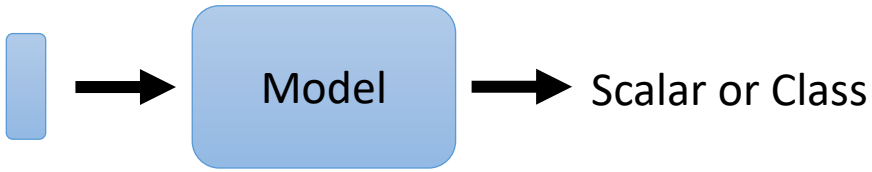
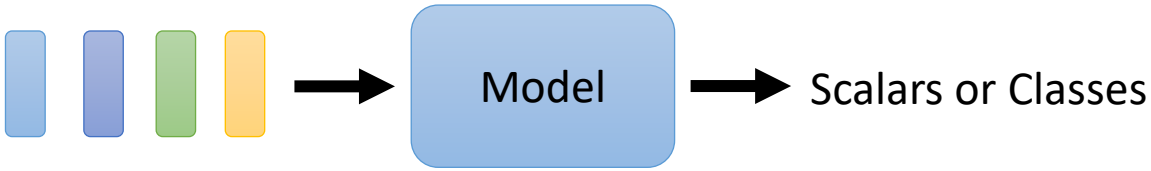
Self-Attention, Transformers, BERT, GPT

Sophisticated Input

- Input is **a vector**

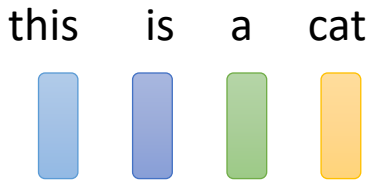


- Input is **a set of vectors**



(may change length)

Vector Set as Input



A 4-dimensional embedding

cat =>	1.2	-0.1	4.3	3.2
mat =>	0.4	2.5	-0.9	0.5
on =>	2.1	0.3	0.1	0.4
...	...			

One-hot Encoding

apple = [1 0 0 0 0 .....]

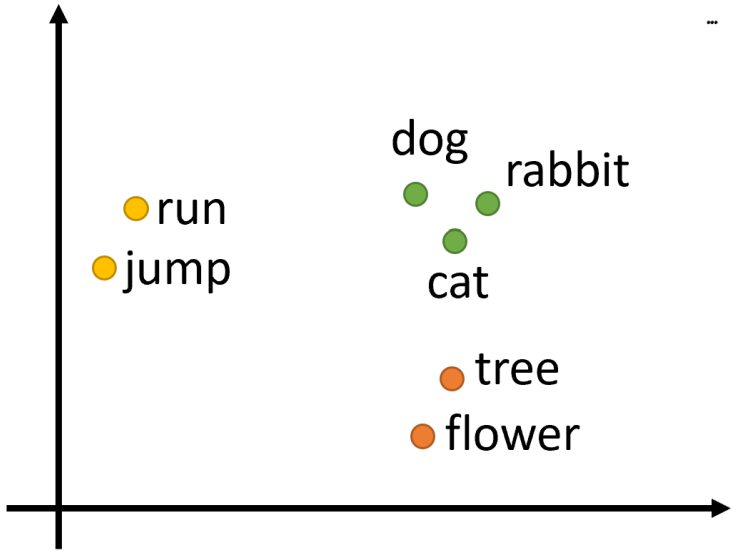
bag = [0 1 0 0 0 .....]

cat = [0 0 1 0 0 .....]

dog = [0 0 0 1 0 .....]

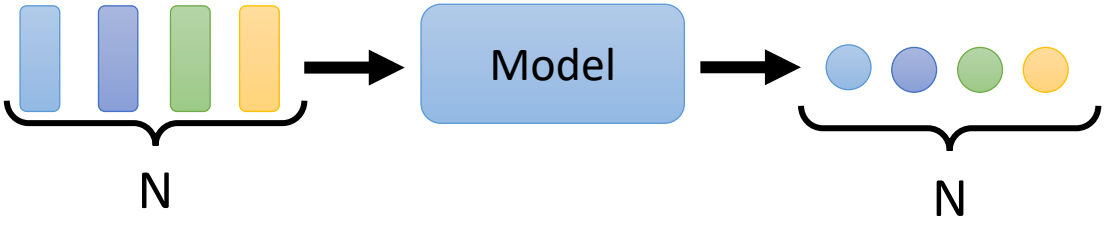
elephant = [0 0 0 0 1 .....]

Word Embedding

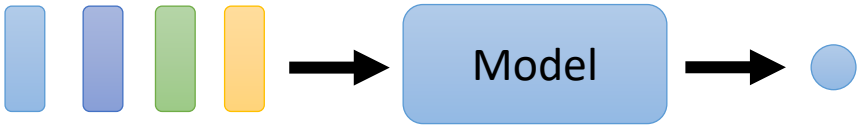


What is the output?

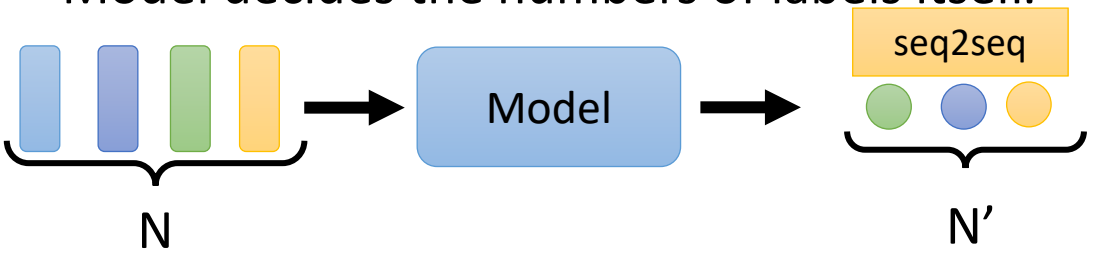
- Each vector has a label.

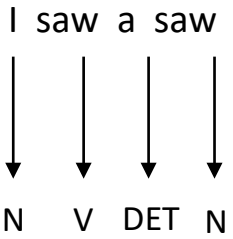


- The whole sequence has a label.



- Model decides the numbers of labels itself.





