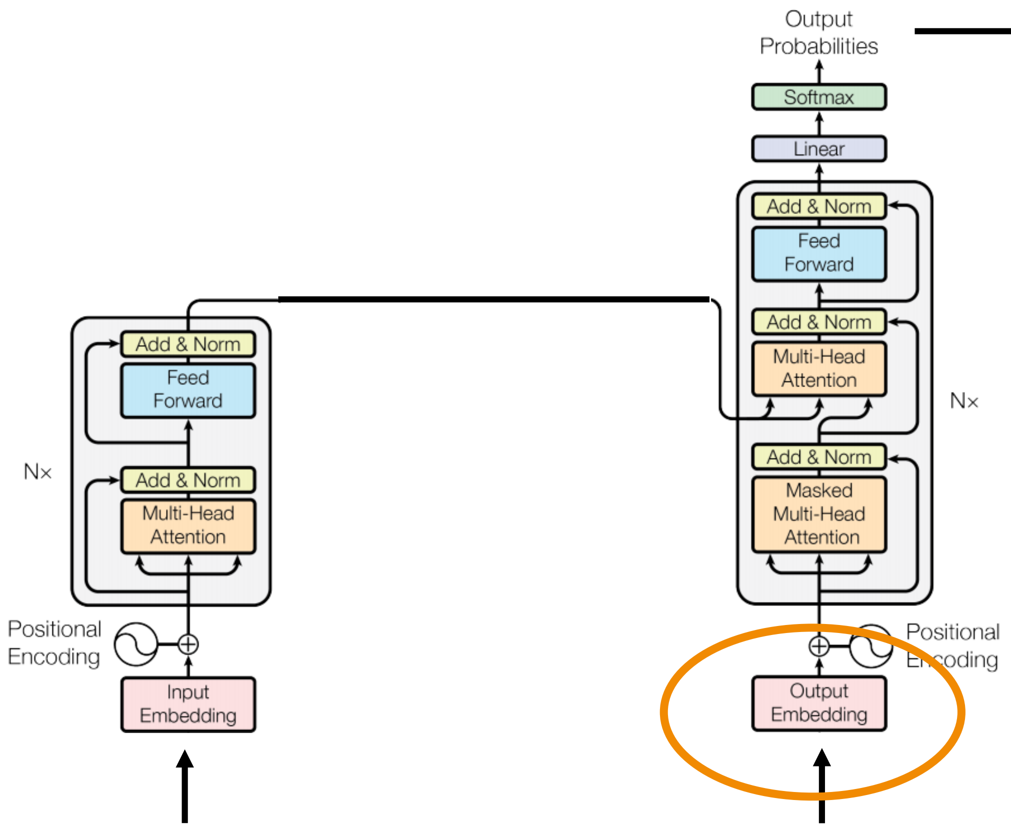
p_1^1	p_2^1	p_3^1	p_4^1	p_5^1	p_6^1
p_1^2	p_2^2	p_3^2	p_4^2	p_5^2	p_6^2

Position-wise Feed-Forward Networks



Position-wise Feed-Forward Networks

Output Embedding

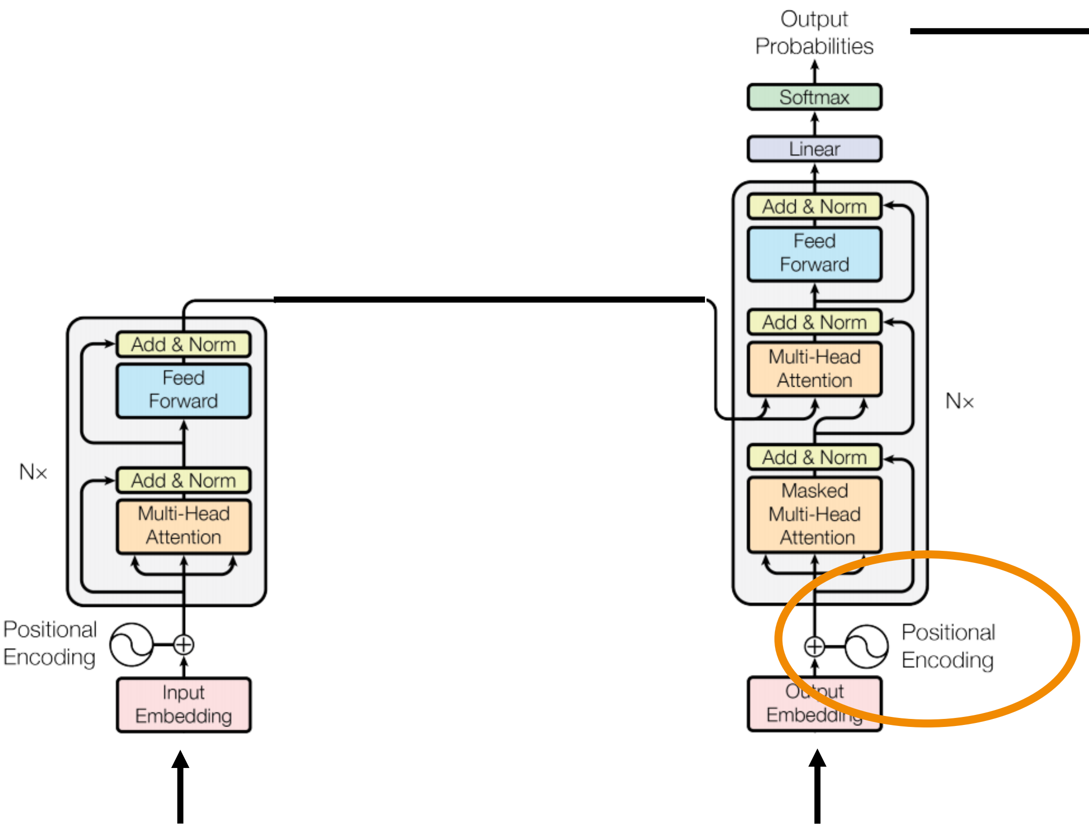


4	5	6	7
8	9	10	2

0	4	5	6	7	8
0	9	10	11	1	2

p_1^1	p_2^1	p_3^1	p_4^1	p_5^1	p_6^1
p_1^2	p_2^2	p_3^2	p_4^2	p_5^2	p_6^2

Positional Encoding

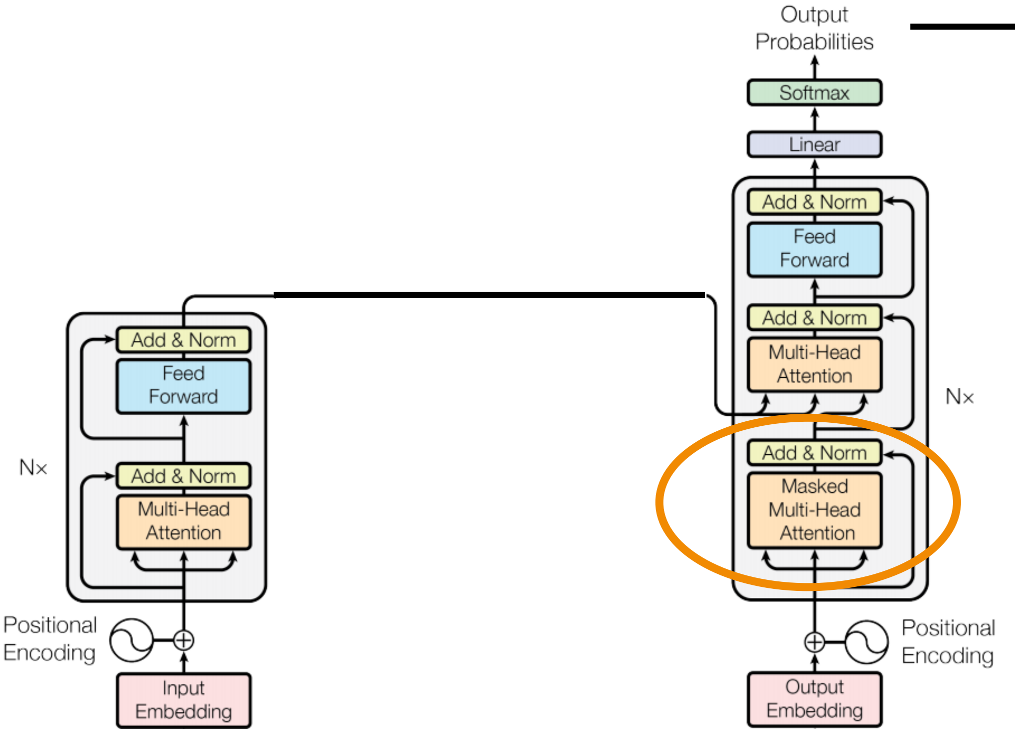


4	5	6	7
8	9	10	2

0	4	5	6	7	8