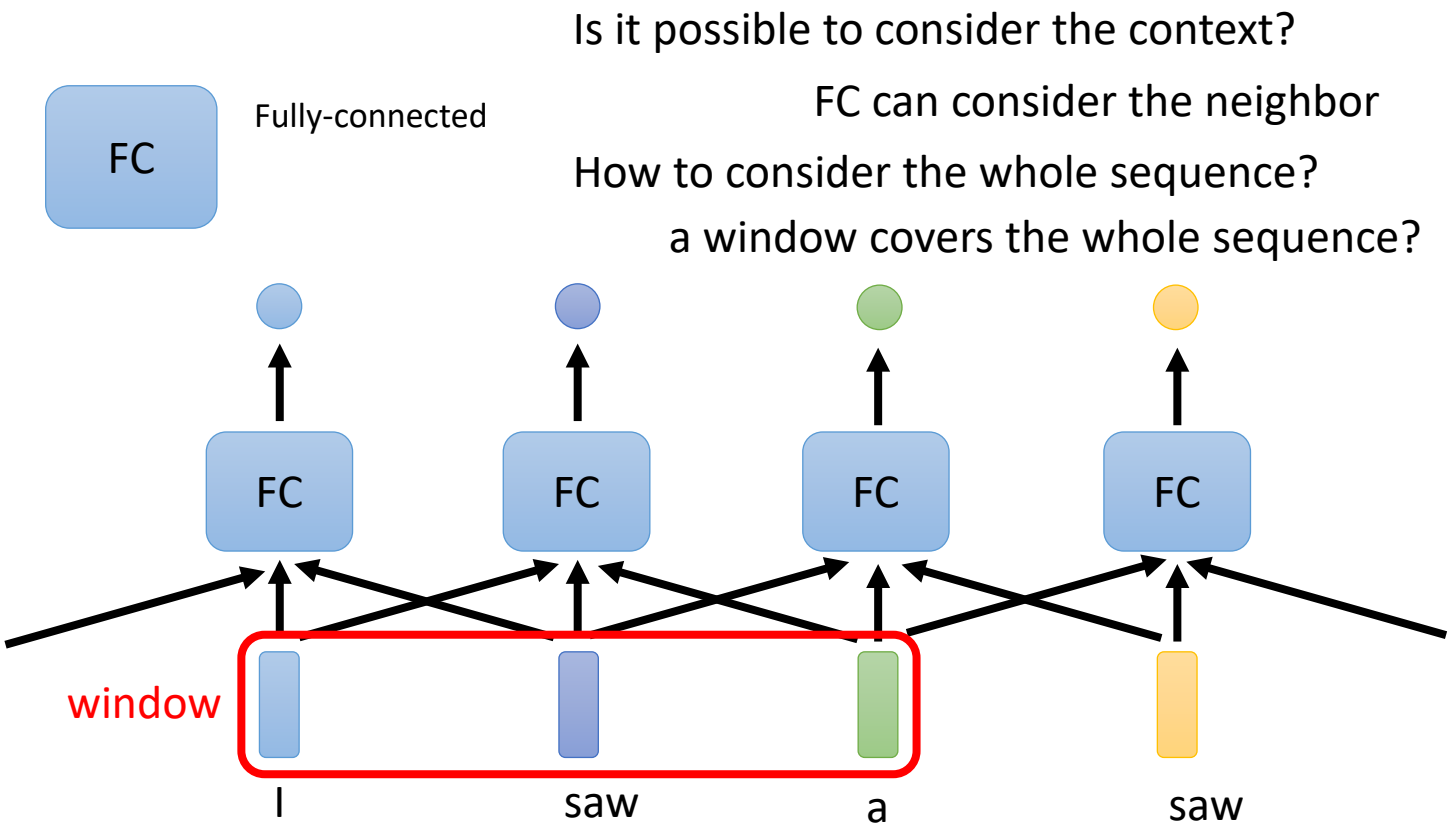
POS tagging

this is good

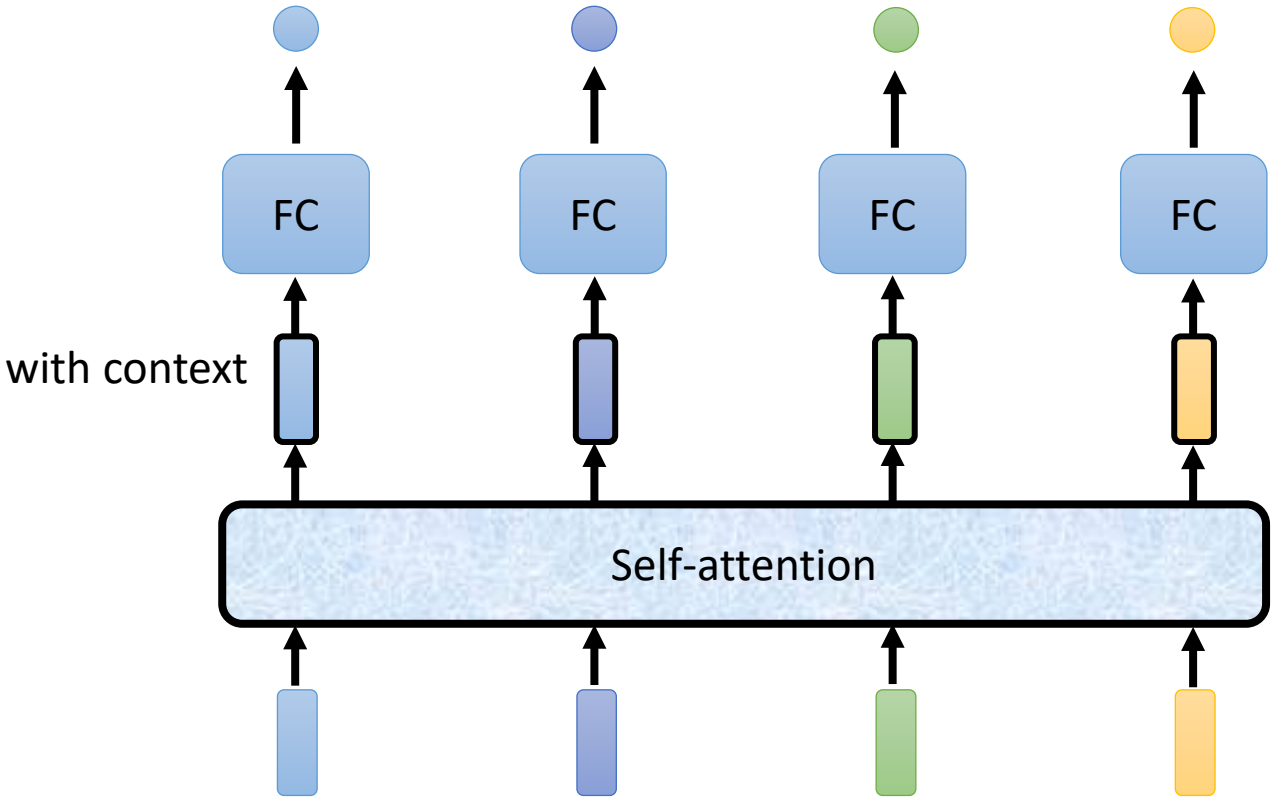
Sentiment analysis

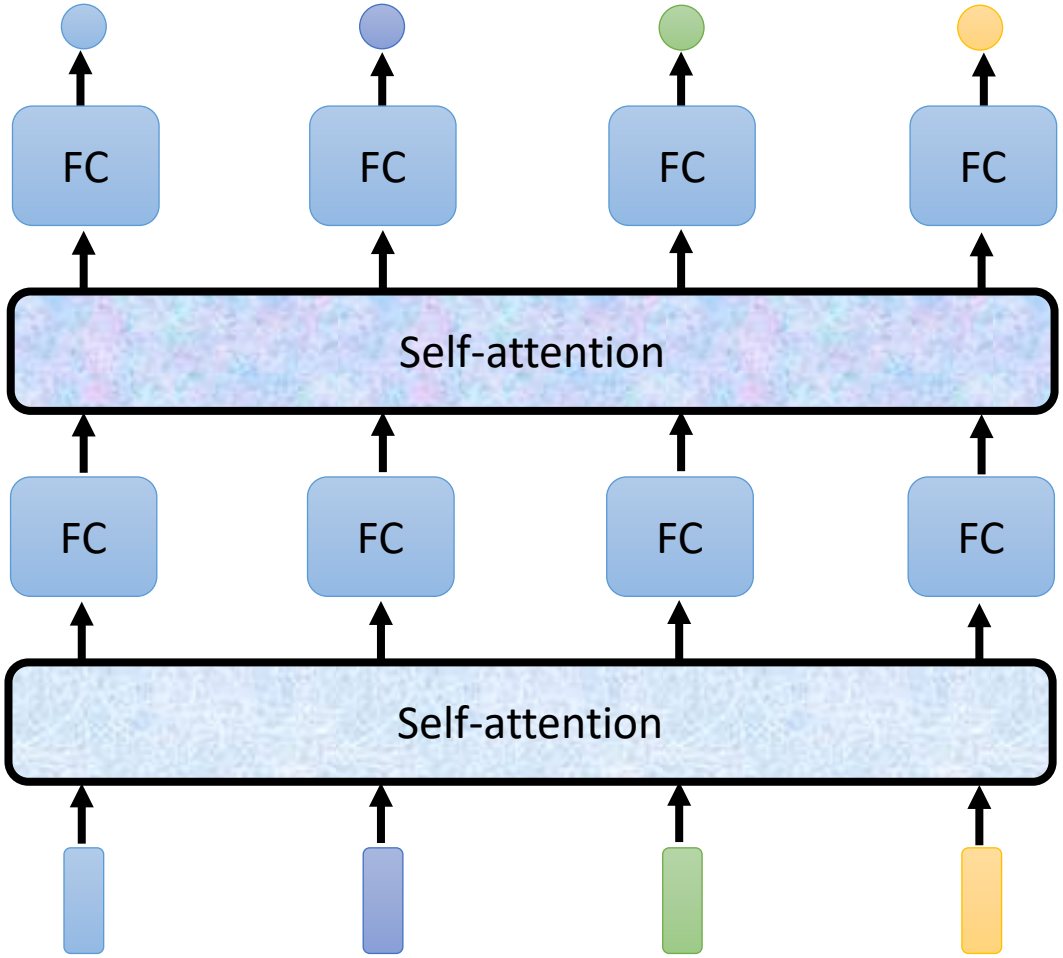
positive

Sequence Labeling



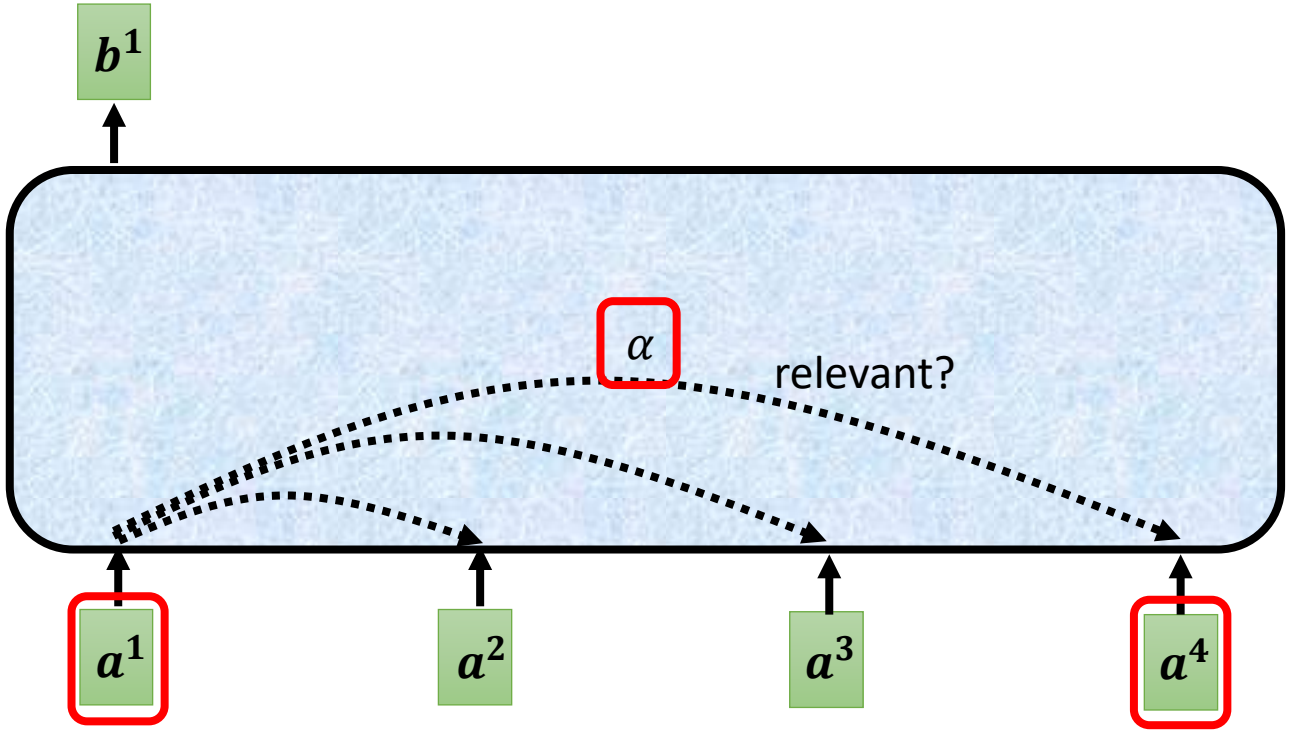
Self-attention





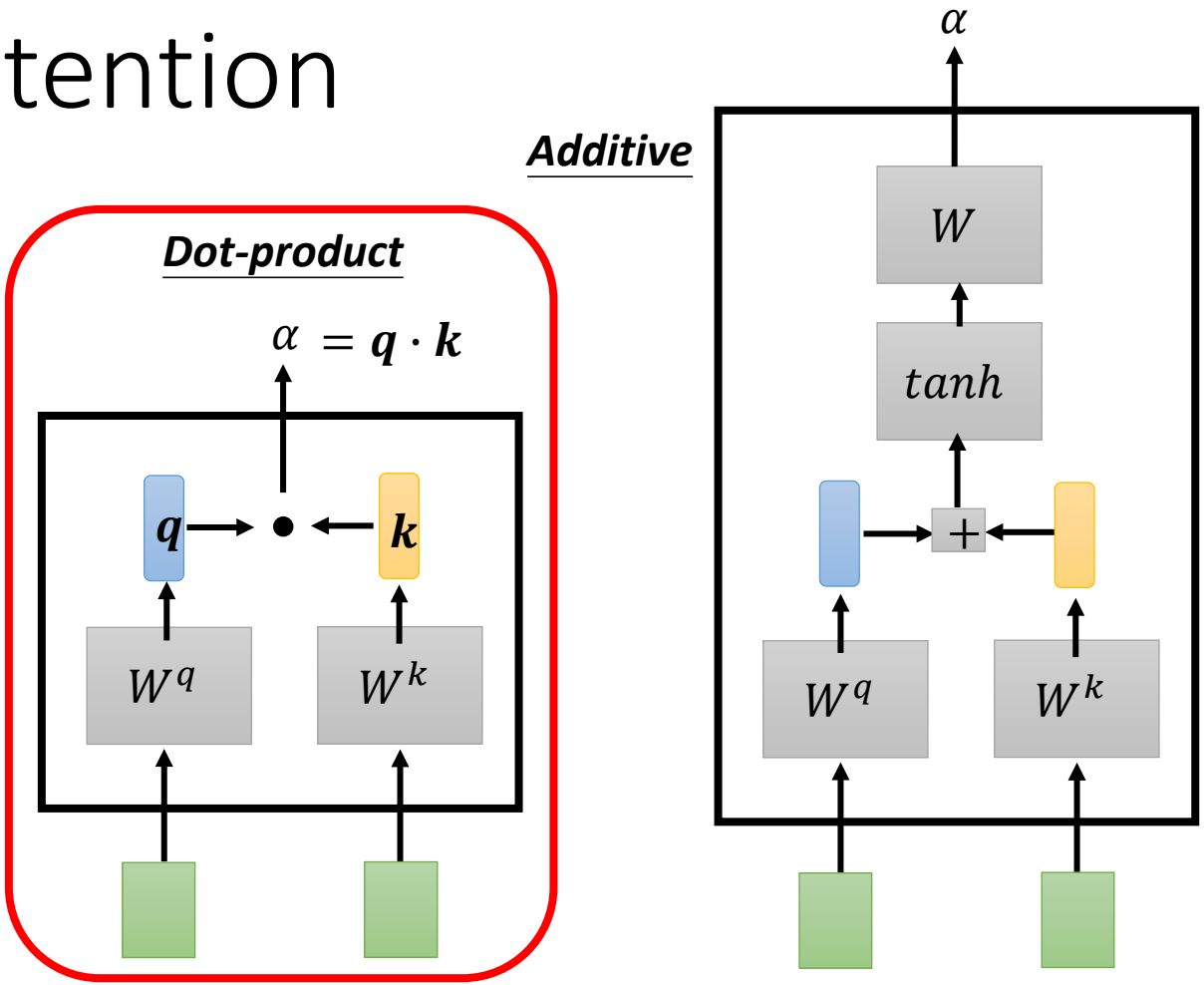
https://arxiv.org/abs/1706.03762

Self-attention

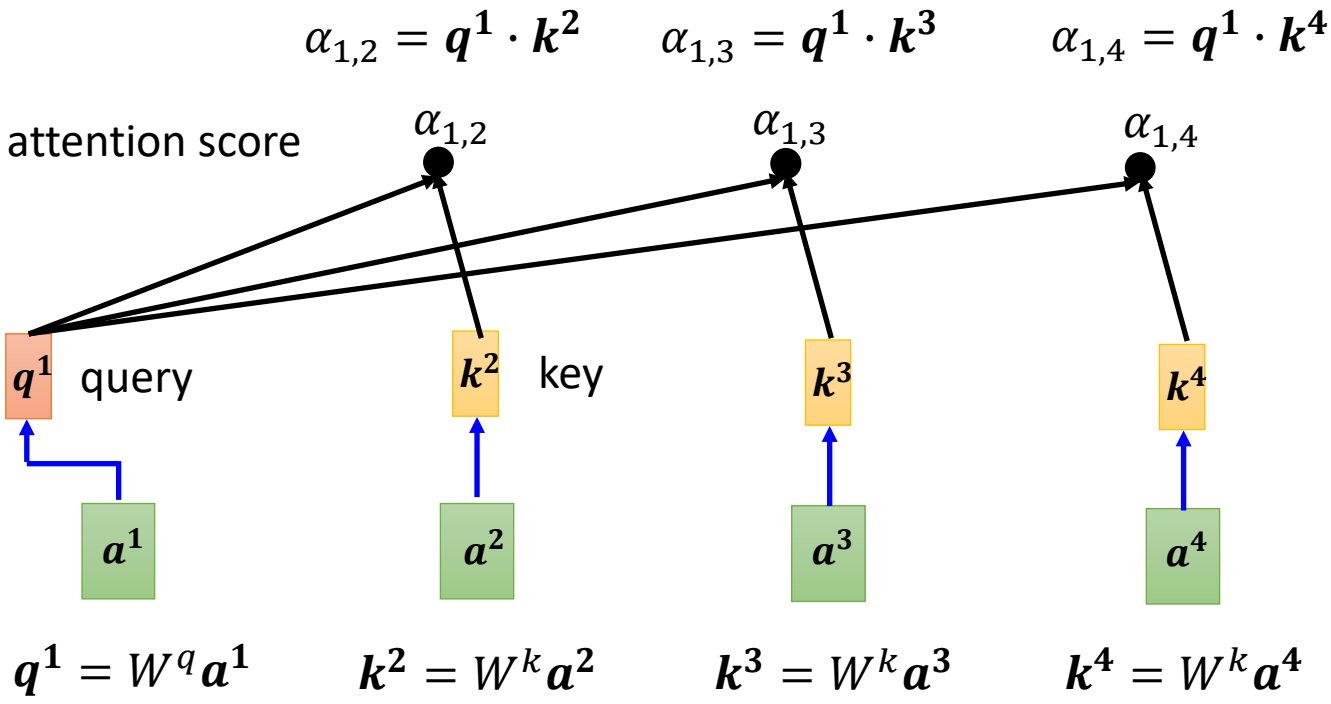


Find the relevant vectors in a sequence

Self-attention

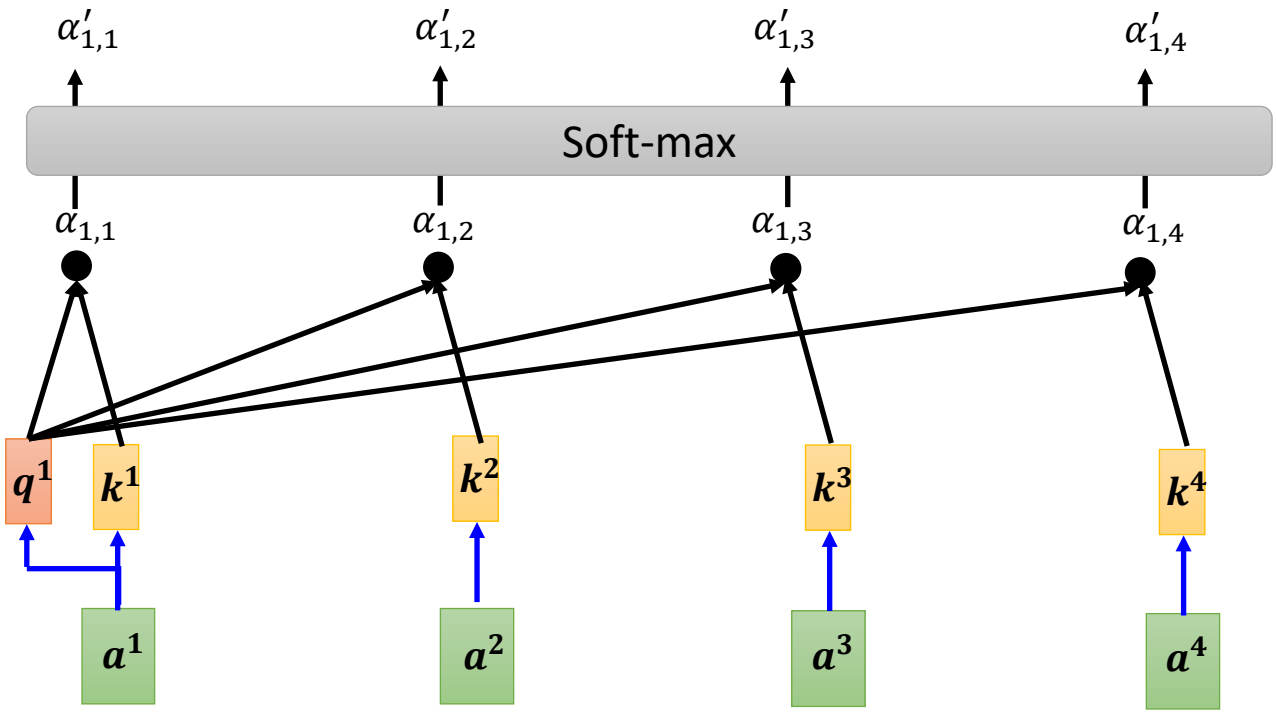


Self-attention



Self-attention