0	9	10	11	1	2

p_1^1	p_2^1	p_3^1	p_4^1	p_5^1	p_6^1
p_1^2	p_2^2	p_3^2	p_4^2	p_5^2	p_6^2

Multi-Head Attention (Masked Self Attention)



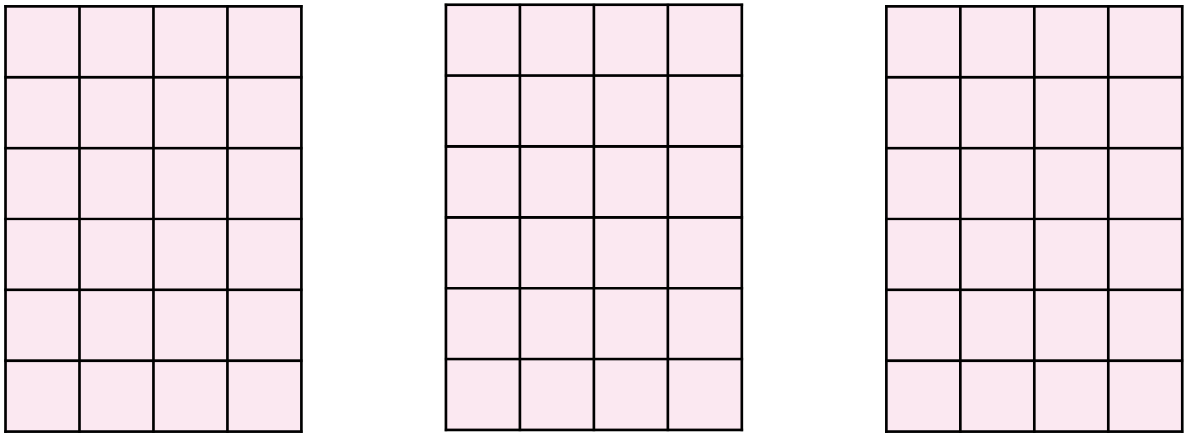
p_1^1	p_2^1	p_3^1	p_4^1	p_5^1	p_6^1
p_1^2	p_2^2	p_3^2	p_4^2	p_5^2	p_6^2

4	5	6	7
8	9	10	2

0	4	5	6	7	8
0	9	10	11	1	2

Multi-Head Attention (Masked Self Attention)

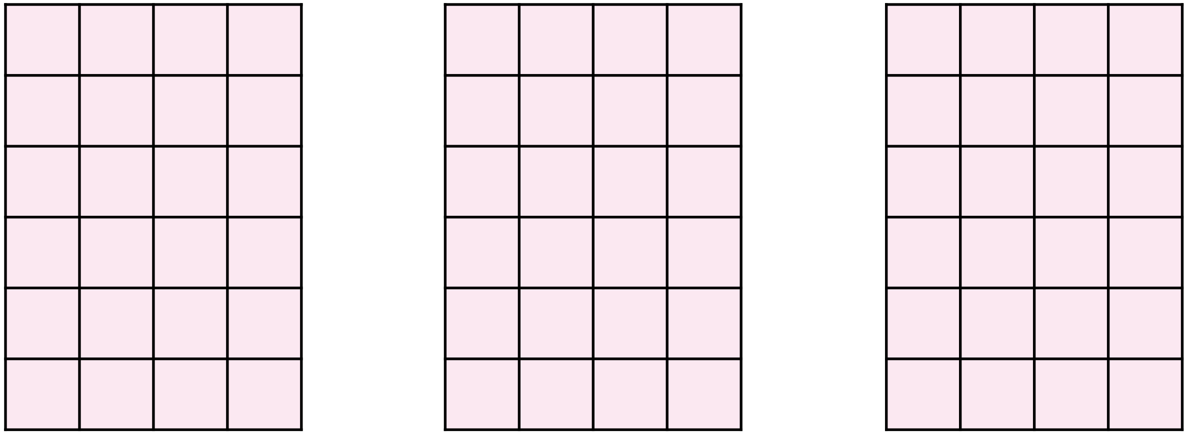
Multi_Head_Attention (



0	1	1	1	1	1
0	0	1	1	1	1
0	0	0	1	1	1
0	0	0	0	1	1
0	0	0	0	0	1
0	0	0	0	0	0

)