$$\alpha'_{1,i} = exp(\alpha_{1,i}) / \sum_j exp(\alpha_{1,j})$$



$q^1 = W^q a^1$

$k^1 = W^k a^1$

$k^2 = W^k a^2$

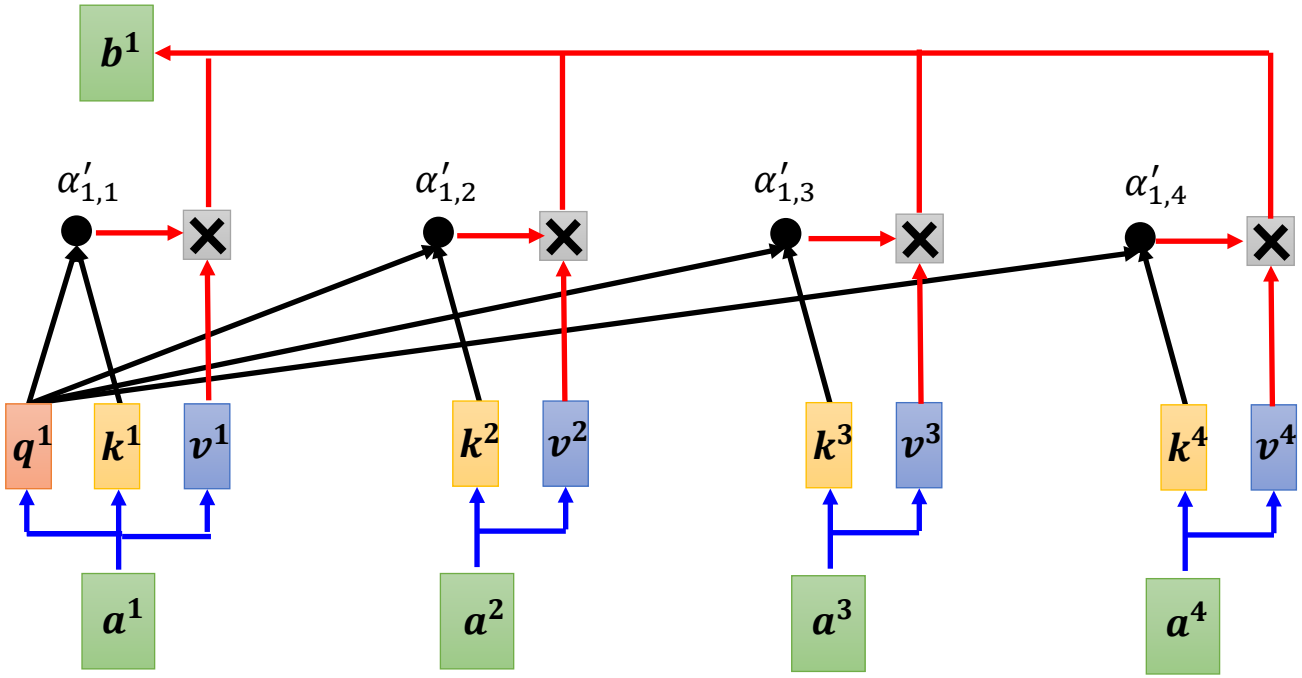
$k^3 = W^k a^3$

$k^4 = W^k a^4$

Self-attention

Extract information based on attention scores

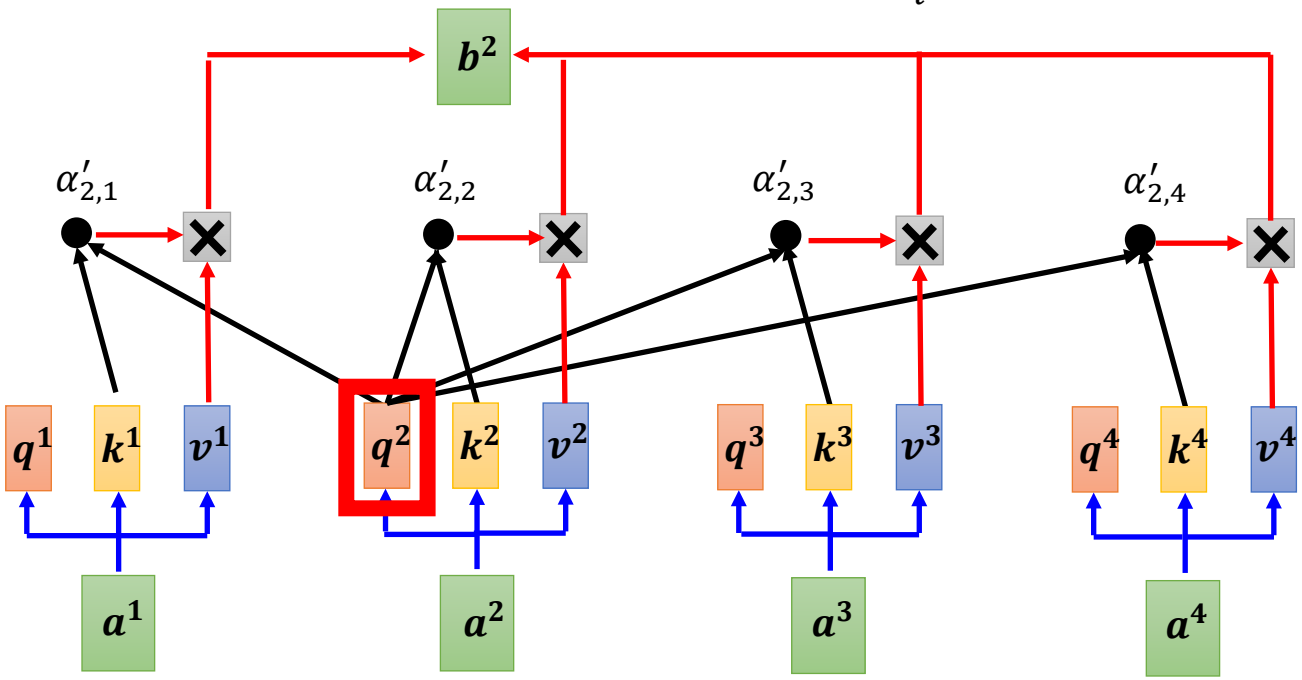
$$\boldsymbol{b^1} = \sum_i \alpha'_{1,i} \boldsymbol{v^i}$$



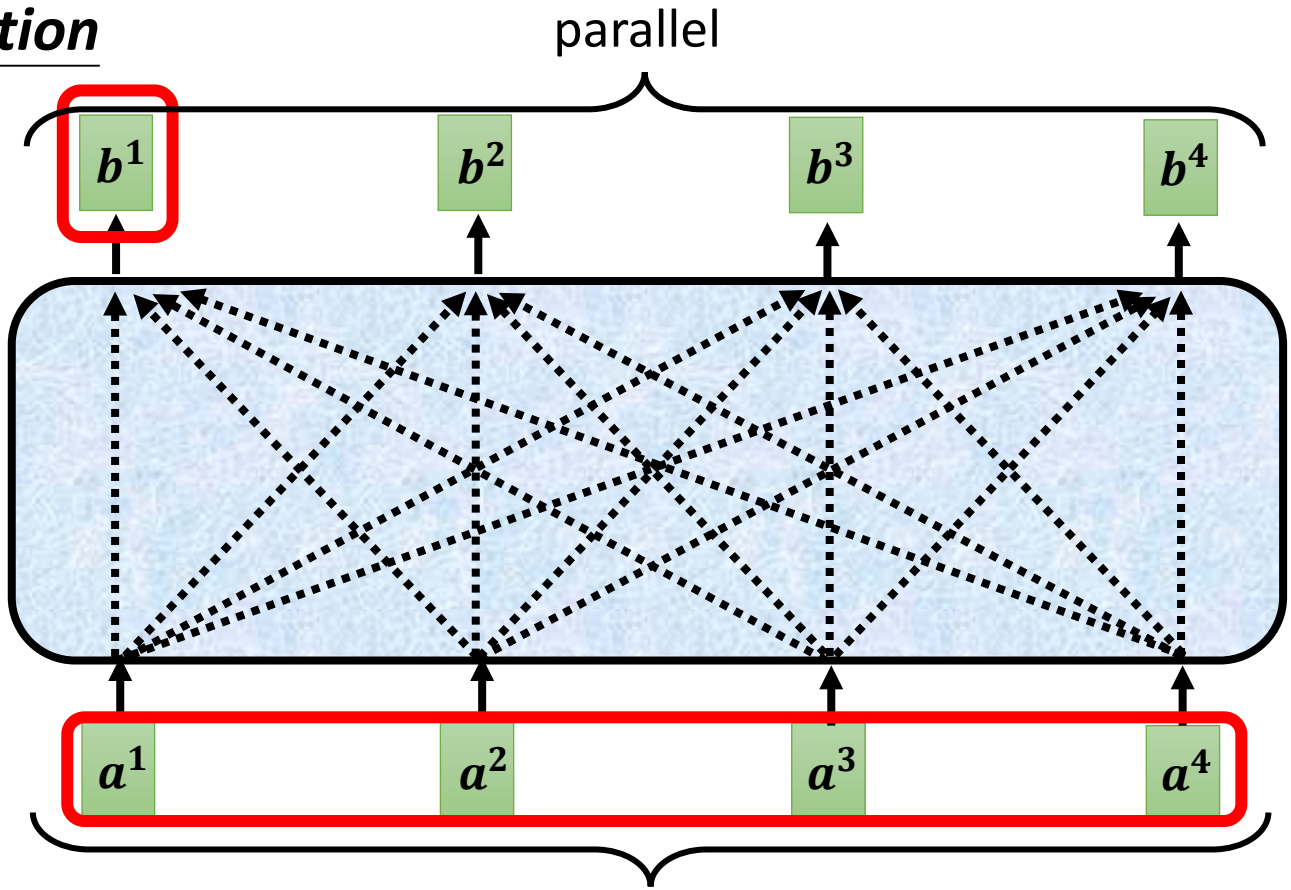
$\boldsymbol{v^1} = W^v \boldsymbol{a^1}$ $\boldsymbol{v^2} = W^v \boldsymbol{a^2}$ $\boldsymbol{v^3} = W^v \boldsymbol{a^3}$ $\boldsymbol{v^4} = W^v \boldsymbol{a^4}$

Self-attention

$$b^2 = \sum_i \alpha'_{2,i} v^i$$

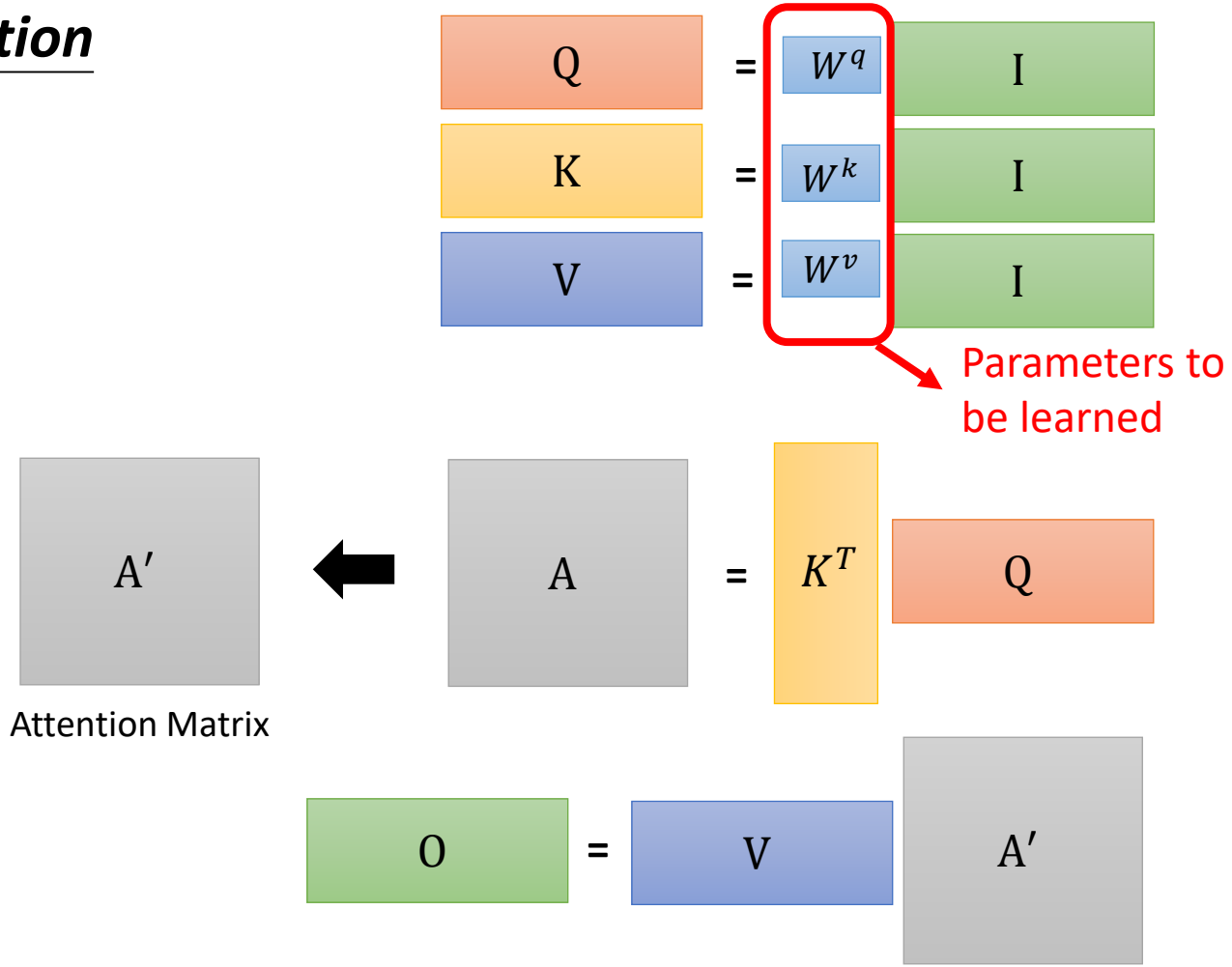


Self-attention



Can be either **input** or **a hidden layer**

Self-attention

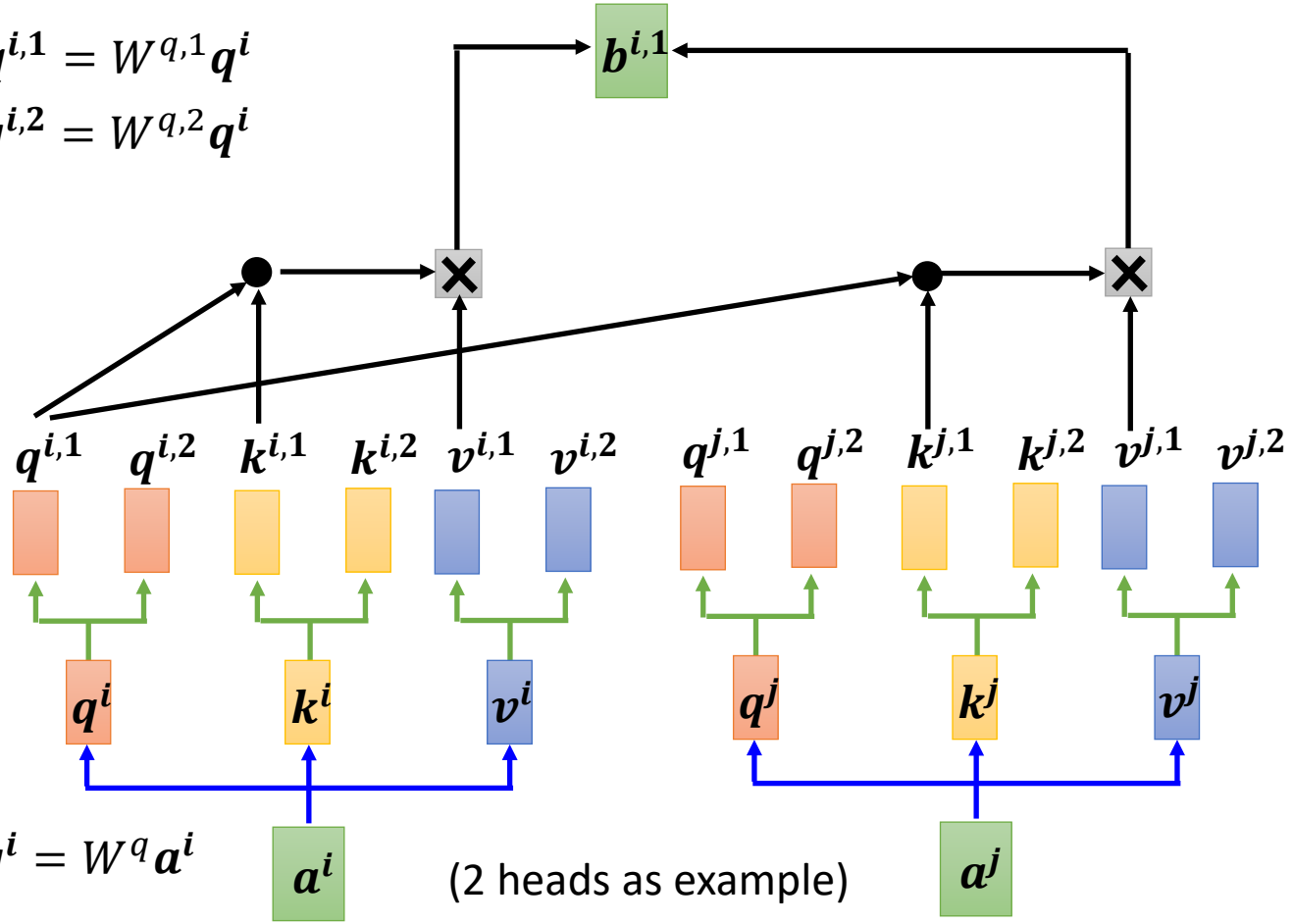


Multi-head Self-attention

Different types of relevance

$q^{i,1} = W^{q,1} q^i$

$q^{i,2} = W^{q,2} q^i$



$q^i = W^q a^i$

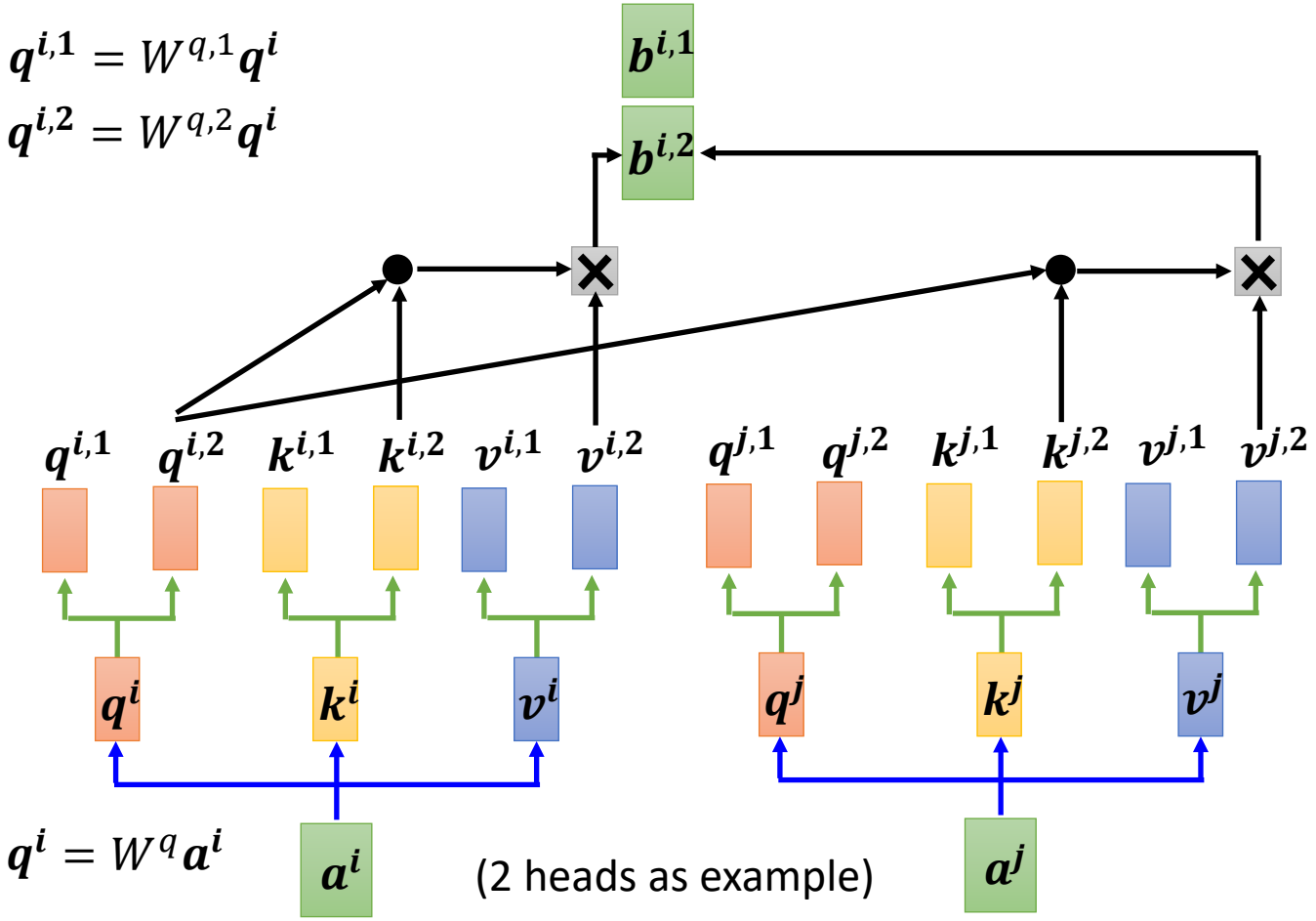
(2 heads as example)

Multi-head Self-attention

Different types of relevance

$$q^{i,1} = W^{q,1} q^i$$

$$q^{i,2} = W^{q,2} q^i$$

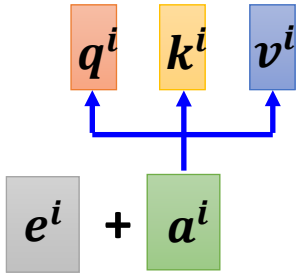


$$q^i = W^q a^i$$

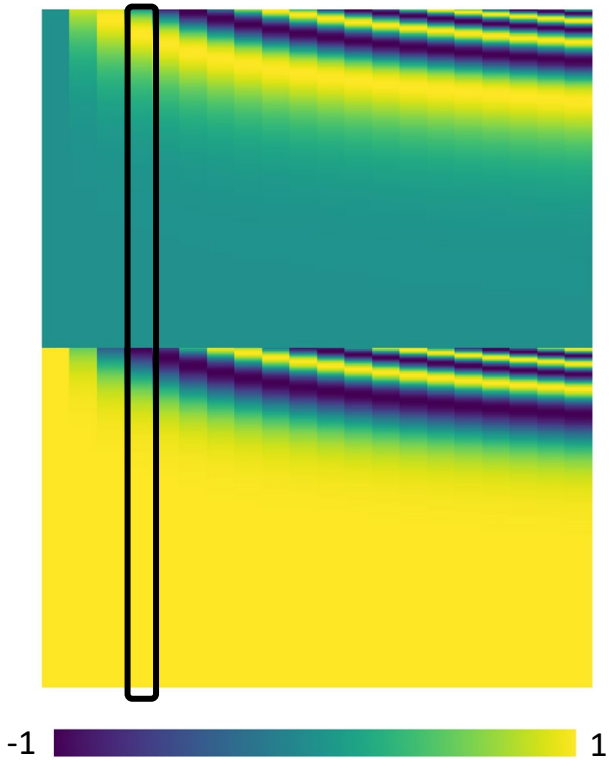
(2 heads as example)

Positional Encoding

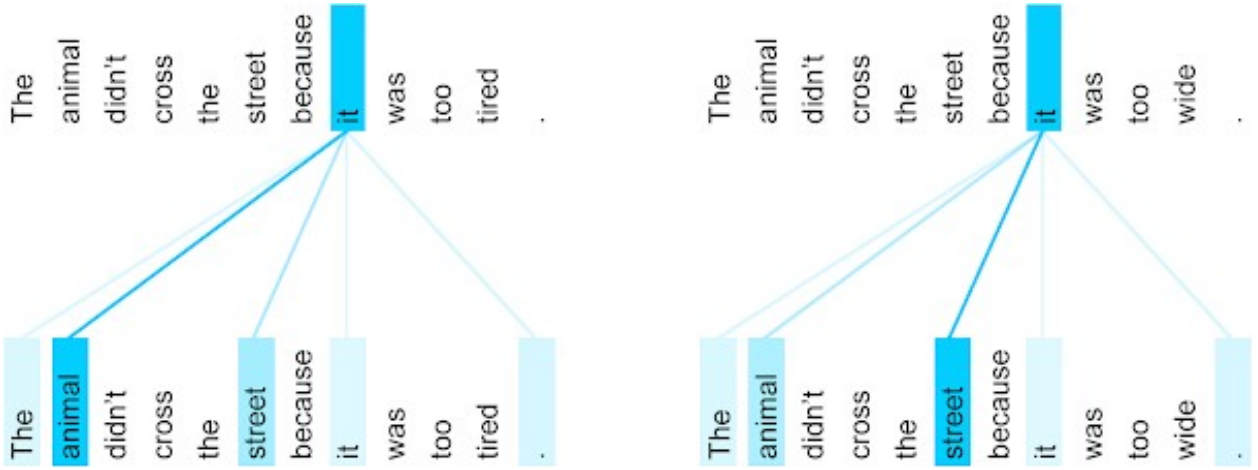
- No position information in self-attention.
- Each position has a unique positional vector e^i
- **hand-crafted**
- **learned from data**



Each column represents a positional vector e^i



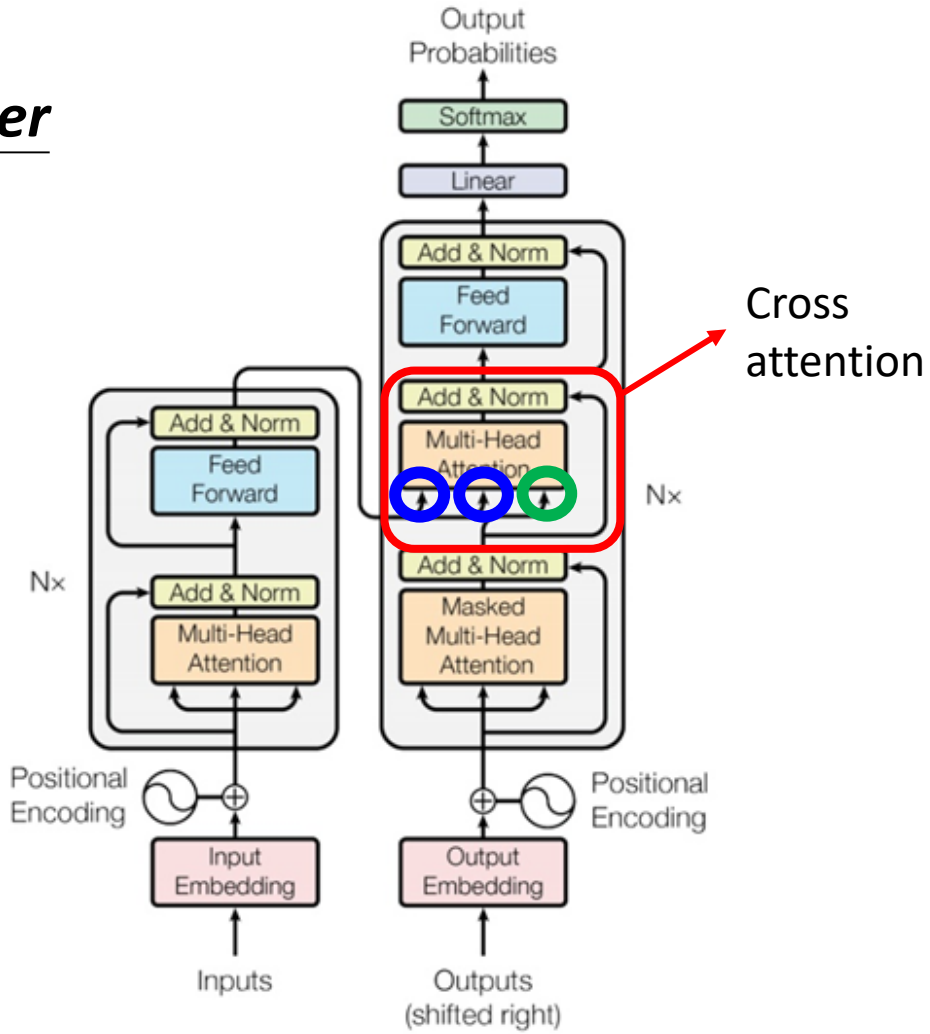
Attention Visualization



The encoder self-attention distribution for the word “it” from the 5th to the 6th layer of a Transformer trained on English to French translation (one of eight attention heads).

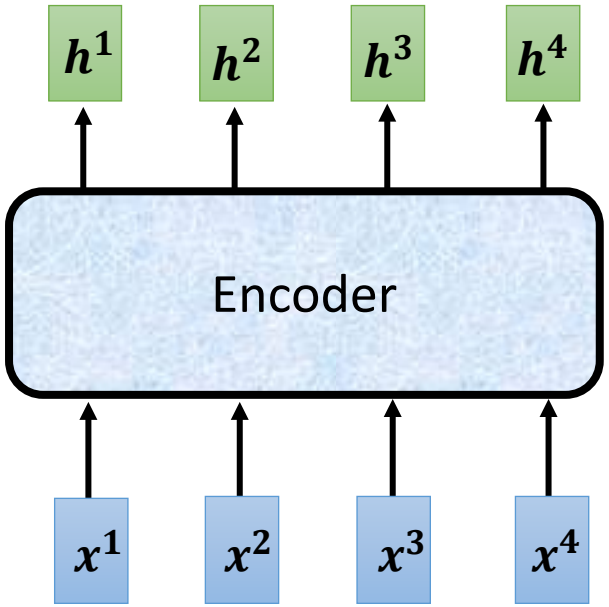
https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