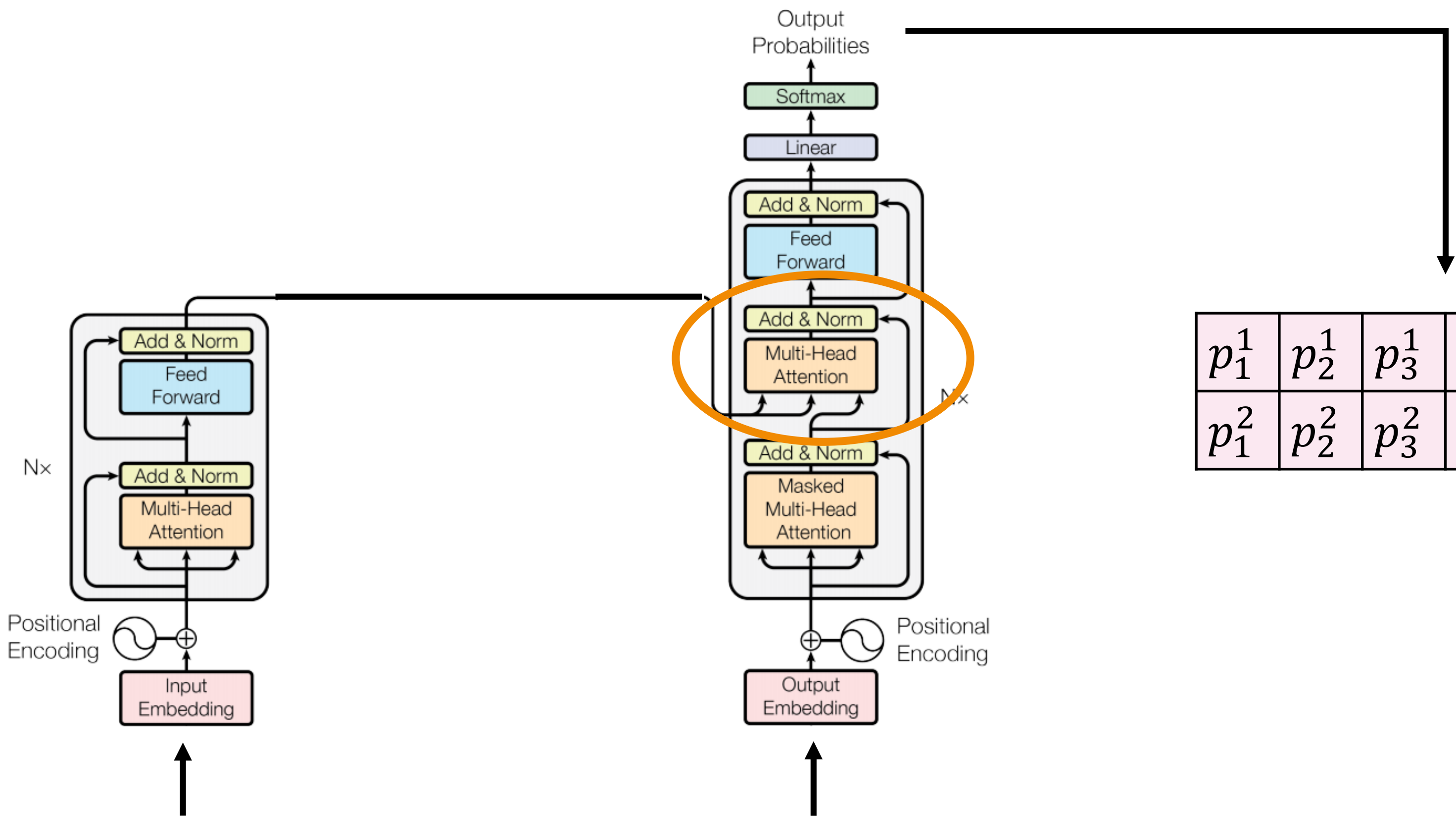
Multi_Head_Attention (



0	1	1	1	1	1
0	0	1	1	1	1
0	0	0	1	1	1
0	0	0	0	1	1
0	0	0	0	0	1
0	0	0	0	0	1

)

Multi-Head Attention (Original Attention)



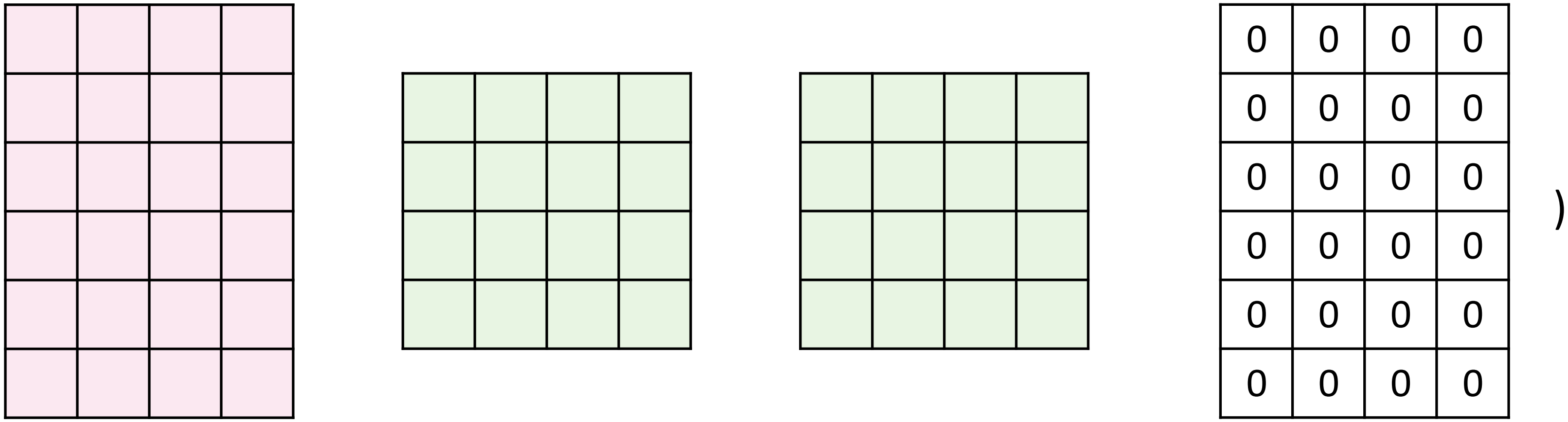
4	5	6	7
8	9	10	2

0	4	5	6	7	8
0	9	10	11	1	2

p_1^1	p_2^1	p_3^1	p_4^1	p_5^1	p_6^1
p_1^2	p_2^2	p_3^2	p_4^2	p_5^2	p_6^2

Multi-Head Attention (Original Attention)