Transformer

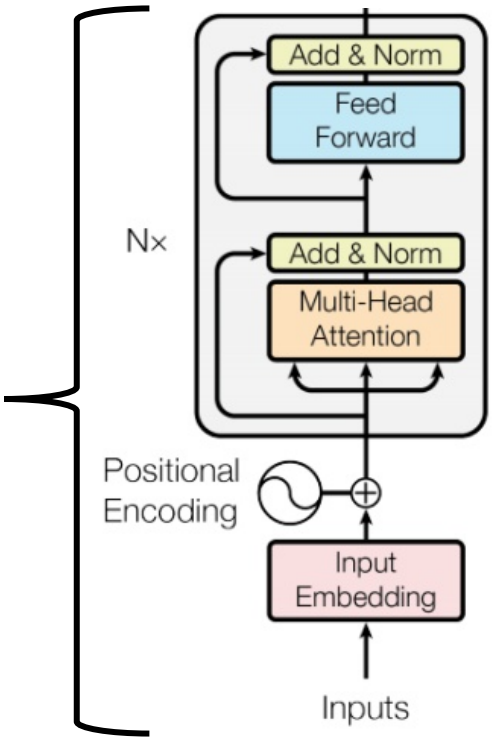


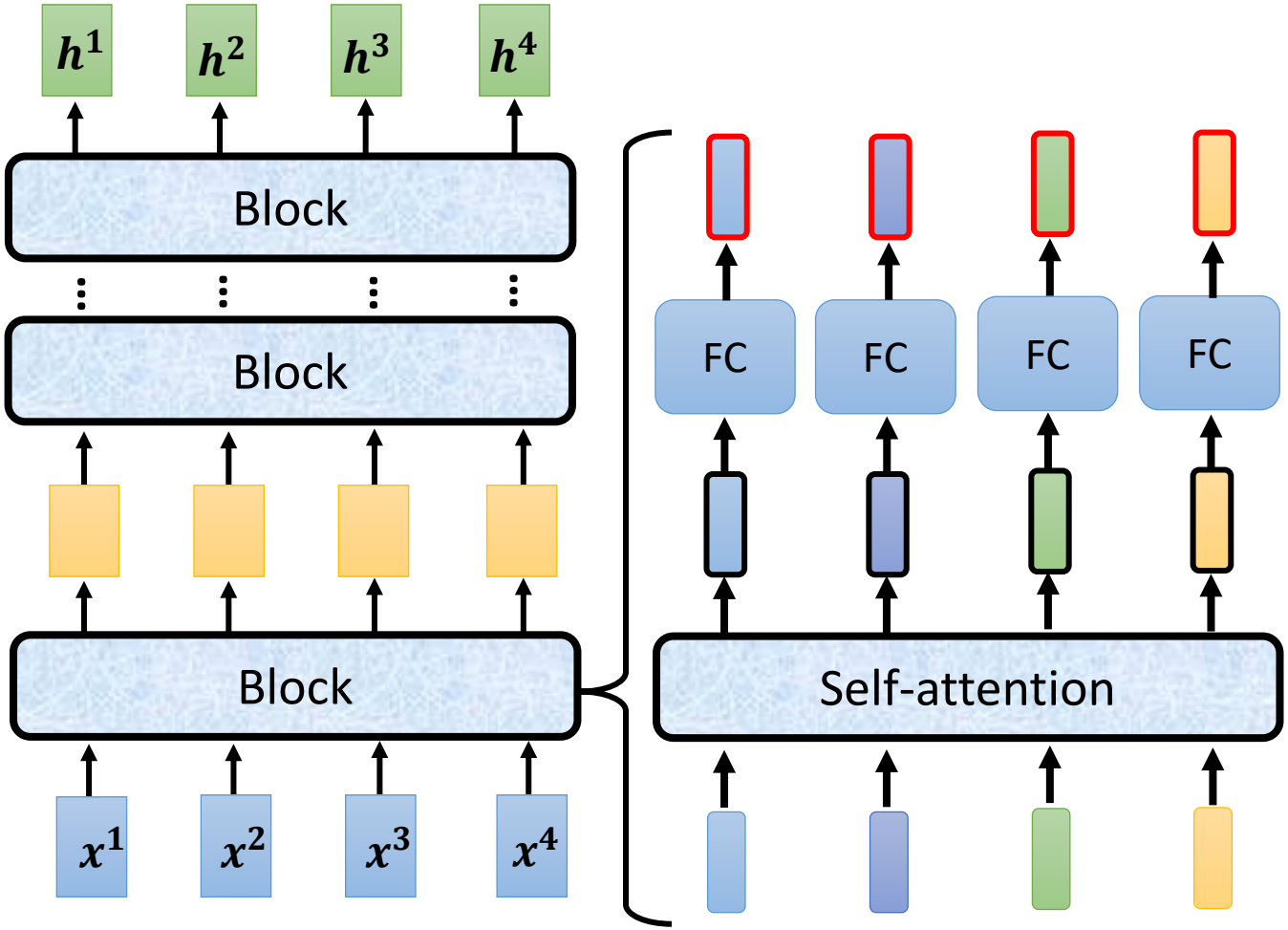
Encoder

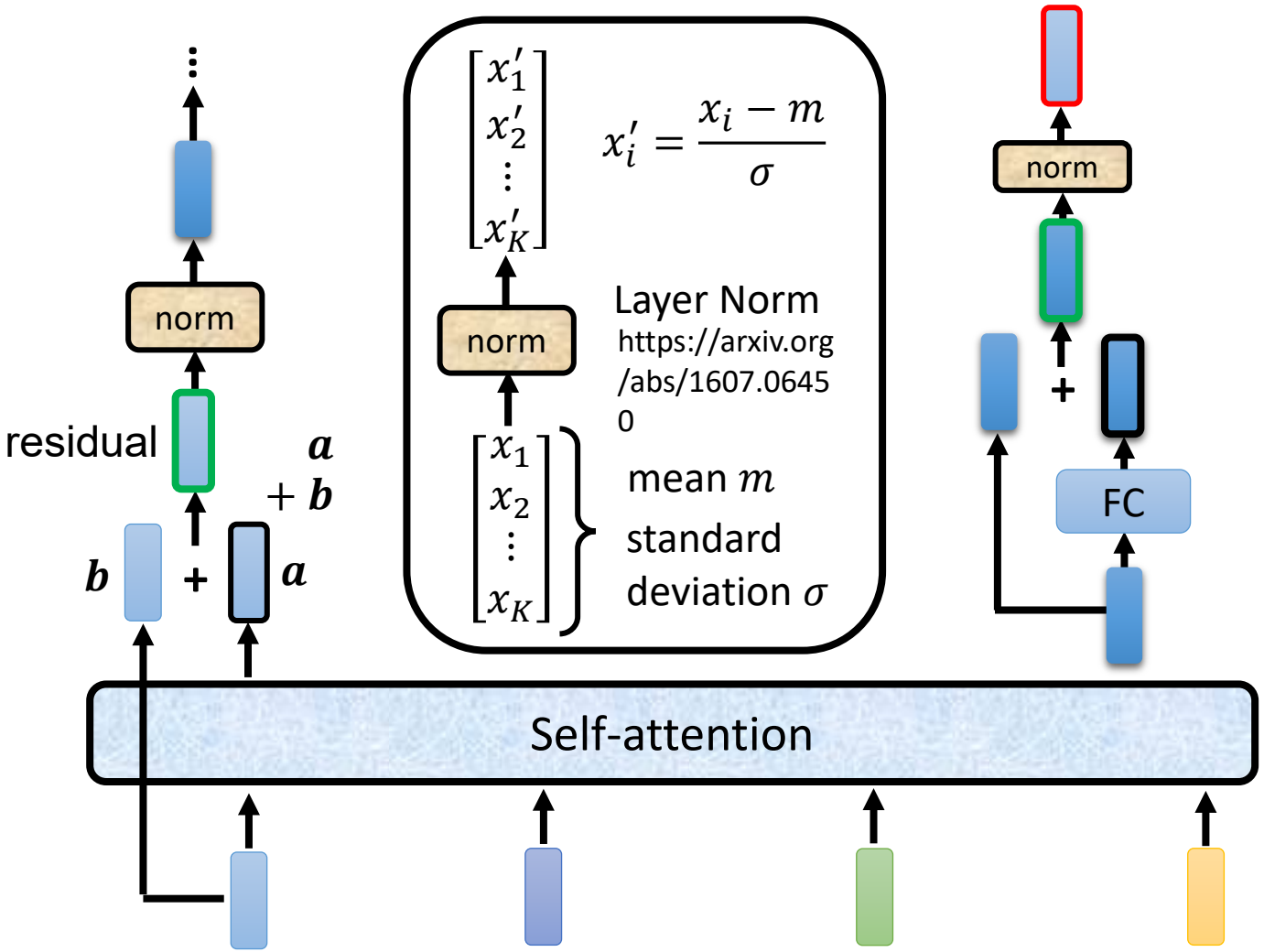
You can use **RNN** or **CNN**.



Transformer's Encoder

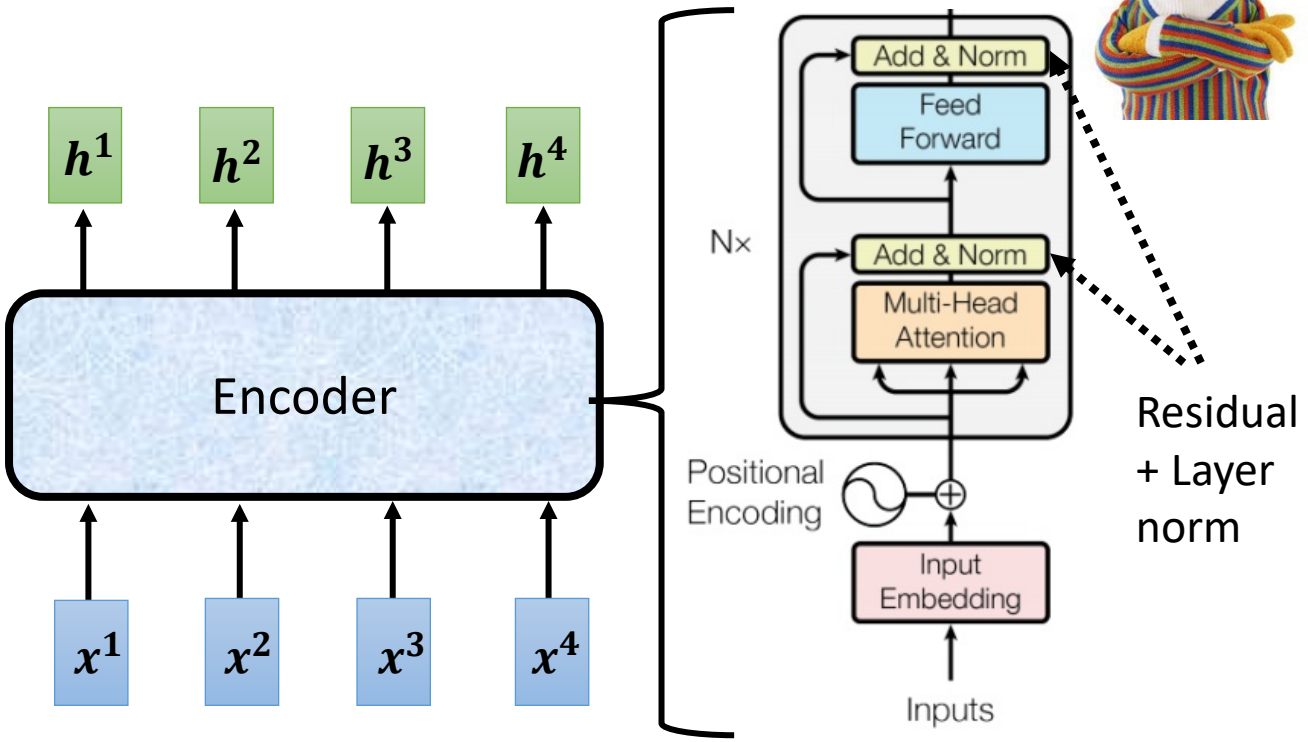



h^1
h^2
h^3
h^4
Block
Block
Block
x^1
x^2
x^3
x^4
FC
FC
FC
FC
Self-attention

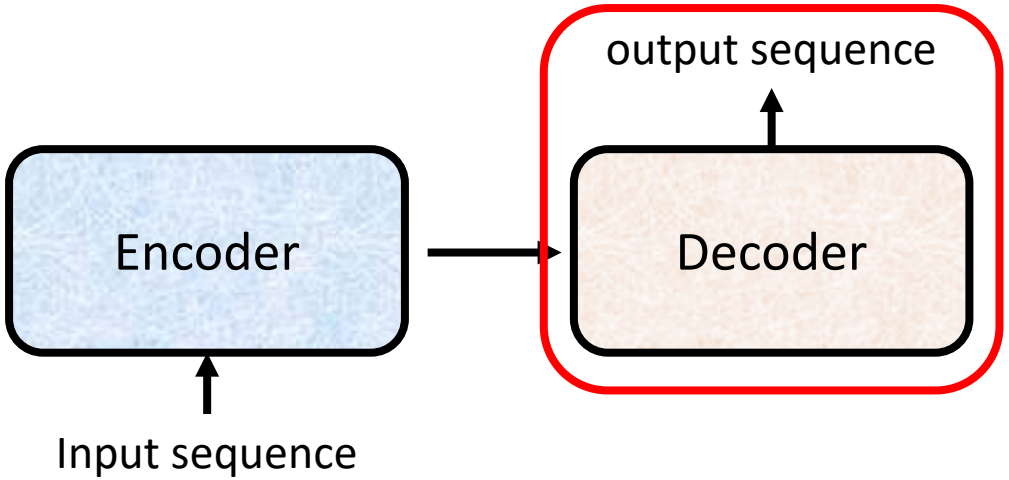

norm
residual
$a + b$
b
+
a
$x_1' \; x_2' \; \vdots \; x_K'$
$x_i' = \frac{x_i - m}{\sigma}$
norm
Layer Norm
https://arxiv.org/abs/1607.06450
$x_1 \; x_2 \; \vdots \; x_K$
mean m
standard deviation σ
norm
+
FC
Self-attention

BERT

I use the **same** network architecture as **transformer encoder**.

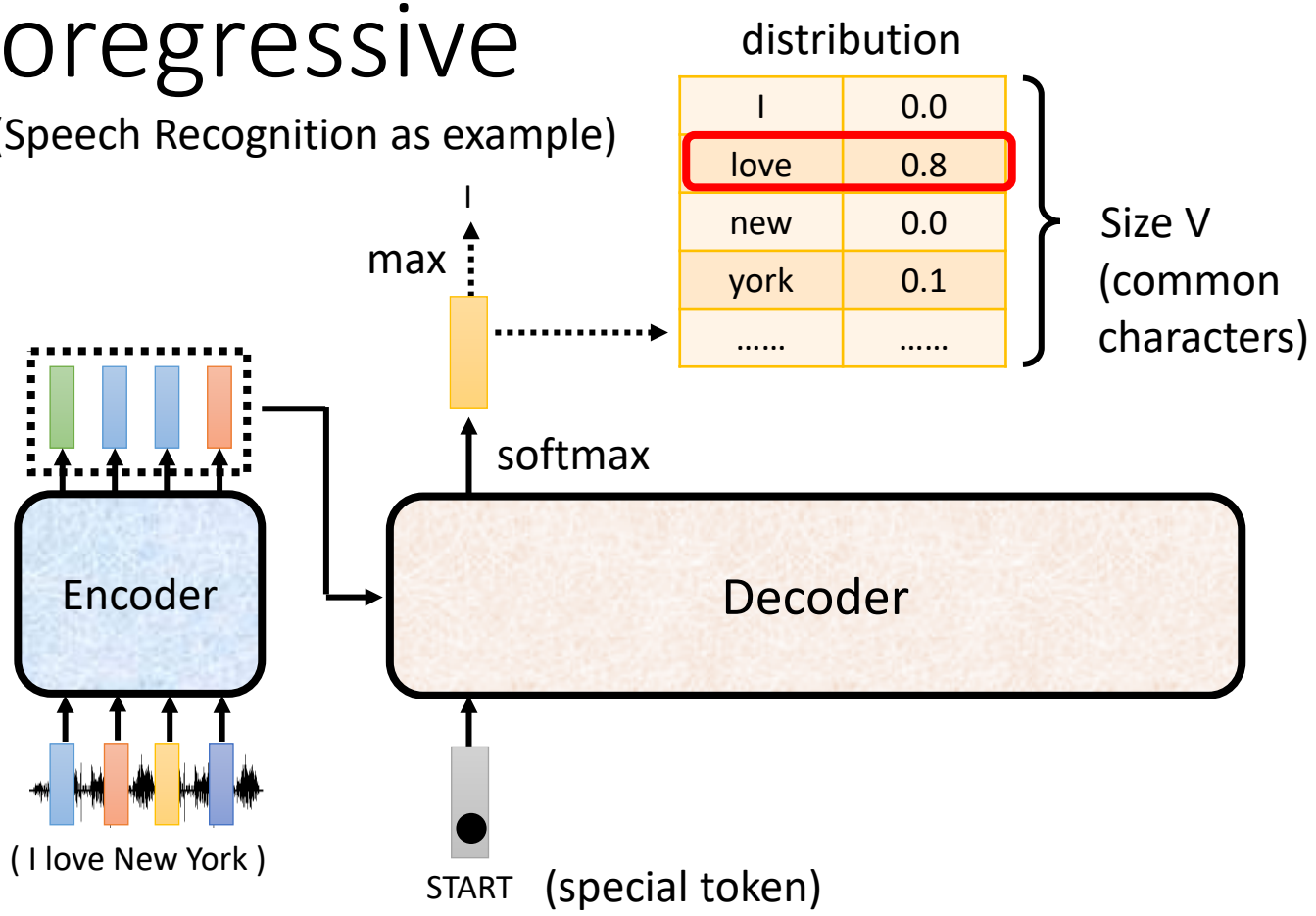


Decoder
– Autoregressive (AT)

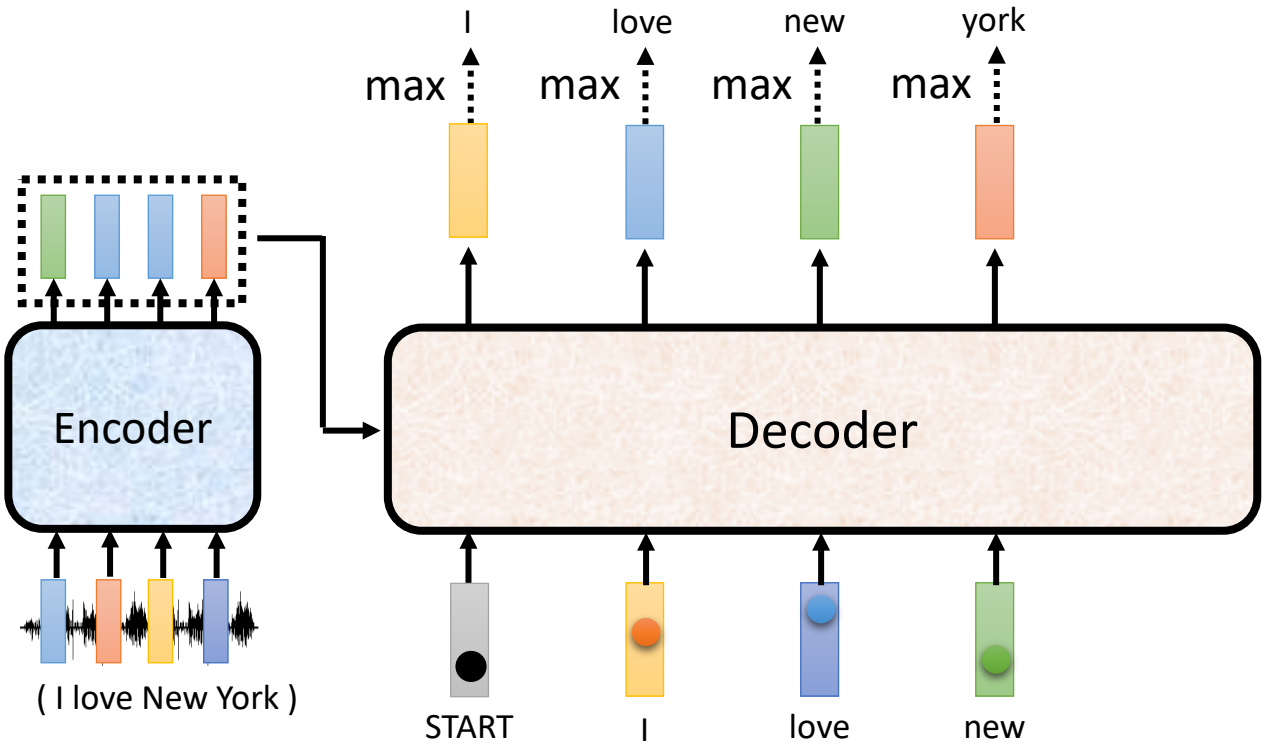


Autoregressive

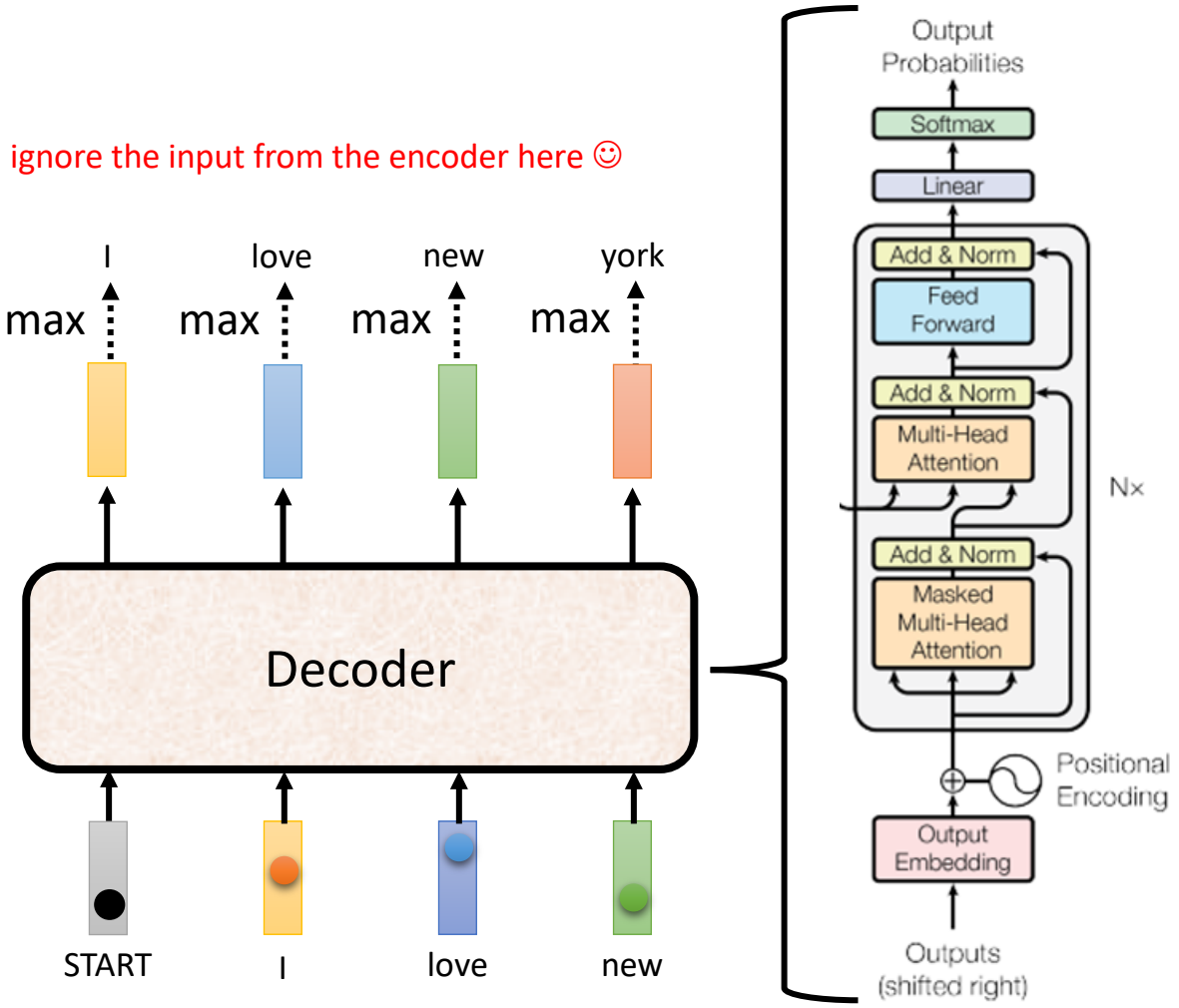
(Speech Recognition as example)

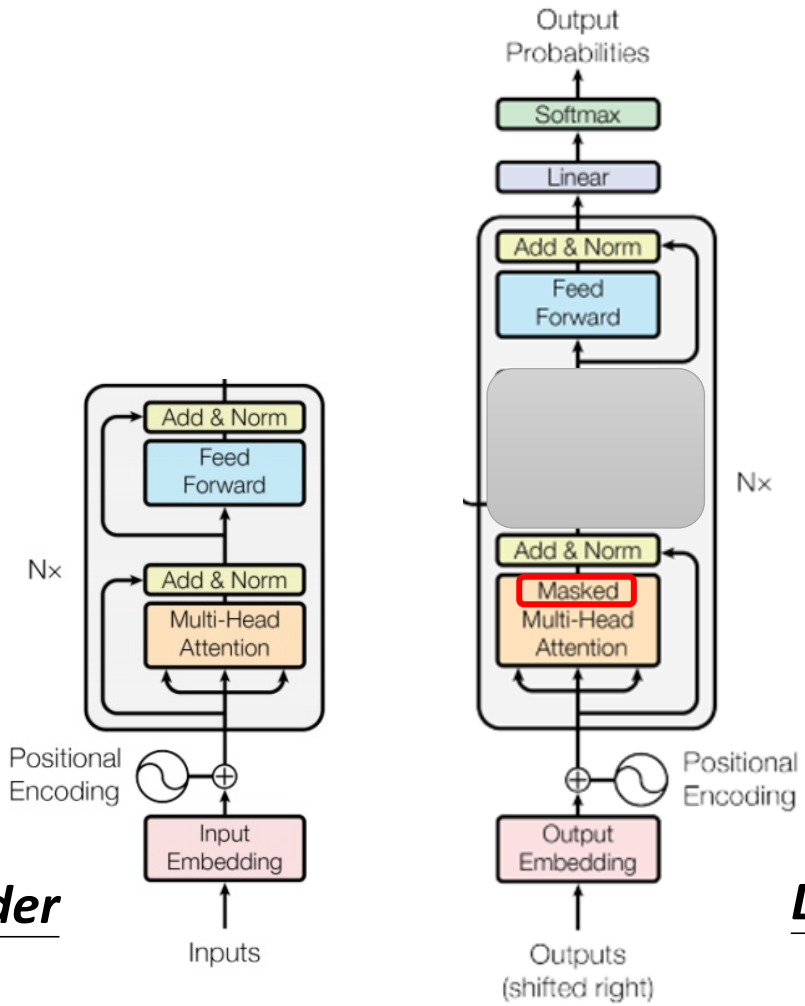


Autoregressive

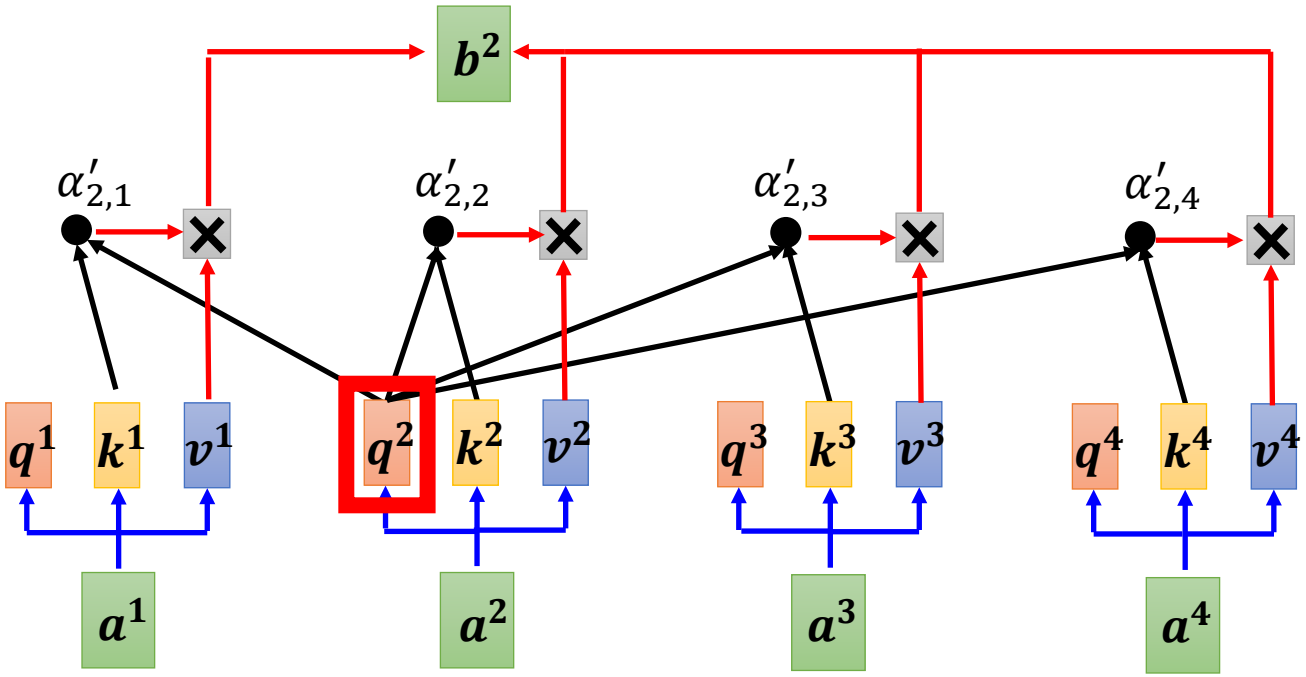


ignore the input from the encoder here ☺




Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Masked Multi-Head Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)
Encoder
Decoder