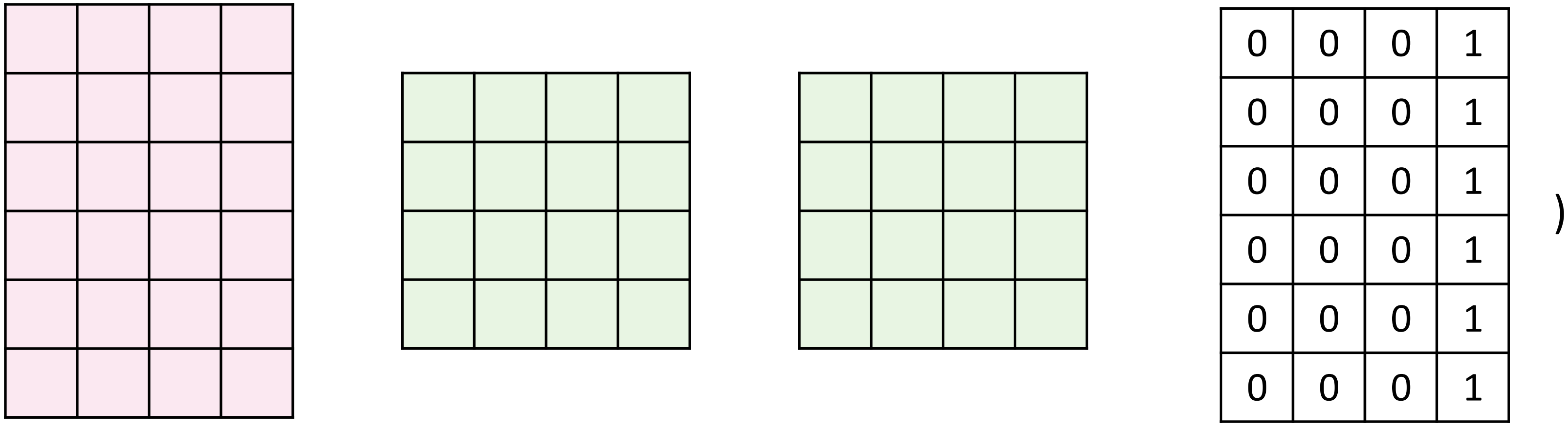
Multi_Head_Attention (



0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

)

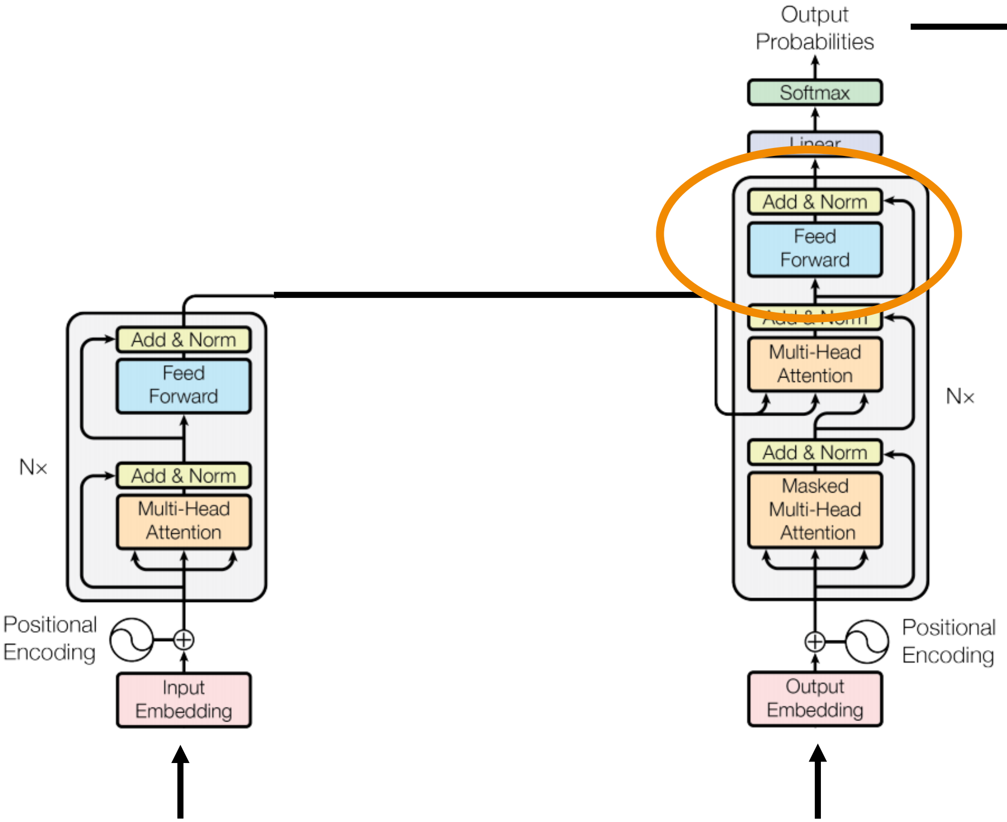
Multi_Head_Attention (



0	0	0	1
0	0	0	1
0	0	0	1
0	0	0	1
0	0	0	1
0	0	0	1

)