Self-attention → *Masked Self-attention*

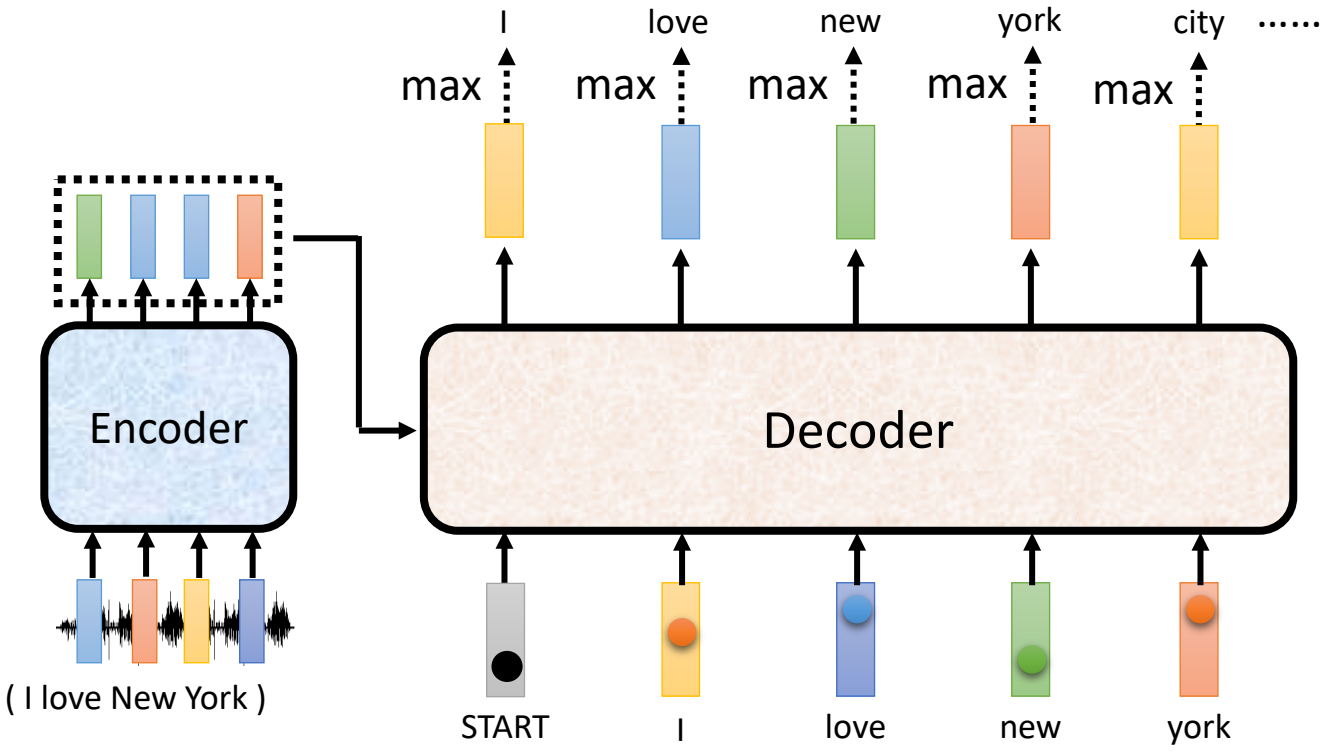


Why masked? Consider how does decoder work

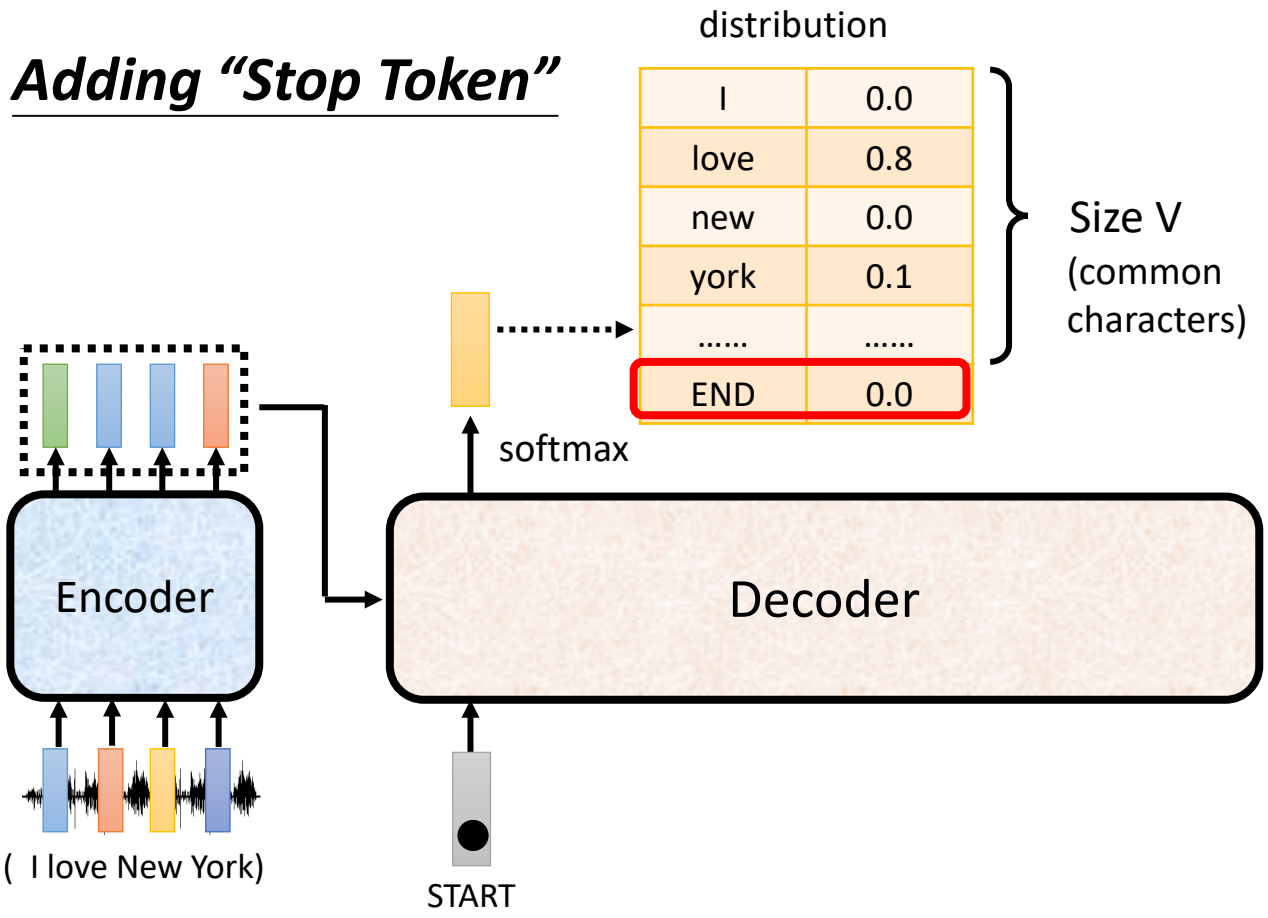
Autoregressive

We do not know the correct output length.

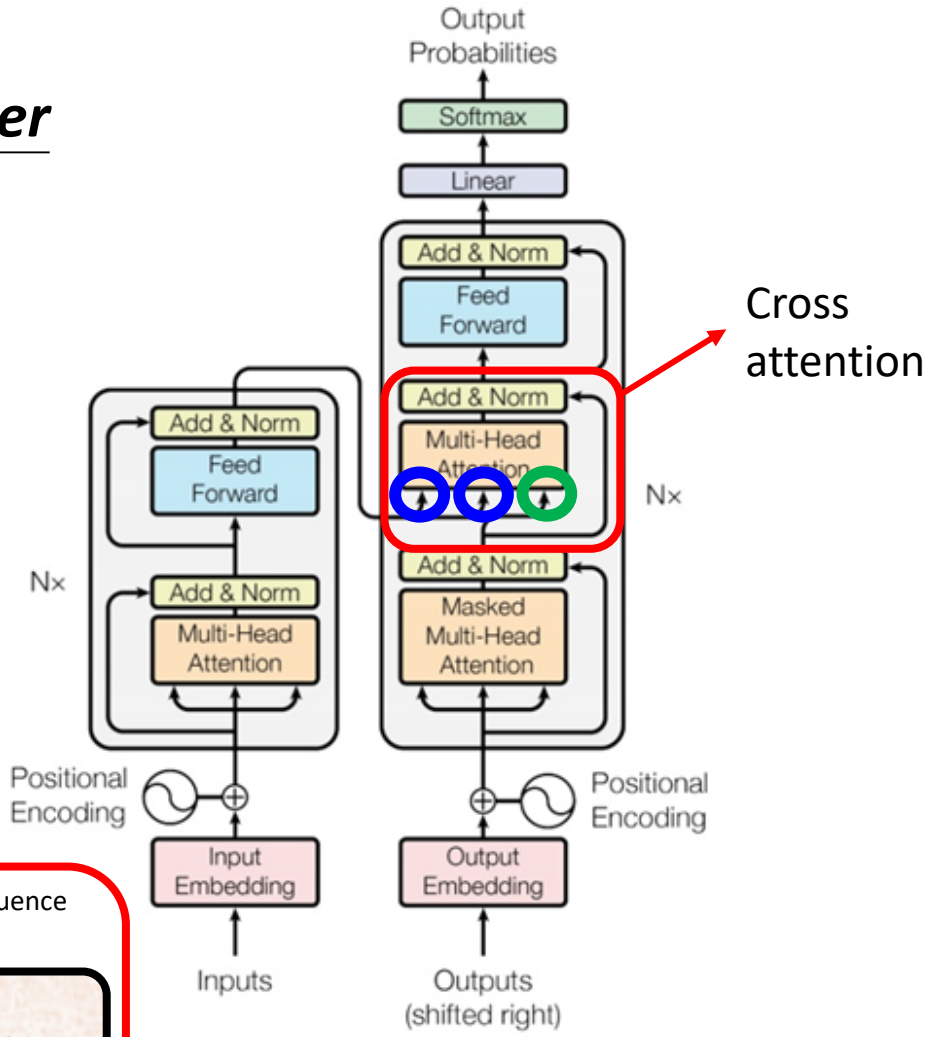
Never stop!



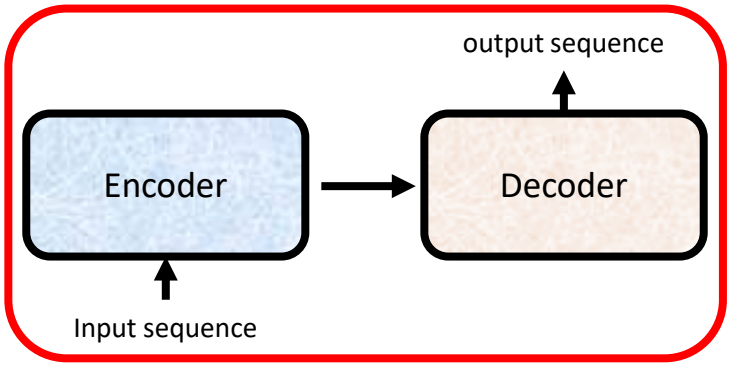
Adding "Stop Token"

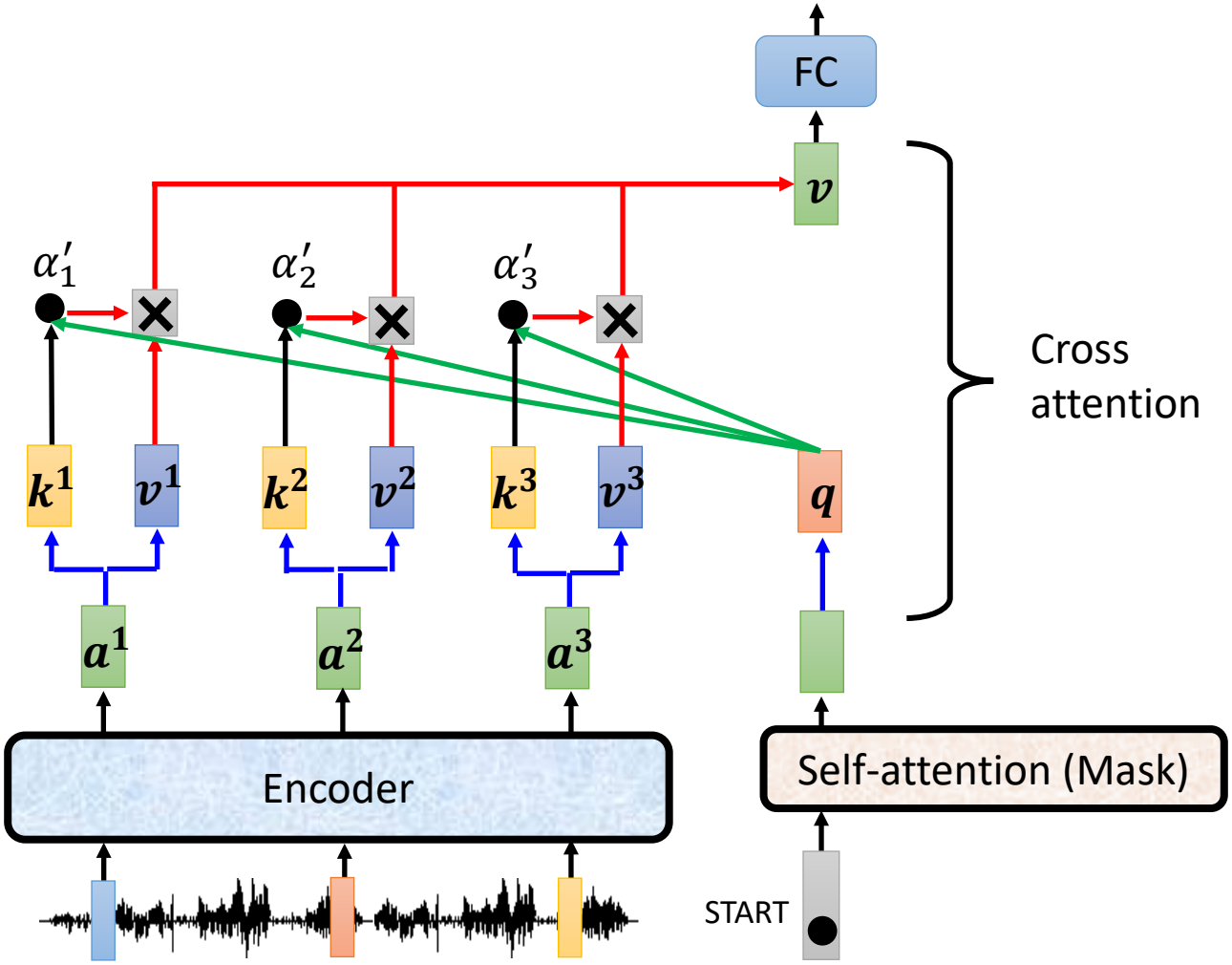


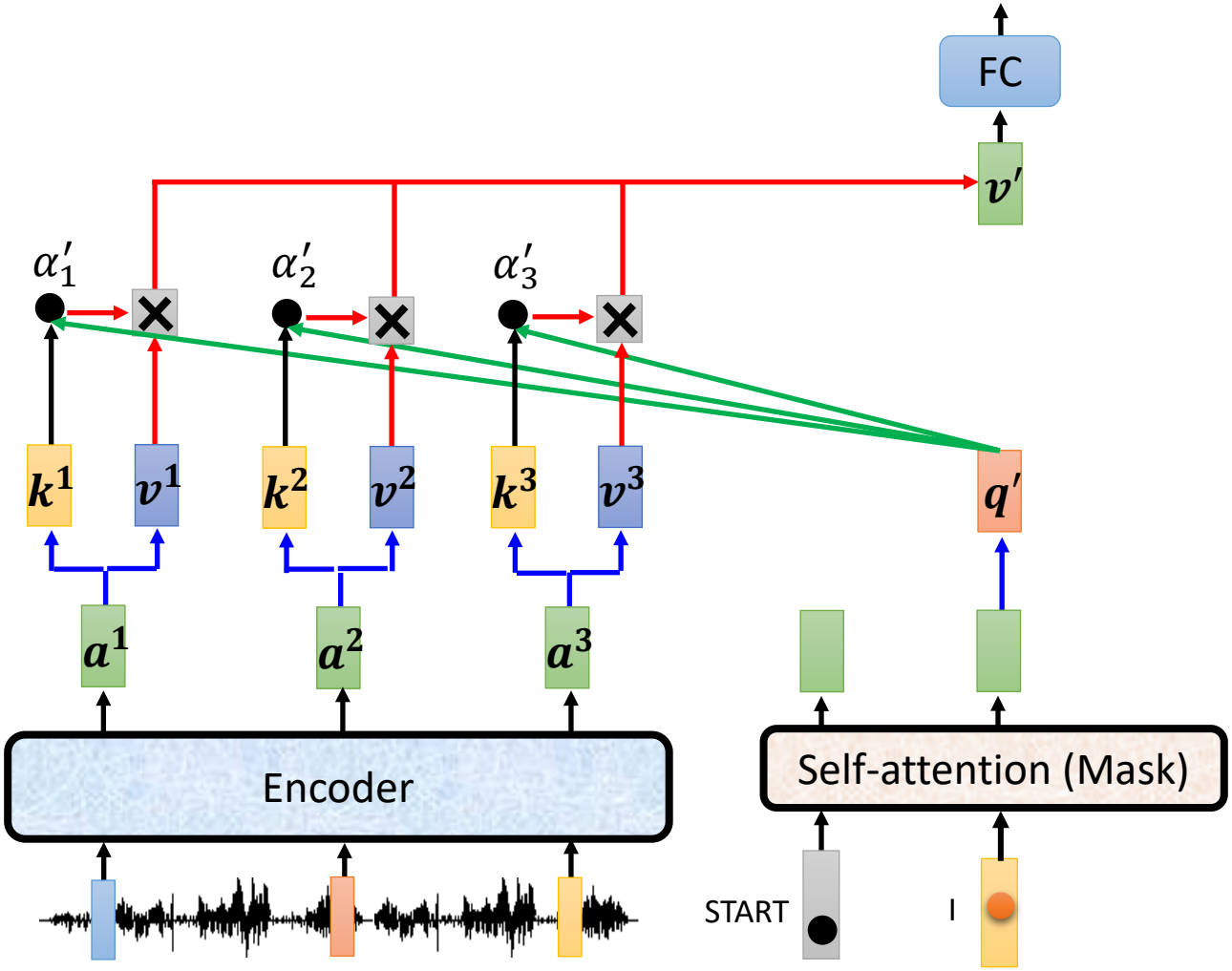
Transformer



Cross attention




FC
v
α'_1
α'_2
α'_3
Cross attention
k^1
v^1
k^2
v^2
k^3
v^3
q
a^1
a^2
a^3
Encoder
Self-attention (Mask)
START