Position-wise Feed-Forward Networks

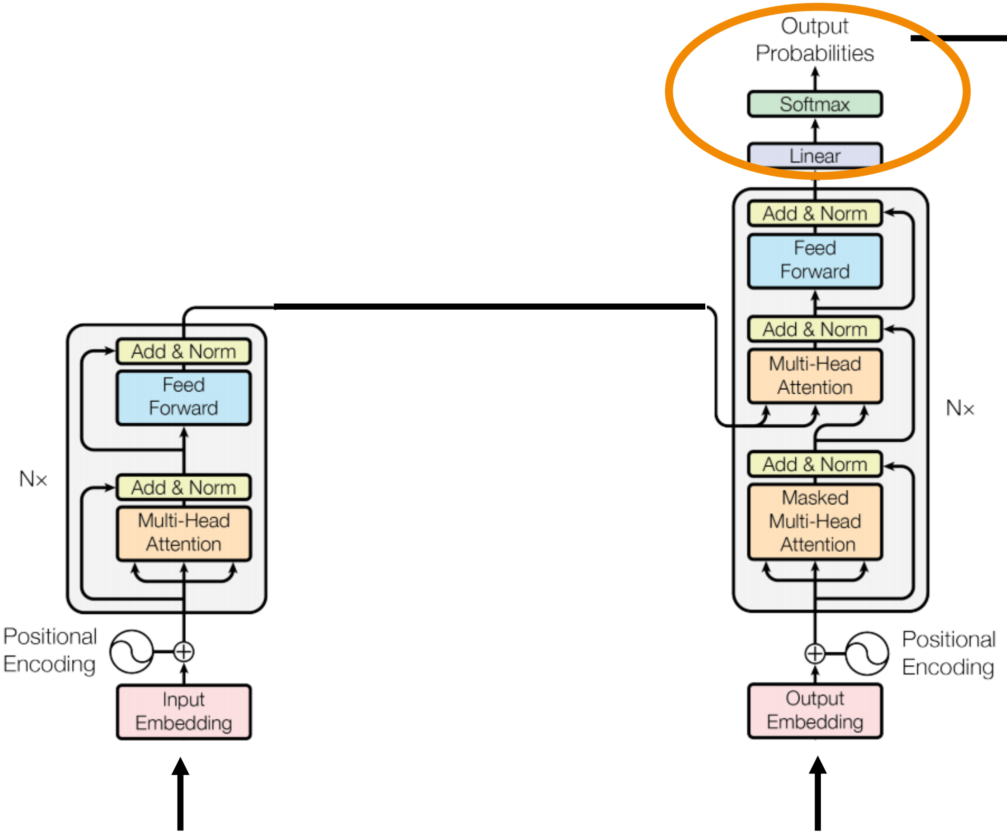


4	5	6	7
8	9	10	2

0	4	5	6	7	8
0	9	10	11	1	2

p_1^1	p_2^1	p_3^1	p_4^1	p_5^1	p_6^1
p_1^2	p_2^2	p_3^2	p_4^2	p_5^2	p_6^2

Prediction



4	5	6	7
8	9	10	2

0	4	5	6	7	8
0	9	10	11	1	2

p_1^1	p_2^1	p_3^1	p_4^1	p_5^1	p_6^1
p_1^2	p_2^2	p_3^2	p_4^2	p_5^2	p_6^2

References

1) Attention Is All You Need (NeurIPS'17)
    - https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf
2) Neural Machine Translation by Jointly Learning to Align and Translate (ICLR'15)
    - https://arxiv.org/pdf/1409.0473.pdf
3) Transformers From Scratch
    - http://www.peterbloem.nl/blog/transformers
4) The Illustrated Transformer
    - https://jalammar.github.io/illustrated-transformer/
5) The Annotated Transformer
    - https://nlp.seas.harvard.edu/2018/04/03/attention.html
6) A Brief Overview of Attention Mechanism
    - https://medium.com/syncedreview/a-brief-overview-of-attention-mechanism-13c578ba9129

Thank you !

Any Questions ?