FC
v'
α'_1
α'_2
α'_3
k^1
v^1
k^2
v^2
k^3
v^3
q'
a^1
a^2
a^3
Encoder
Self-attention (Mask)
START
I

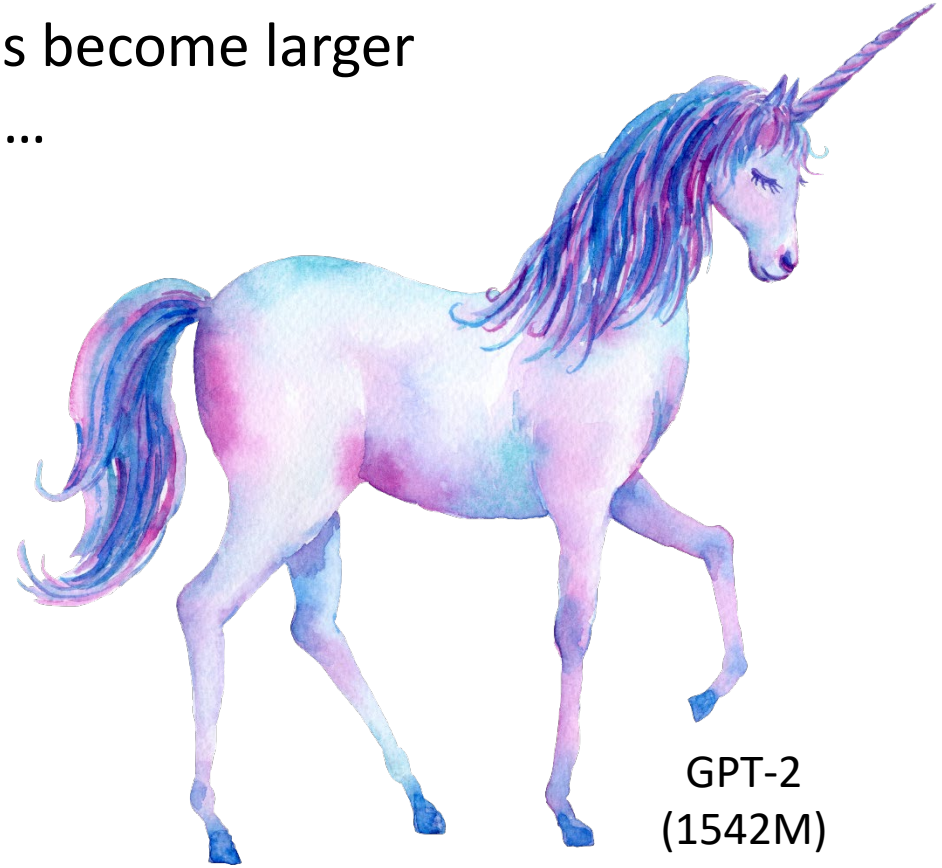
ERNIE (Enhanced Representation through Knowledge Integration)

Big Bird: Transformers for Longer Sequences

BERT (Bidirectional Encoder Representations from Transformers)

ELMo (Embeddings from Language Models)

The models become larger and larger …



ELMO (94M)

BERT (340M)

GPT-2 (1542M)

Source of image: https://huaban.com/pins/1714071707/

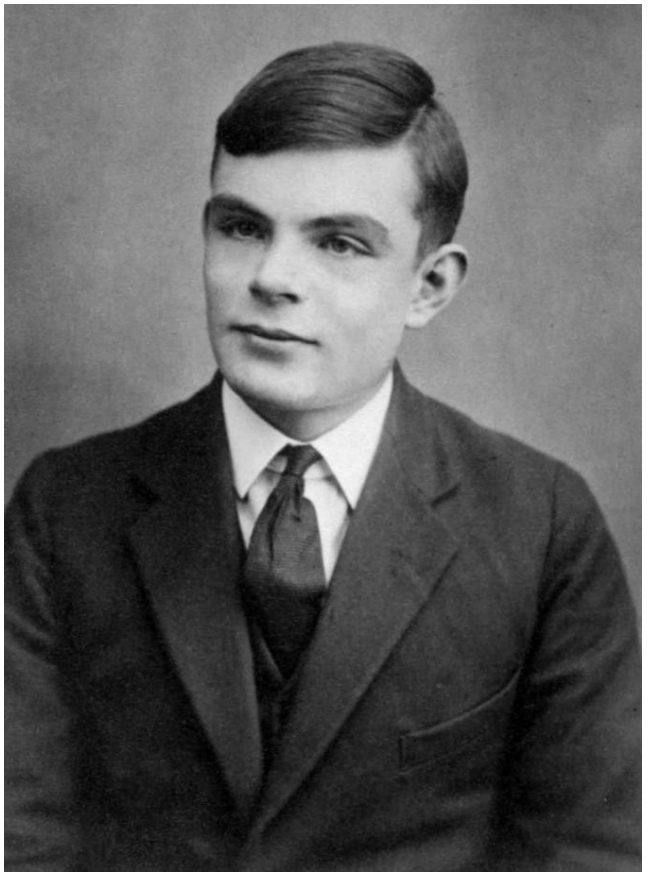
The models become larger and larger …

GPT-3 is **10** times larger than Turing NLG.

GPT-2

Megatron (8B)

Turing NLG (17B)

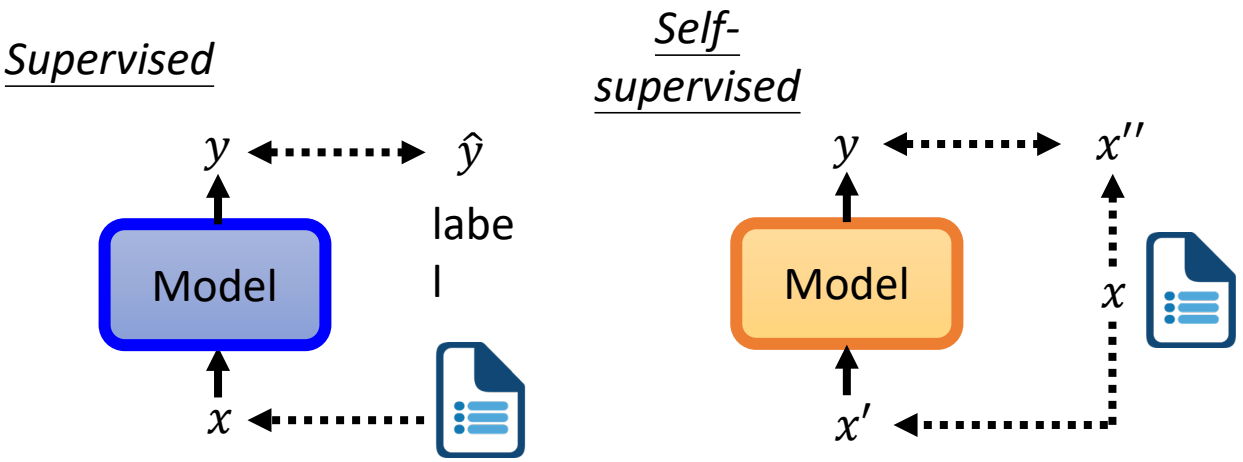


Outline

BERT series

GPT series

Self-supervised Learning



Yann LeCun
2019年4月30日 ·

I now call it "self-supervised learning", because "unsupervised" is both a loaded and confusing term.

In self-supervised learning, the system learns to predict part of its input from other parts of it input. In other words a portion of the input is used as a supervisory signal to a predictor fed with the remaining portion of the input.

Self-supervised Learning

Pre-train

BERT

- Masked token prediction
- Next sentence prediction

Fine-tune

Model for Task 1

Model for Task 2

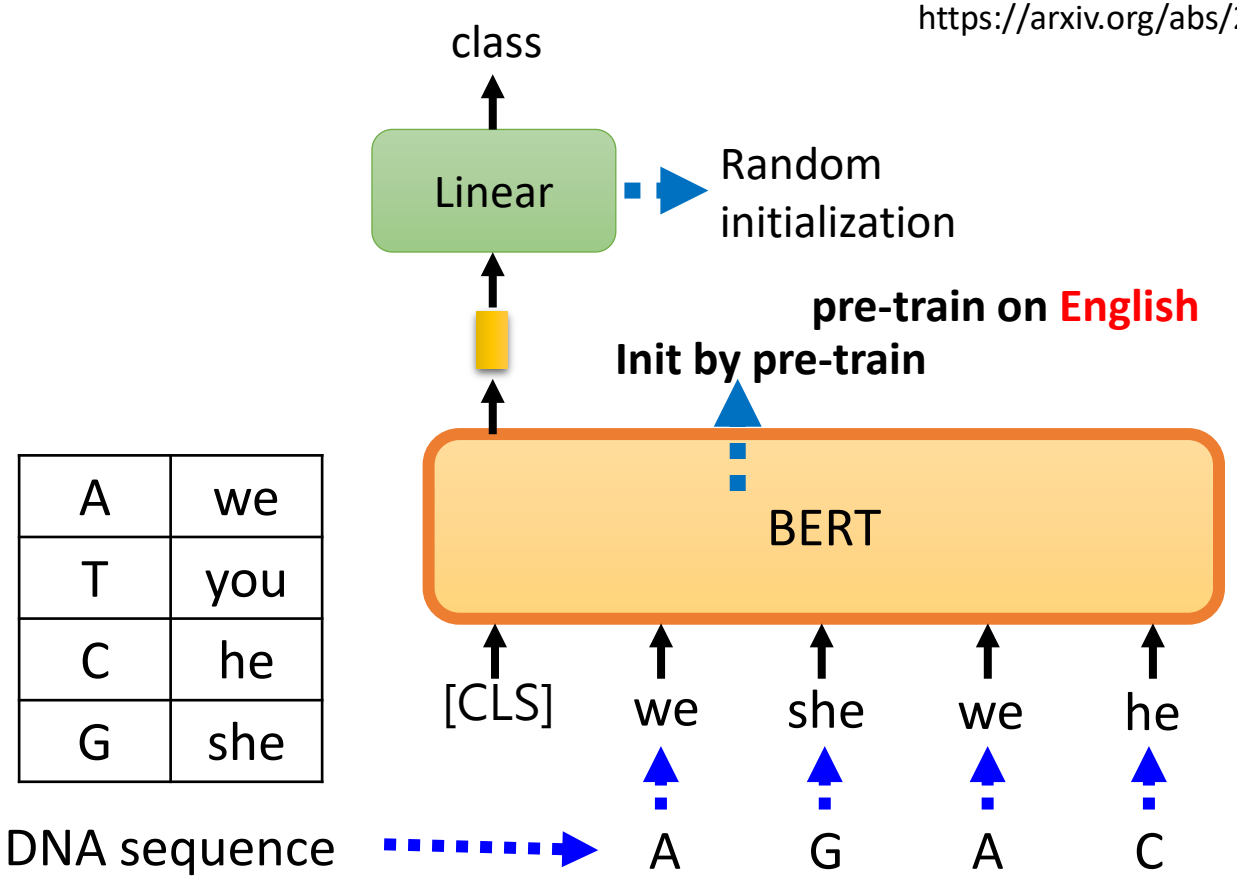
Model for Task 3

Downstream Tasks

- The tasks we care
- We have a little bit labeled data.

Why does BERT work?

https://arxiv.org/abs/2103.07162

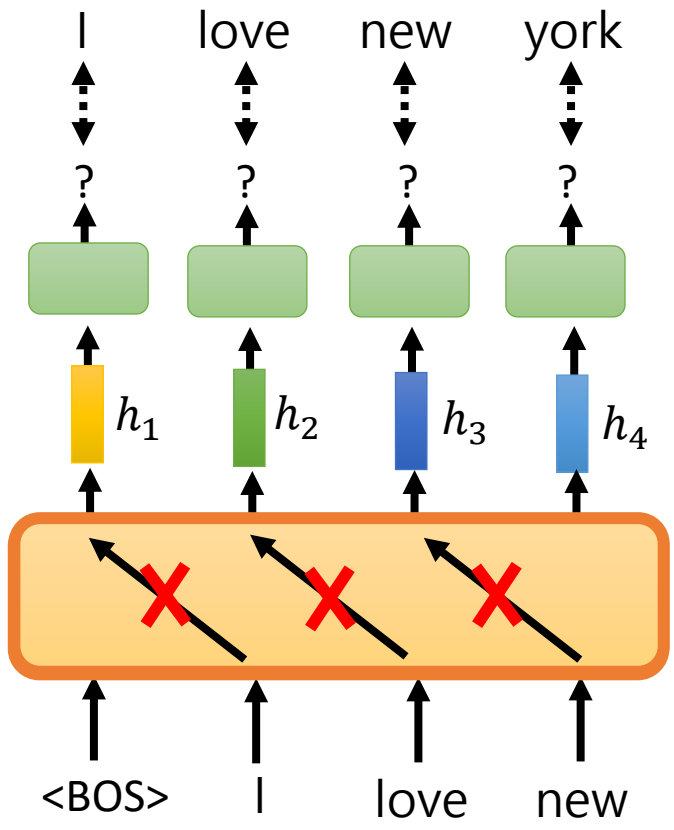


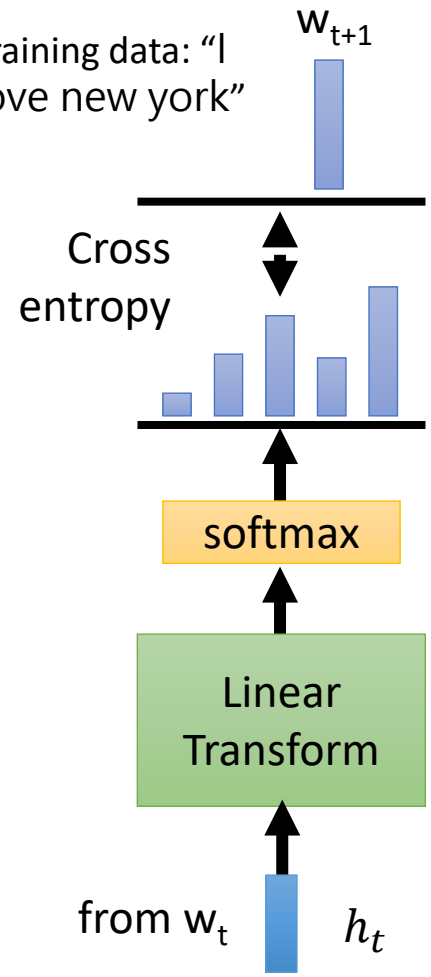
Outline

BERT series

GPT series

Predict Next Token





“In-context” Learning

Few-shot Learning

(no gradient descent)

```
Translate English to French:        ← task description
sea otter => loutre de mer          ← examples
peppermint => menthe poivrée        ←
plush girafe => girafe peluche      ←
cheese =>                           ← prompt
```

One-shot Learning

```
Translate English to French:        ← task description
sea otter => loutre de mer          ← example
cheese =>                           ← prompt
```

Zero-shot Learning

```
Translate English to French:        ← task description
cheese =>                           ← prompt
```

GPT-3

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*

Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry

Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan

Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter

Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray

Benjamin Chess Jack Clark Christopher Berner

Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI