Transformer

Attention Is All You Need

Seyedi

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

† Work performed while at Google Brain.
‡ Work performed while at Google Research.

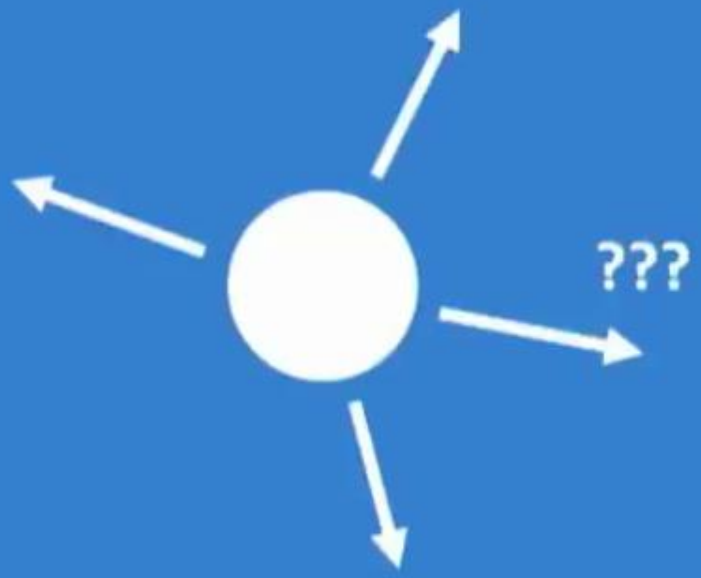
31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Sequence

Given an image of a ball,
can you predict where it will go next?

Sequence

Given an image of a ball,
can you predict where it will go next?



Sequence

Given an image of a ball,
can you predict where it will go next?

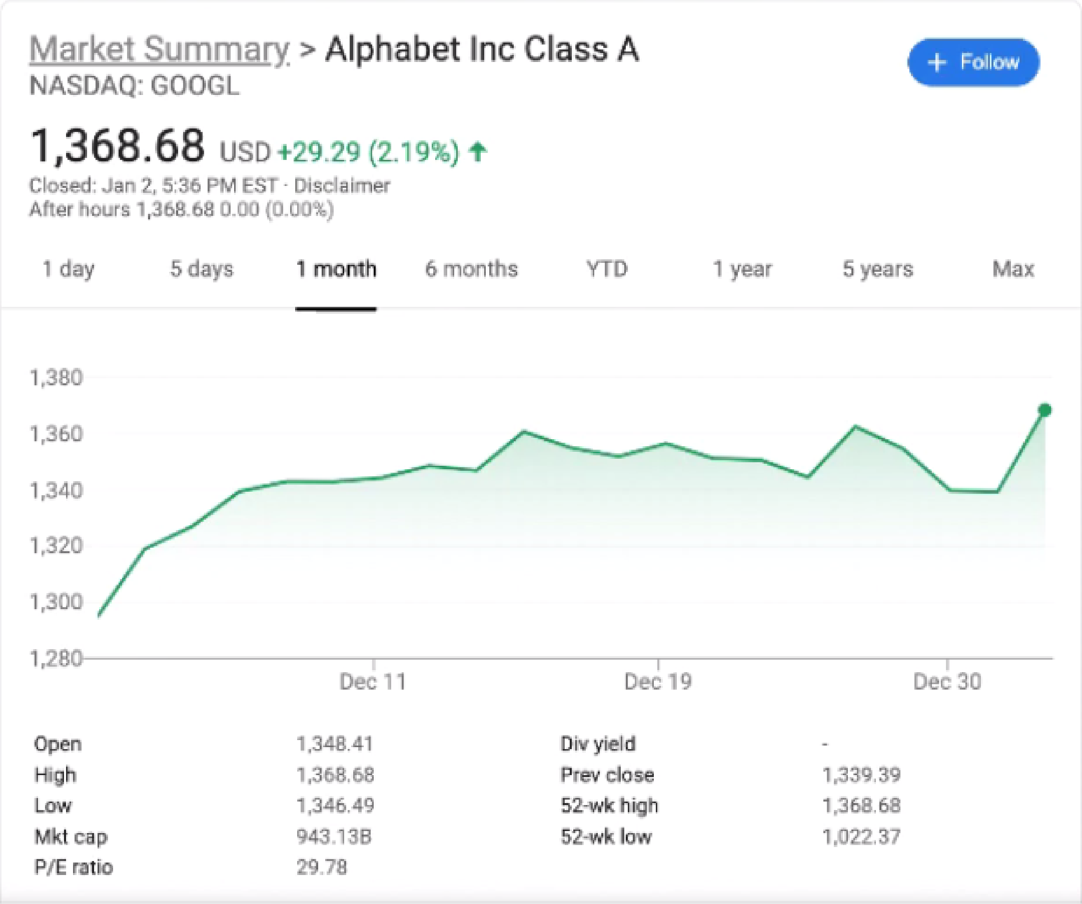
Sequence

Given an image of a ball,
can you predict where it will go next?

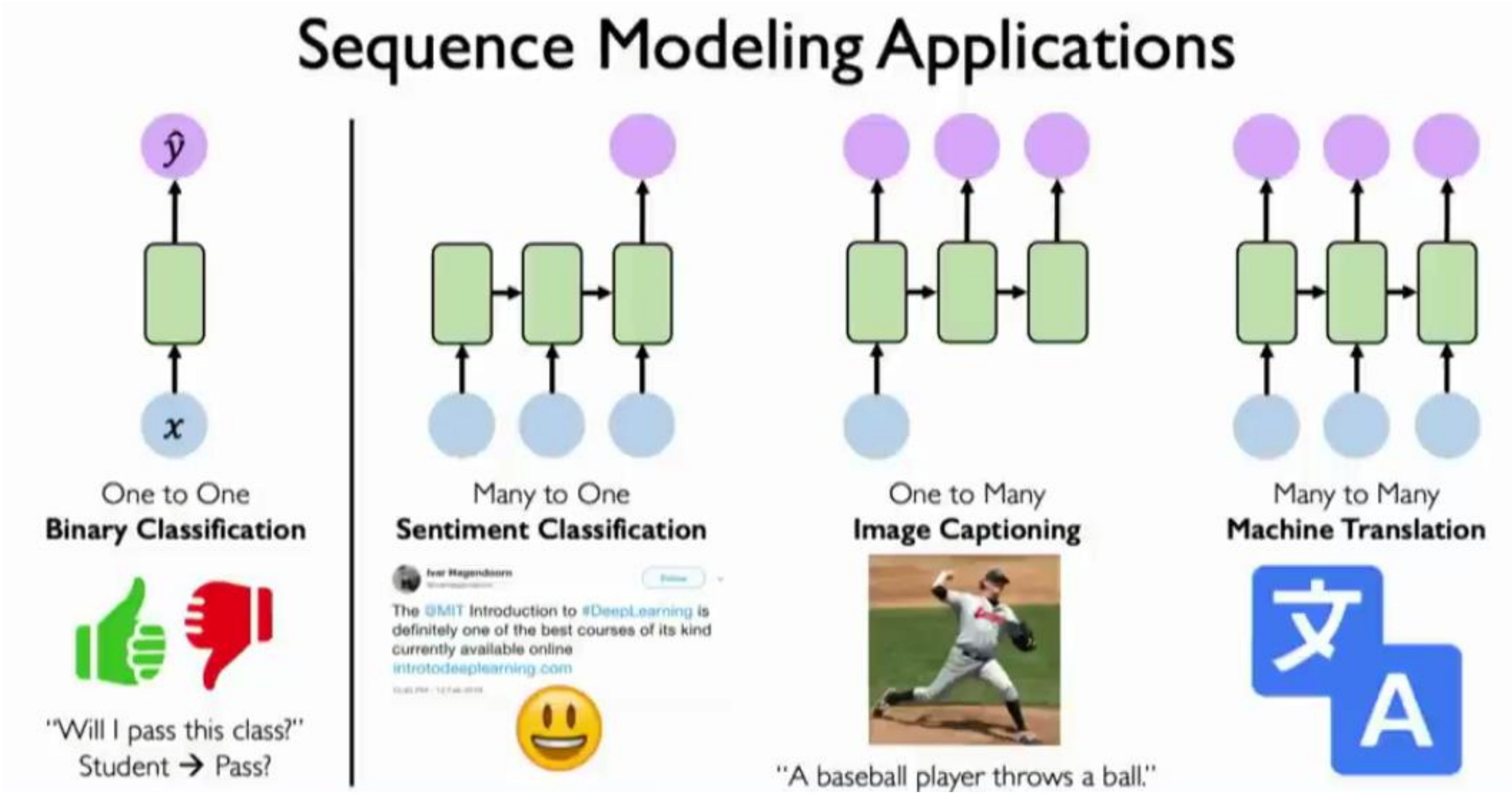
Sequence

The unicorn is scotland's national animal

Sequences

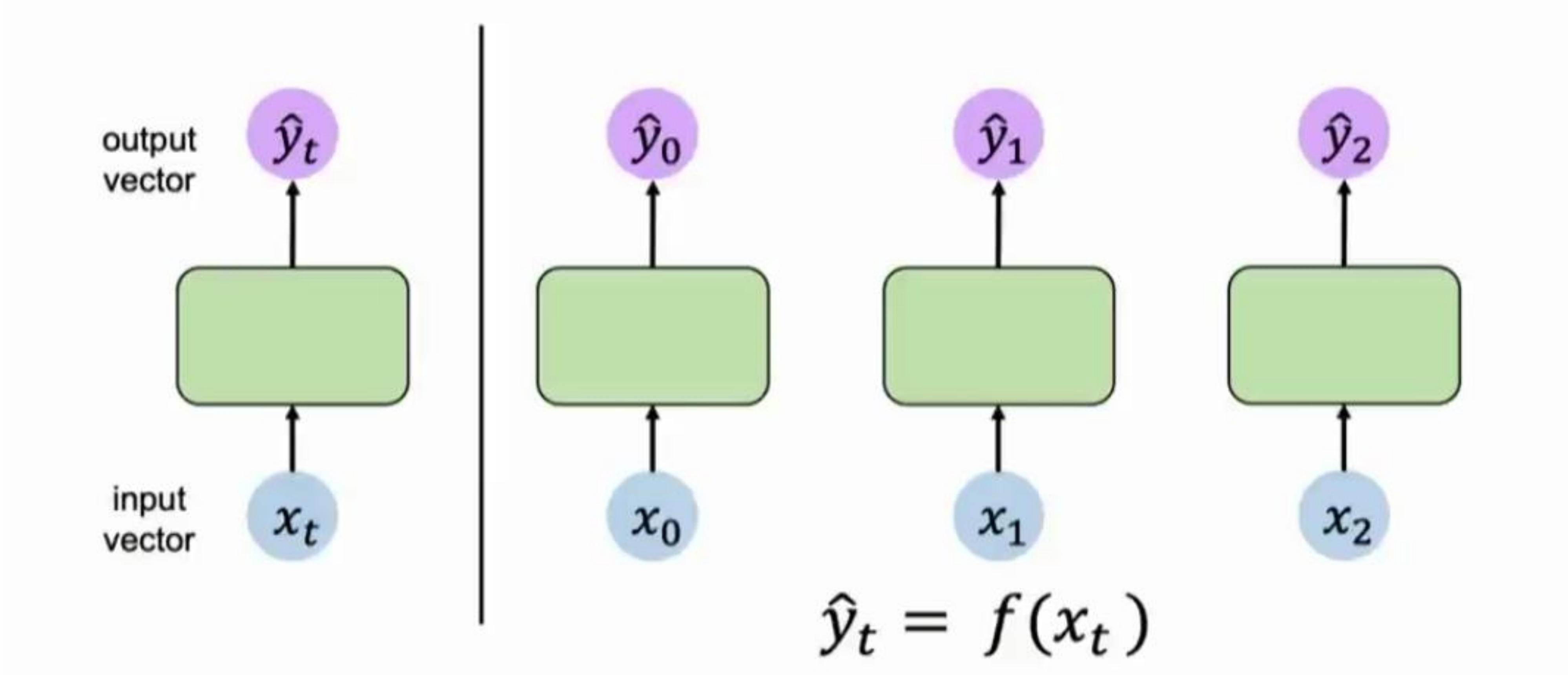


Sequence

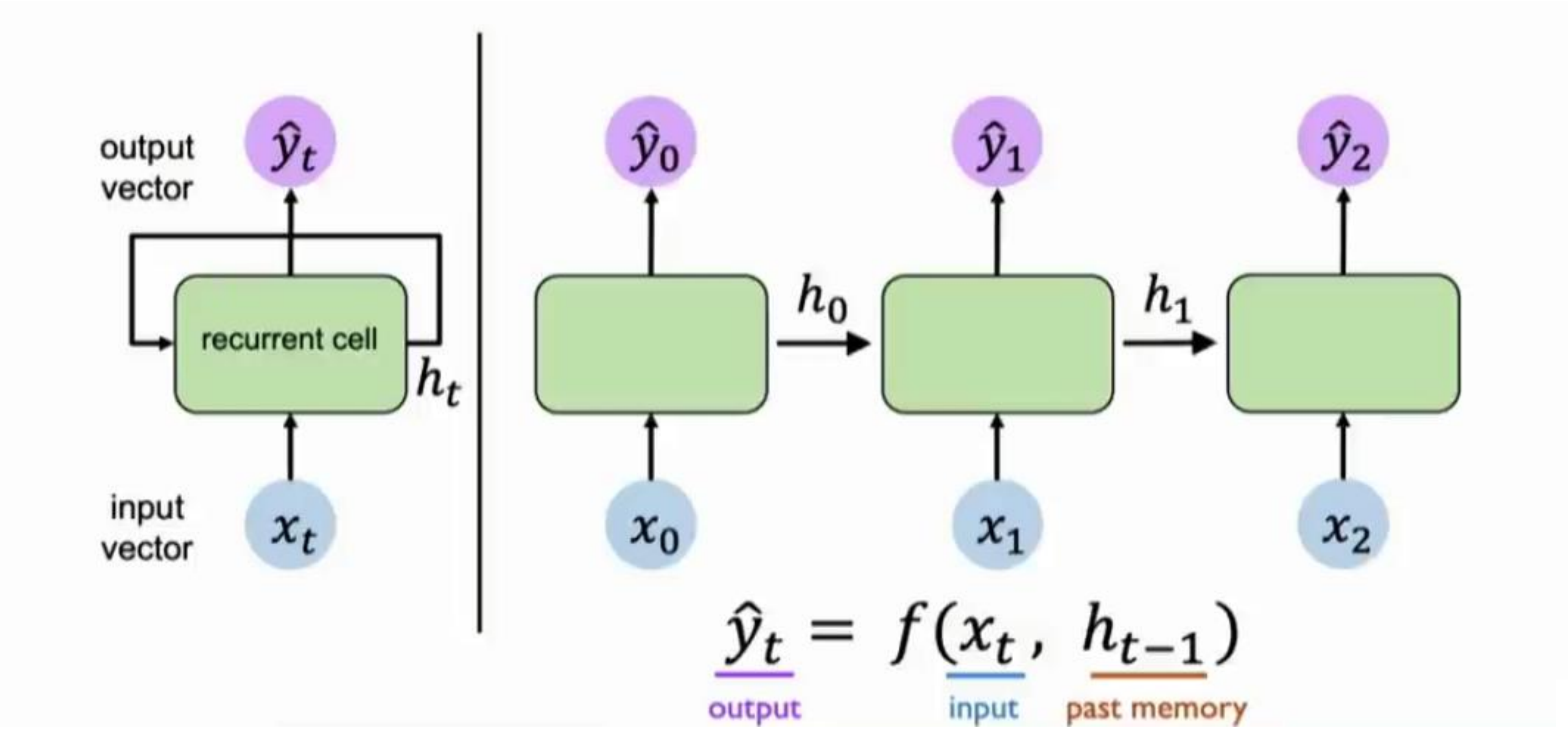


Sequence Modeling Applications
$\hat{y}$
x
One to One
Binary Classification
"Will I pass this class?"
Student → Pass?
Many to One
Sentiment Classification
The @MIT Introduction to #DeepLearning is definitely one of the best courses of its kind currently available online
introtodeeplearning.com
One to Many
Image Captioning
"A baseball player throws a ball."
Many to Many
Machine Translation
文
A

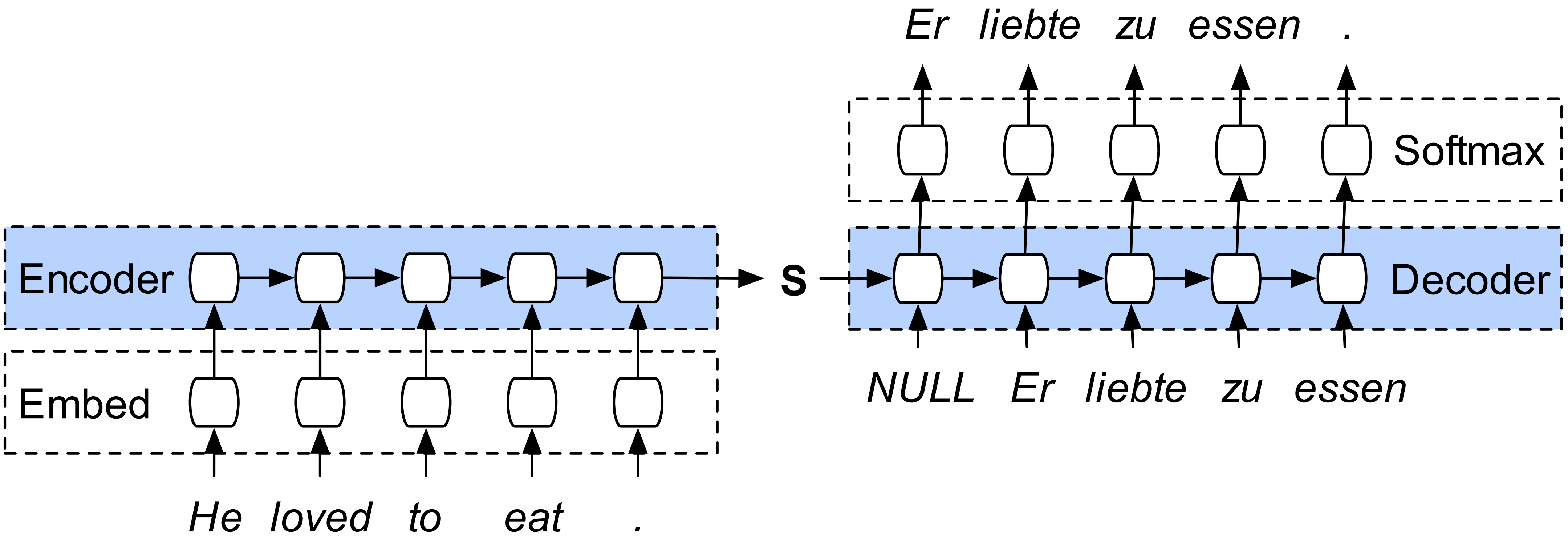
Handling Individual Time Steps



Neurons with Recurrence



Recurrent Neural Networks



Recurrent Neural Networks

Recurrent Neural Networks has a short reference window

As aliens entered our planet and began to colonize earth a certain group of extraterrestrials ...

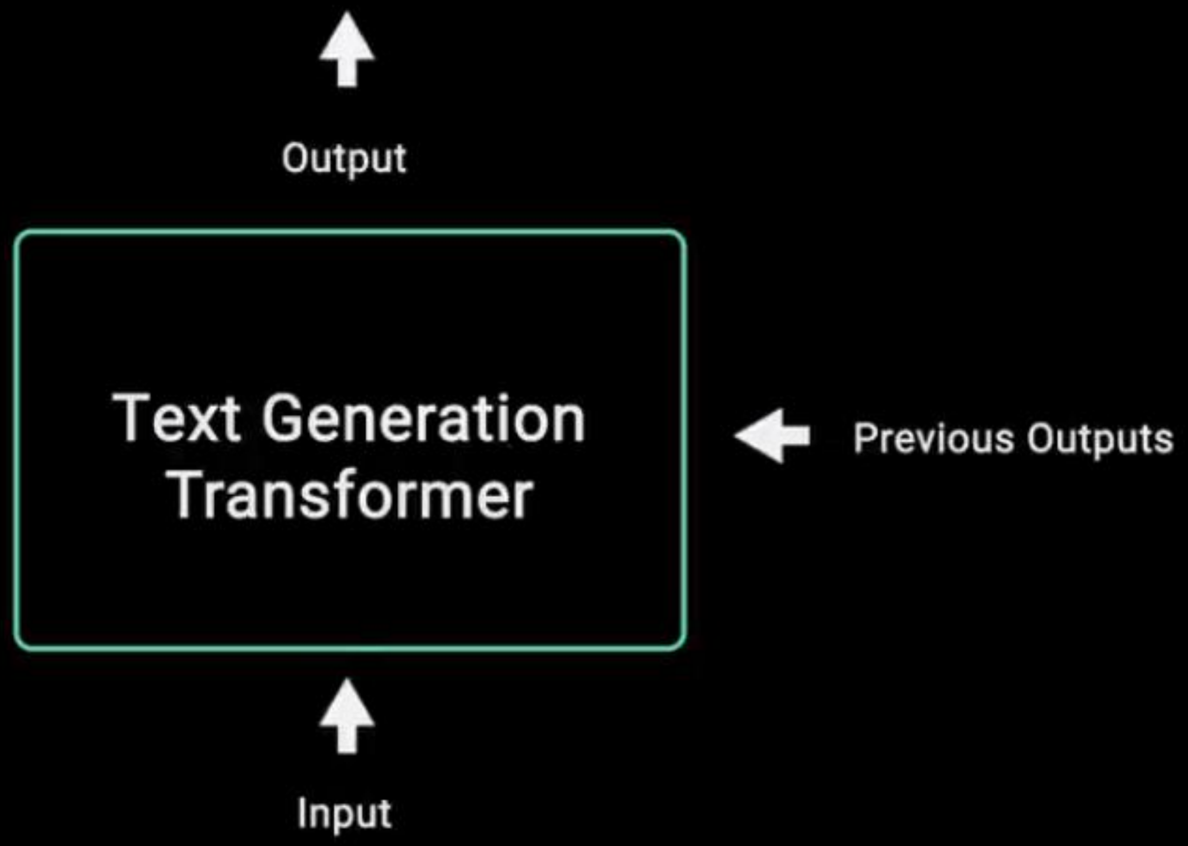
GRUs and LSTMs

GRU’s and LSTM’s have a longer reference window than RNN’s

As aliens entered our planet

and began to colonize earth a certain group of extraterrestrials ...

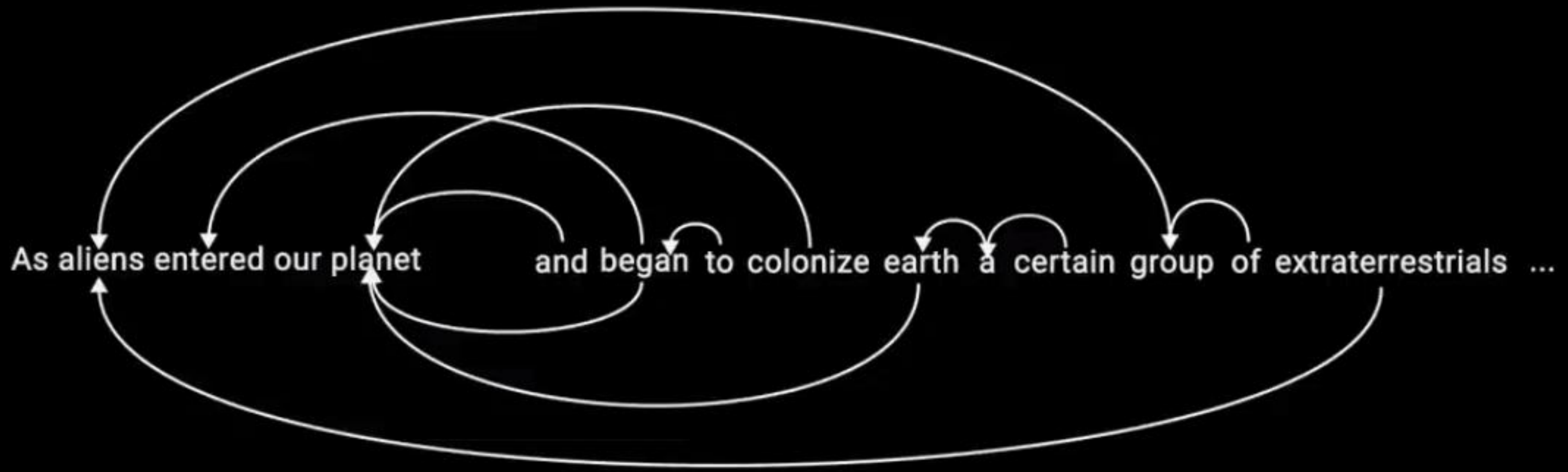
Text Generation and Attention



Text Generation and Attention

Text Generation
Transformer

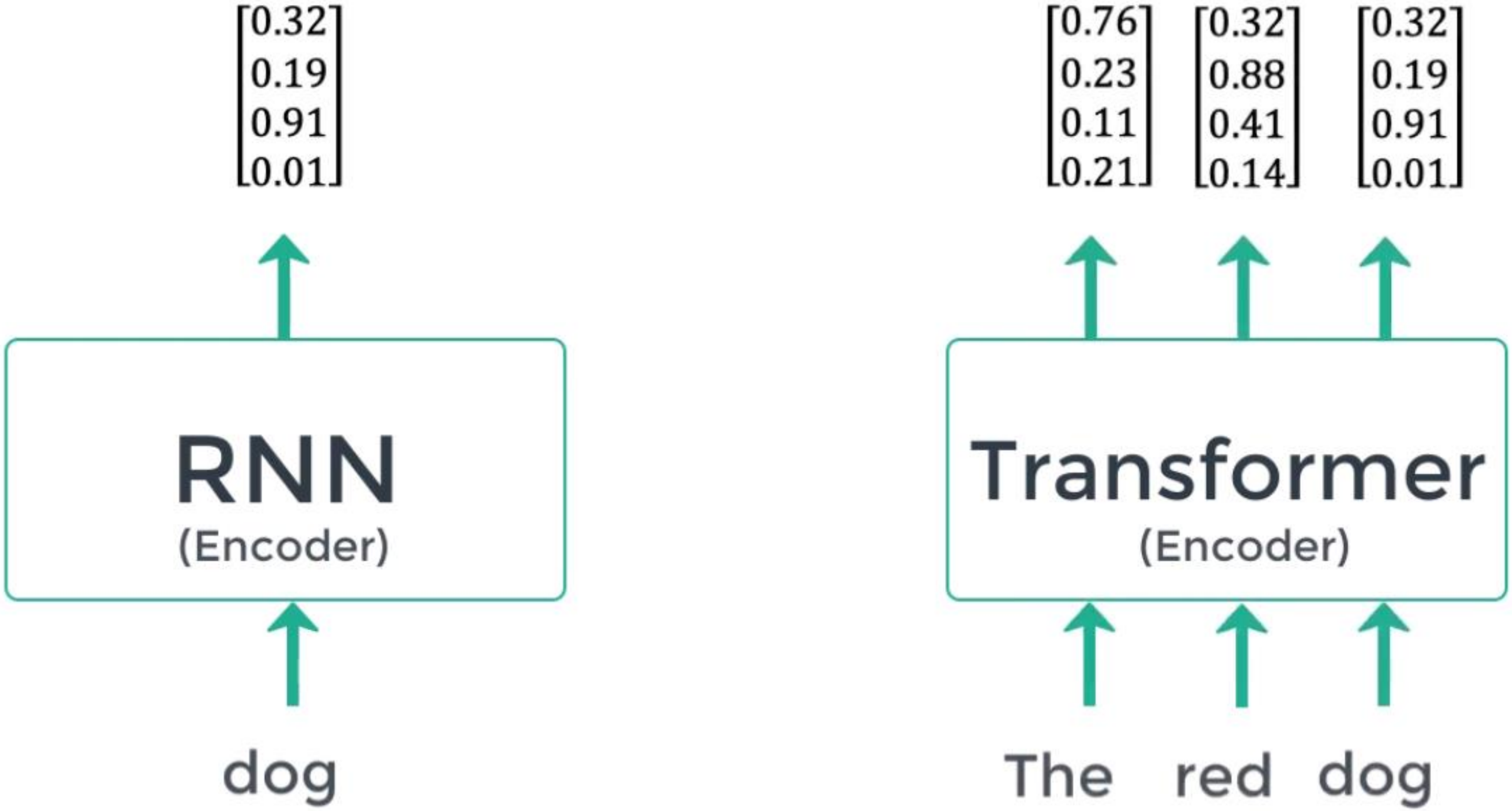
Text Generation and Attention



Attention Mechanism has an infitnite reference window

As aliens entered our planet and began to colonize earth a certain group of extraterrestrials ...

RNNs vs Transformers



RNNs vs Transformers

<u>Challenges with RNNs</u>

- Long range dependencies
- Gradient vanishing and explosion
- Large # of training steps
- Recurrence prevents parallel computation

<u>Transformer Networks</u>

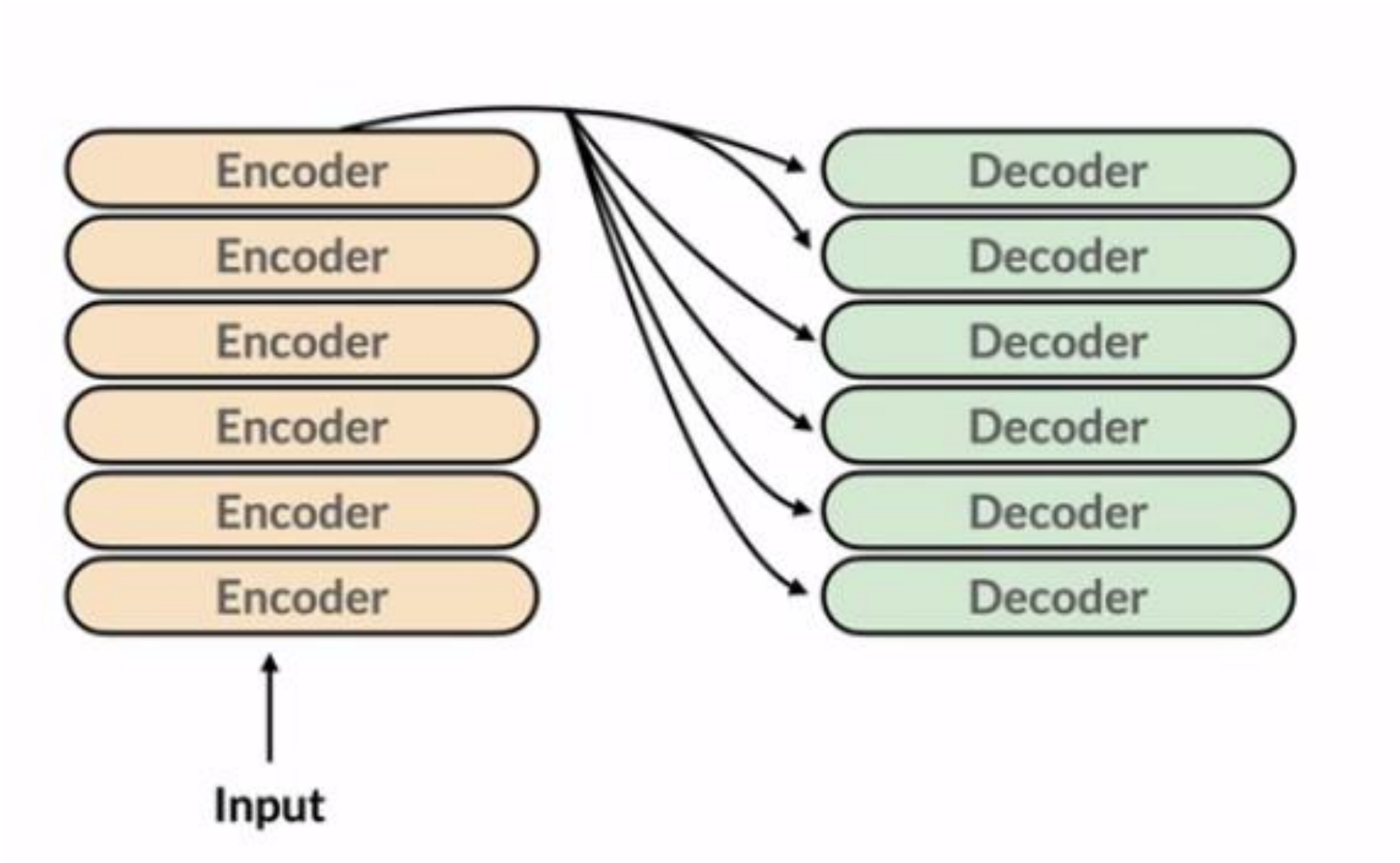
- Facilitate long range dependencies
- No gradient vanishing and explosion
- Fewer training steps
- No recurrence that facilitate parallel computation

Transformers Encoder

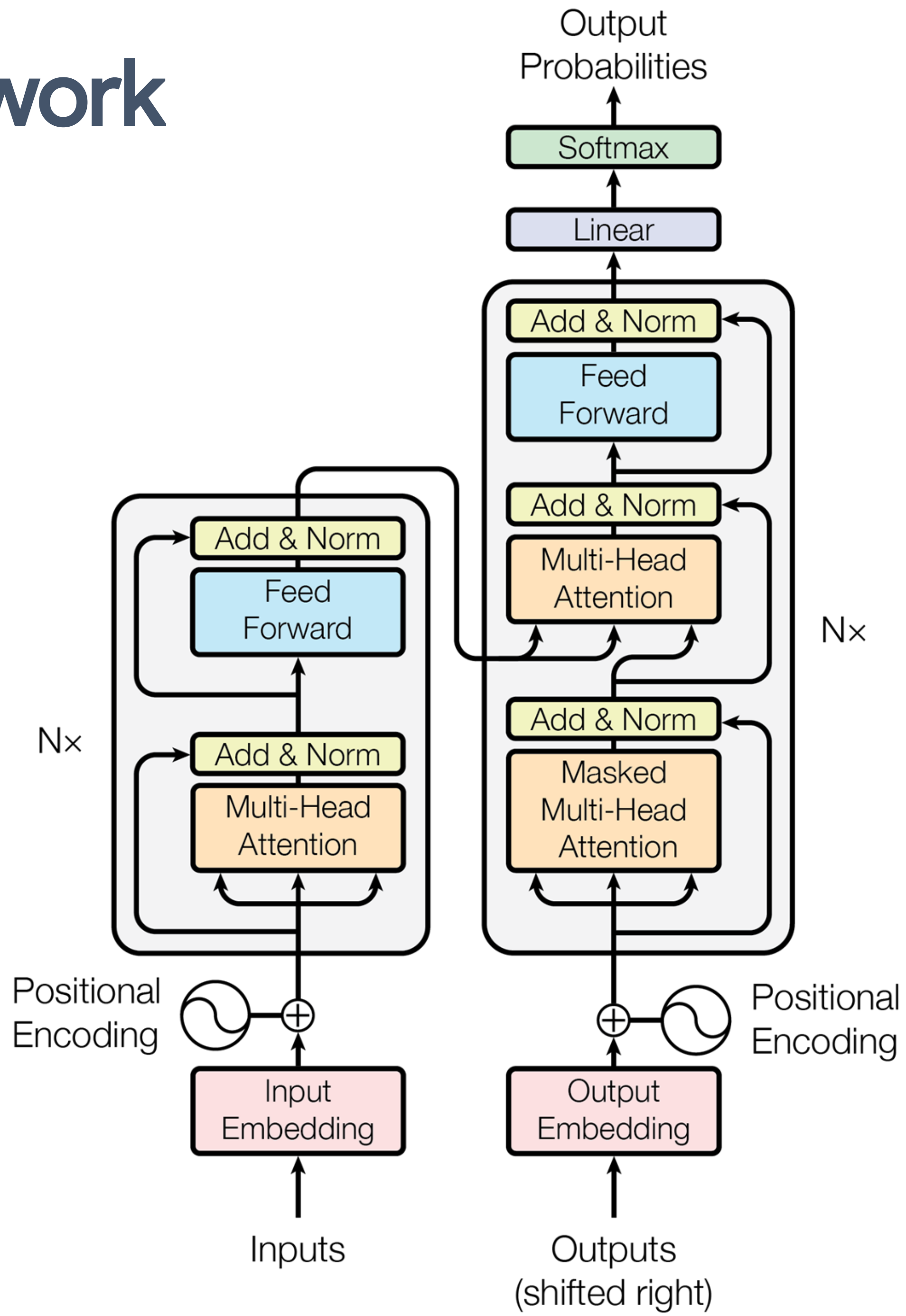
Transformers Decoder

Transformer Network

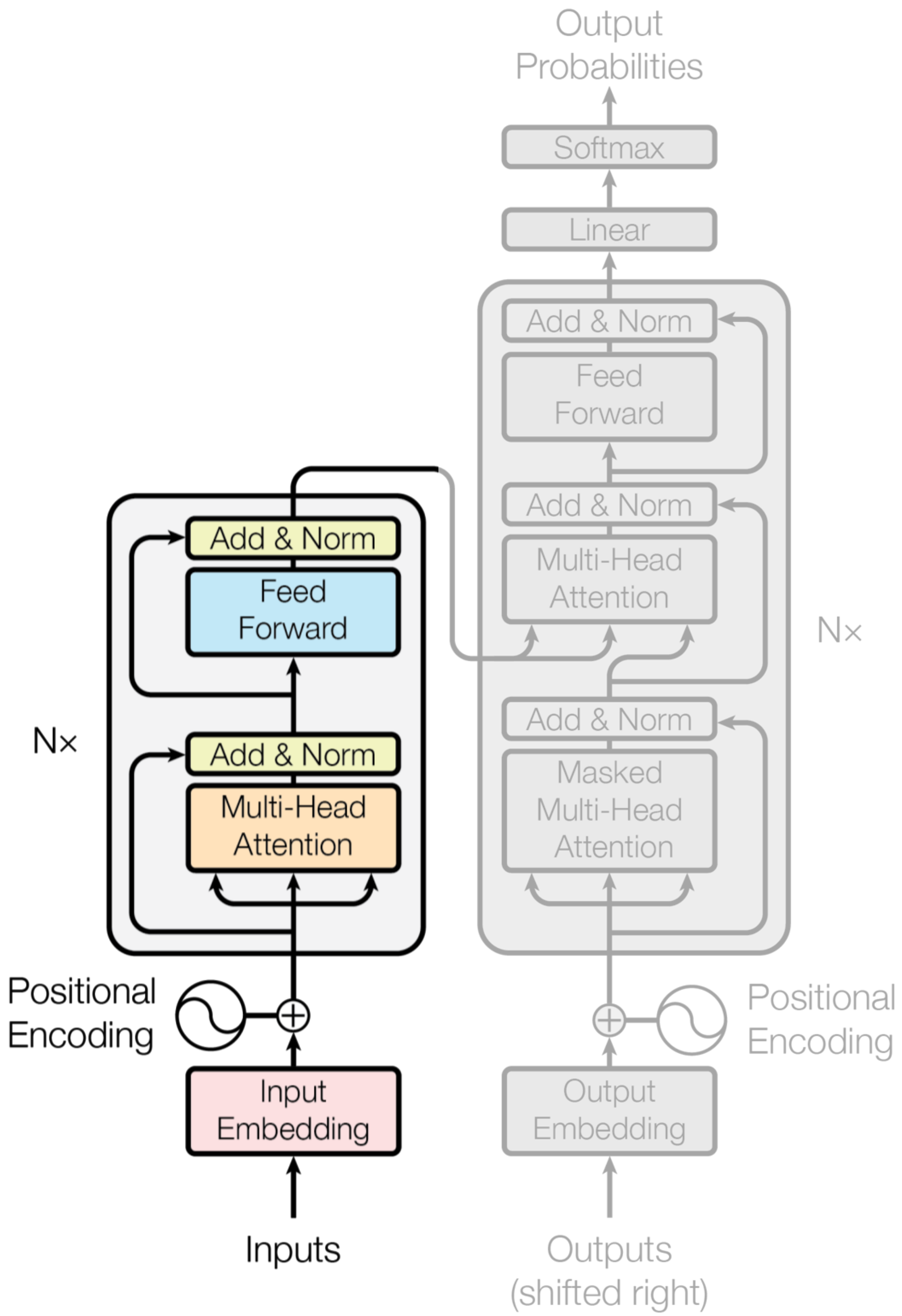
Transformer Network



Transformer Network

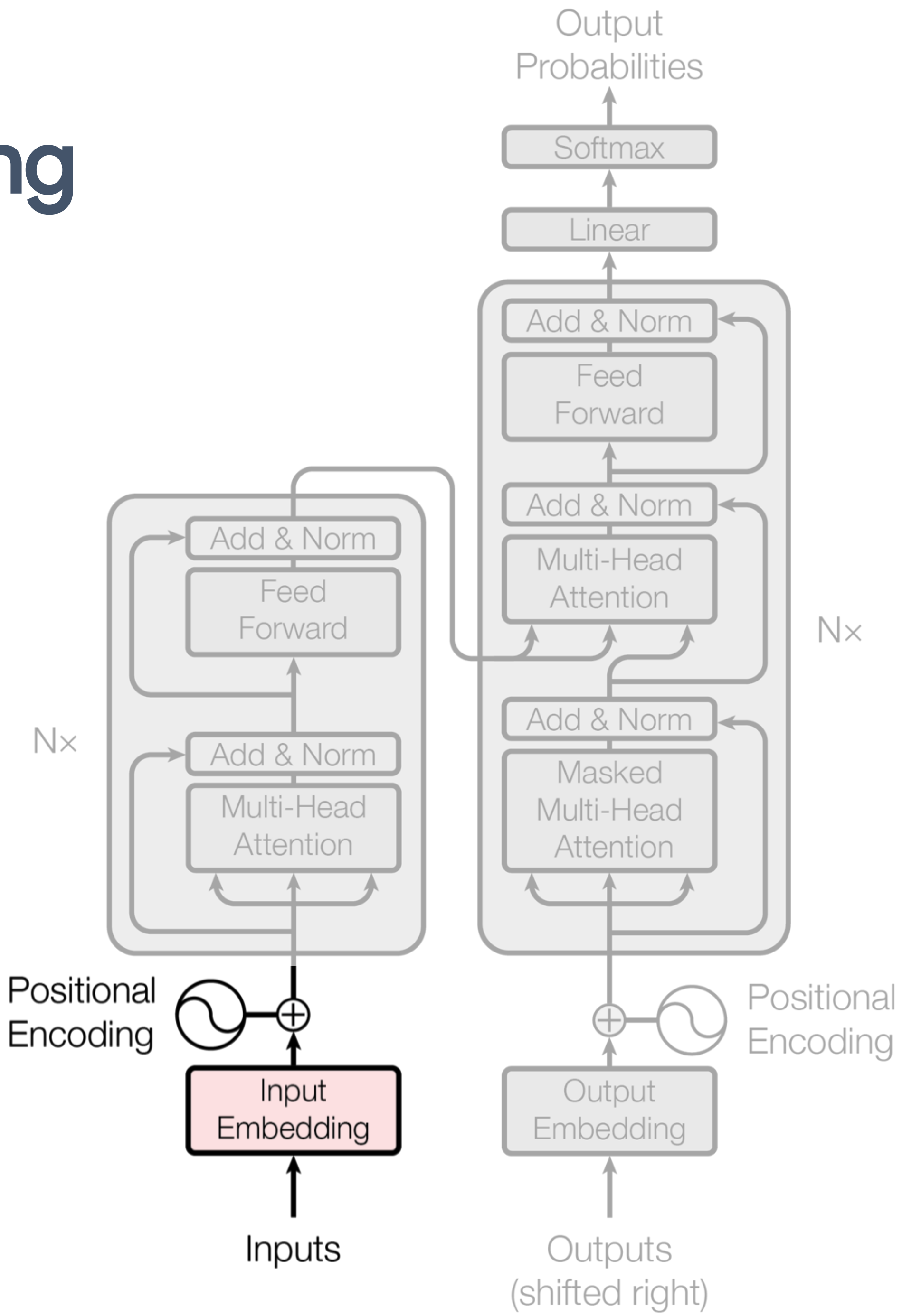


Encoder



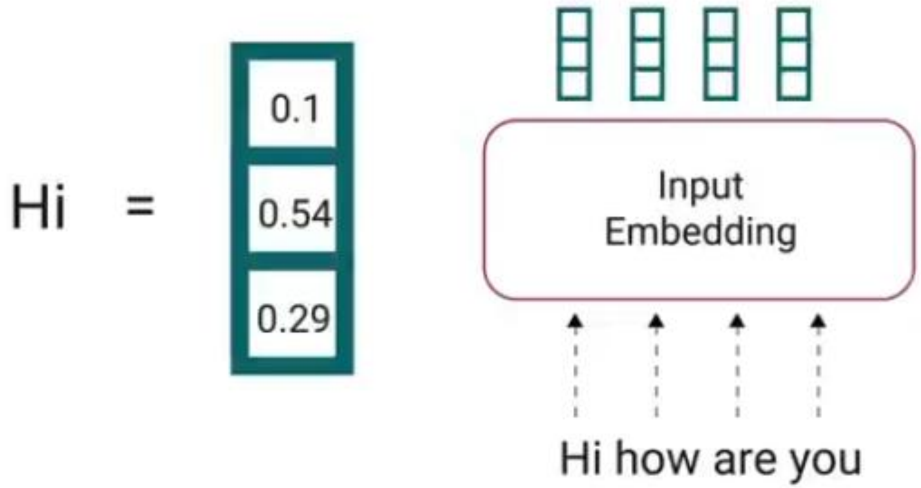
Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

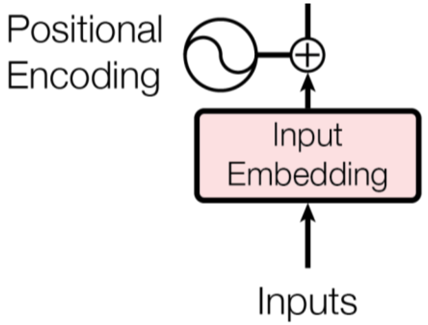
Embedding & Positional Encoding



Embedding & Positional Encoding

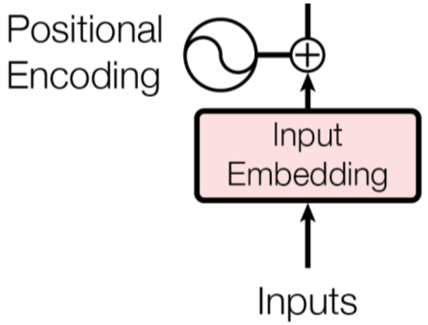
1. Input Embedding





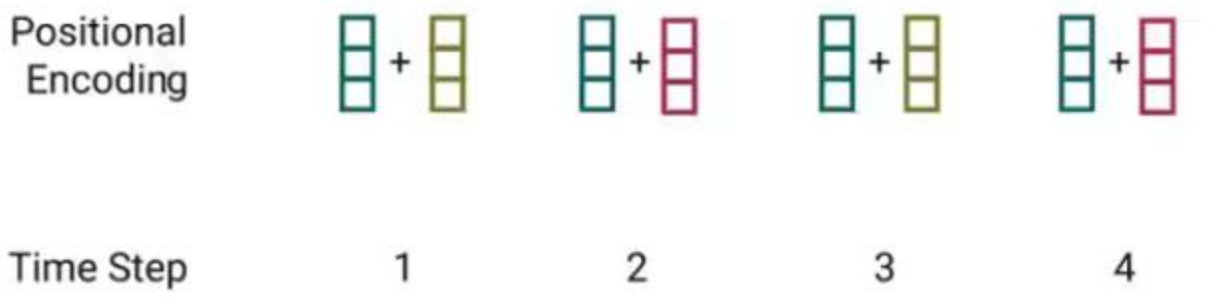
Embedding & Positional Encoding

2. Positional Encoding



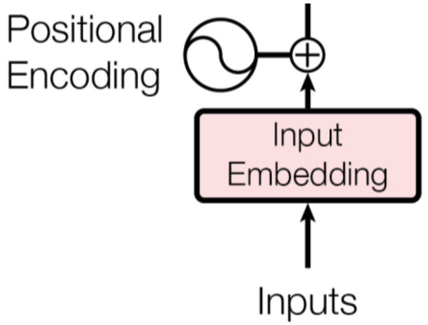
Embedding & Positional Encoding

2. Positional Encoding



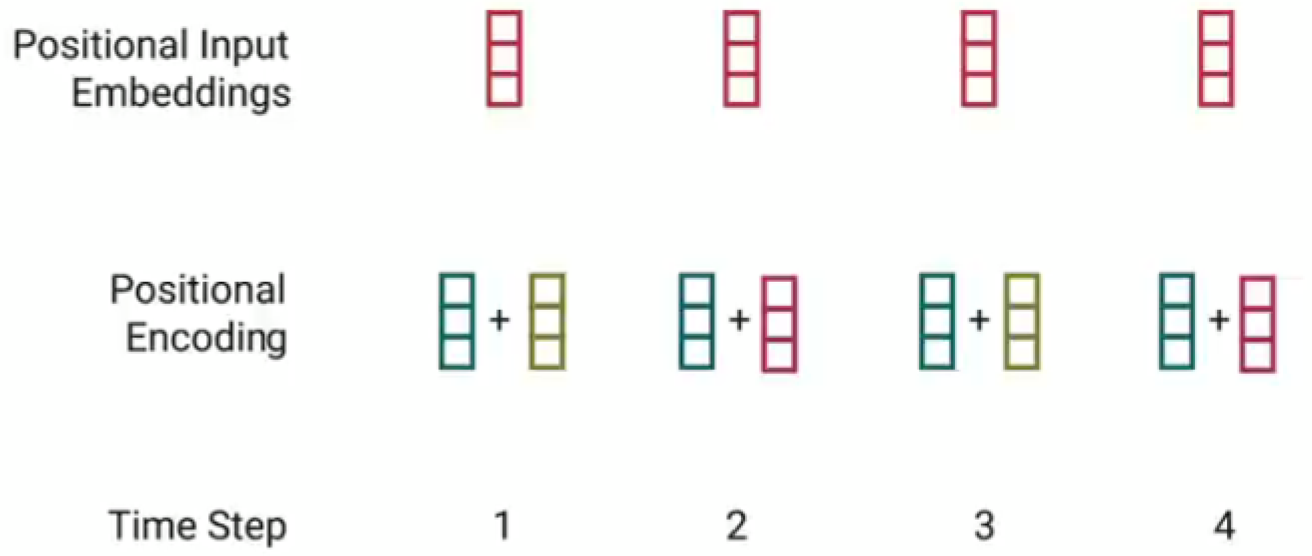
$$PE(pos, 2i+1) = cos(\frac{pos}{10000^{2i/dmodel}})$$

$$PE(pos, 2i) = sin(\frac{pos}{10000^{2i/dmodel}})$$



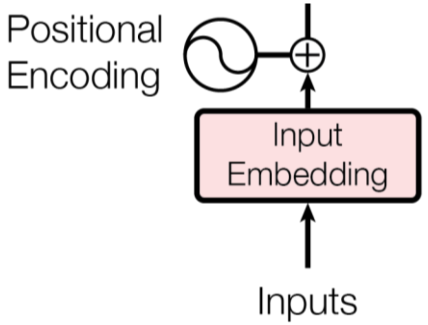
Embedding & Positional Encoding

2. Positional Encoding

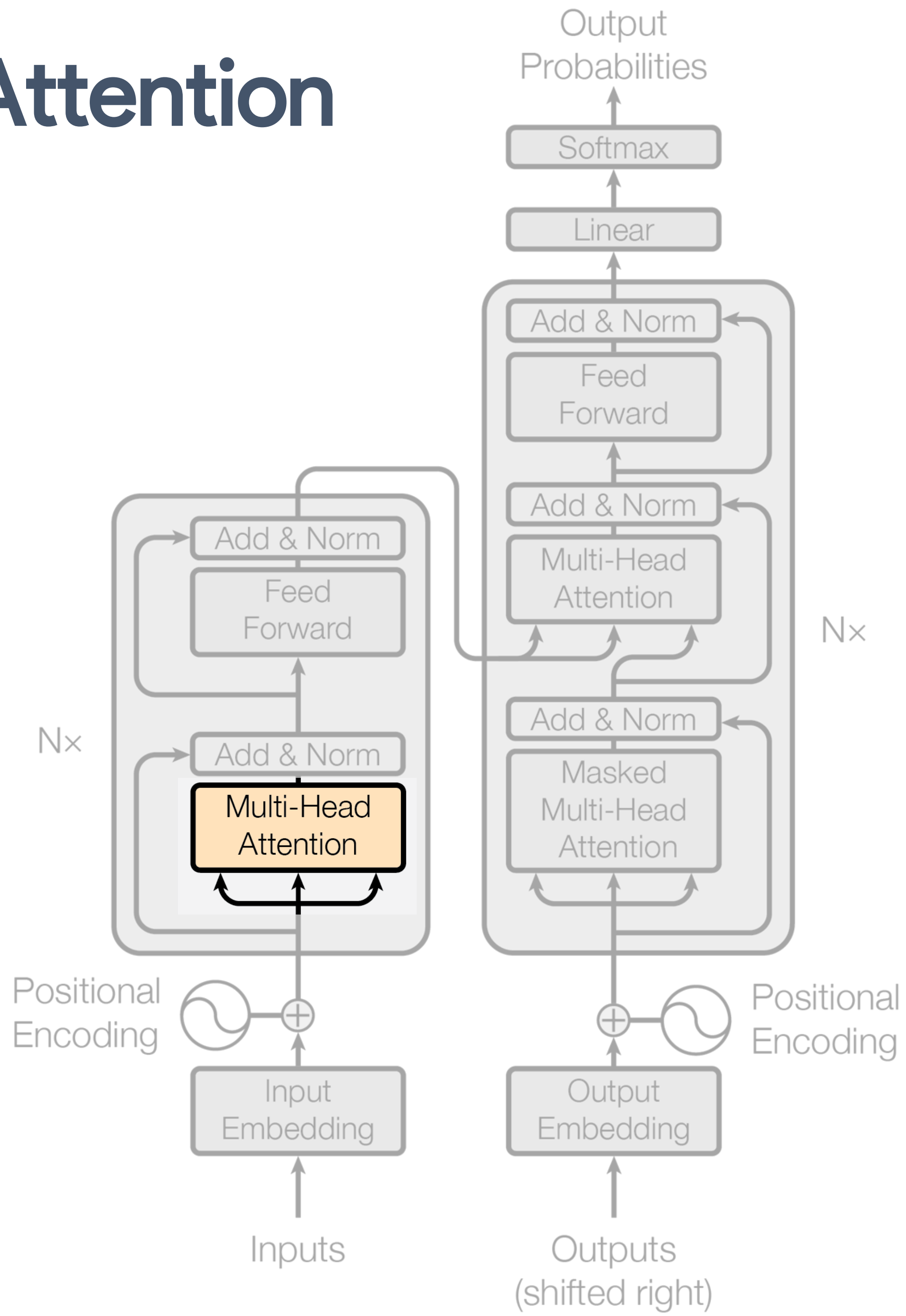


$$PE(pos, 2i+1) = cos(\frac{pos}{10000^{2i/dmodel}})$$

$$PE(pos, 2i) = sin(\frac{pos}{10000^{2i/dmodel}})$$

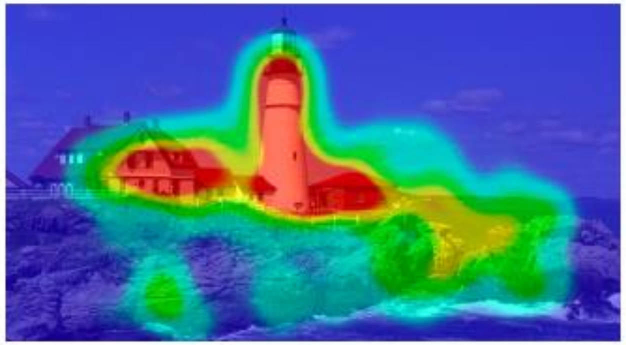


Multi-Head Self-Attention



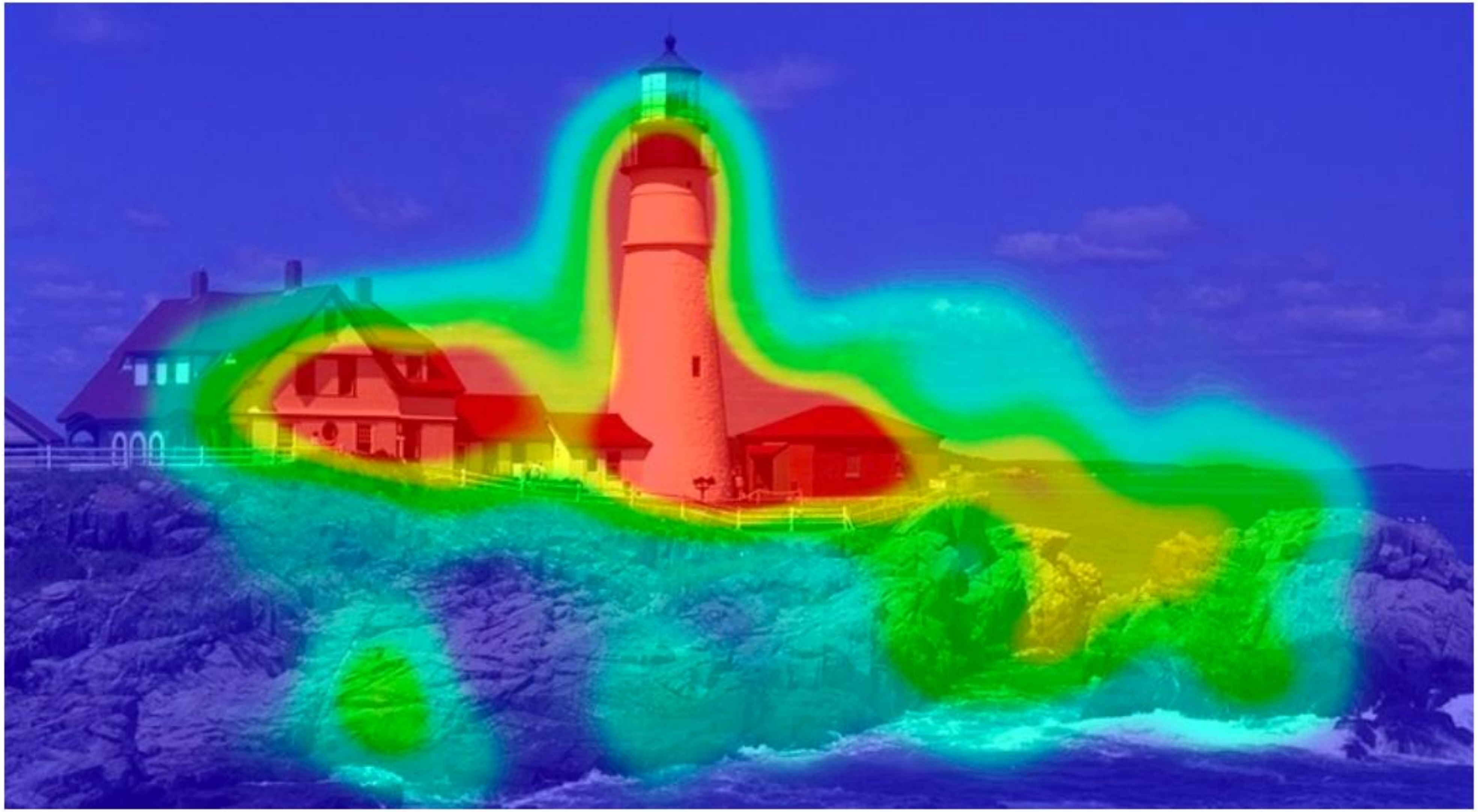
Attention

- Attention in Computer Vision
  - 2014: Attention used to highlight important parts of an image that contribute to a desired output



- Attention in NLP
  - 2015: Aligned machine translation
  - 2017: Language modeling with **Transformer networks**

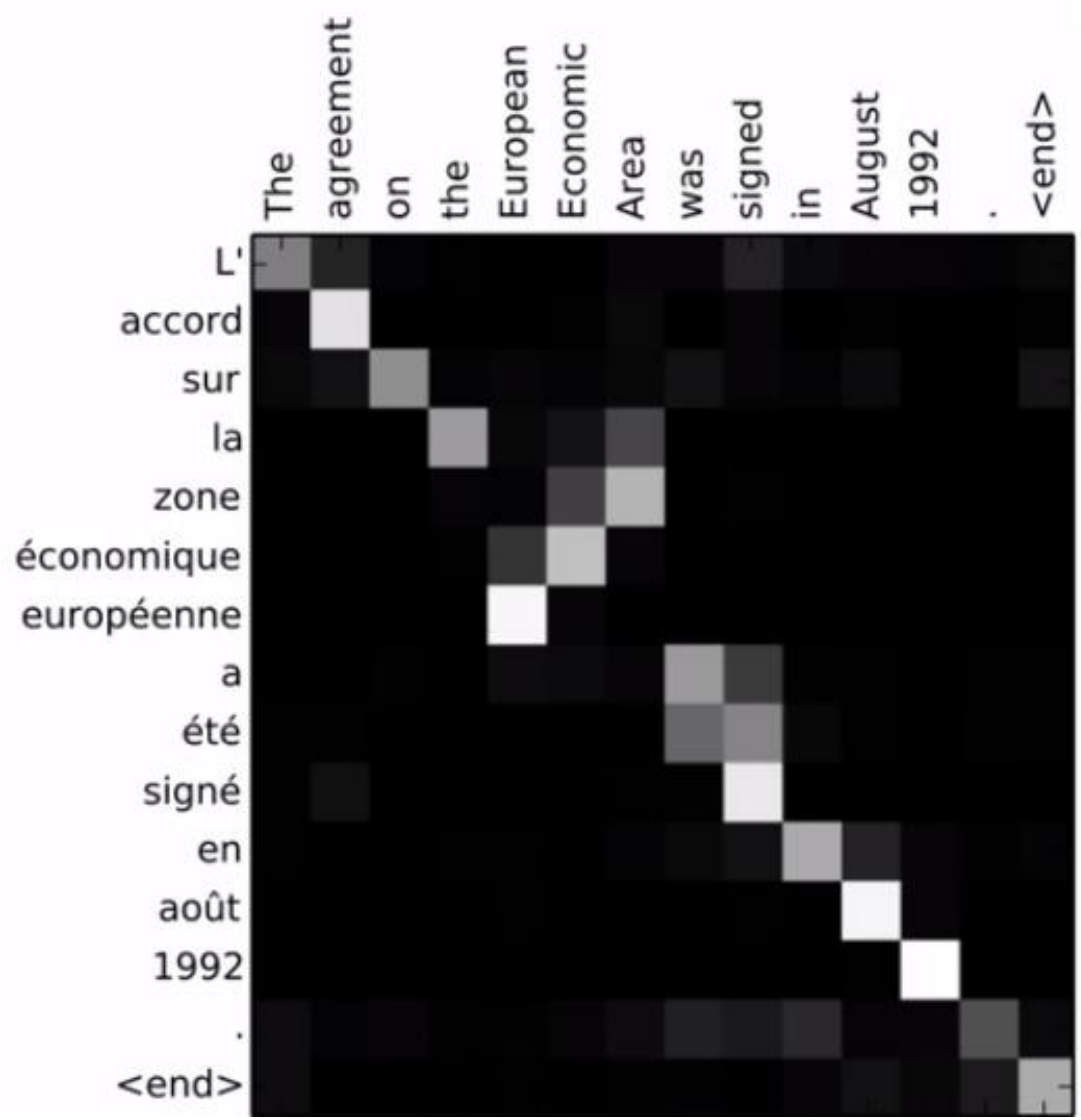
Attention



Attention


Affenpinscher
Brussels Griffon
Cavalier King Charles Spaniel
Chihuahua
Chinese Crested
English Toy Spaniel
Havanese
Italian Greyhound
Japanese Chin
Maltese
Manchester Terrier
Miniature Pinscher
Papillon
Pekingese
Pomeranian
Pug
Shih Tzu
Silky Terrier
Toy Fox Terrier
Toy Poodle
Yorkshire Terrier

Attention



The
agreement
on
the
European
Economic
Area
was
signed
in
August
1992
.
<end>
L'
accord
sur
la
zone
économique
européenne
a
été
signé
en
août
1992
.
<end>

Intuition behind Self-Attention

Attending to the most important parts of an input.

Intuition behind Self-Attention

Attending to the most important parts of an input.

1. Identify which parts to attend to
2. Extract the features with high attention

Intuition behind Self-Attention

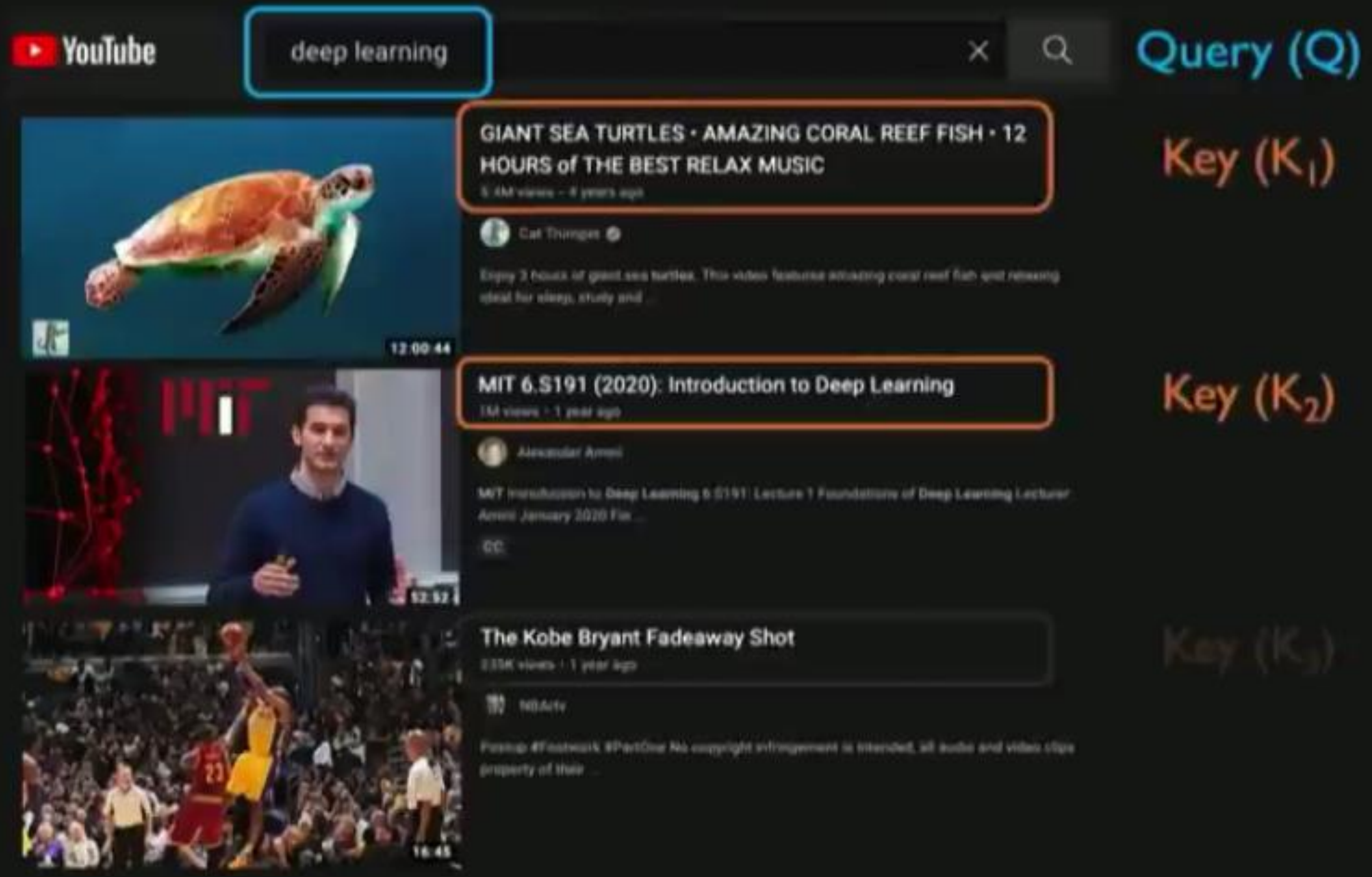
Attending to the most important parts of an input.

1. Identify which parts to attend to

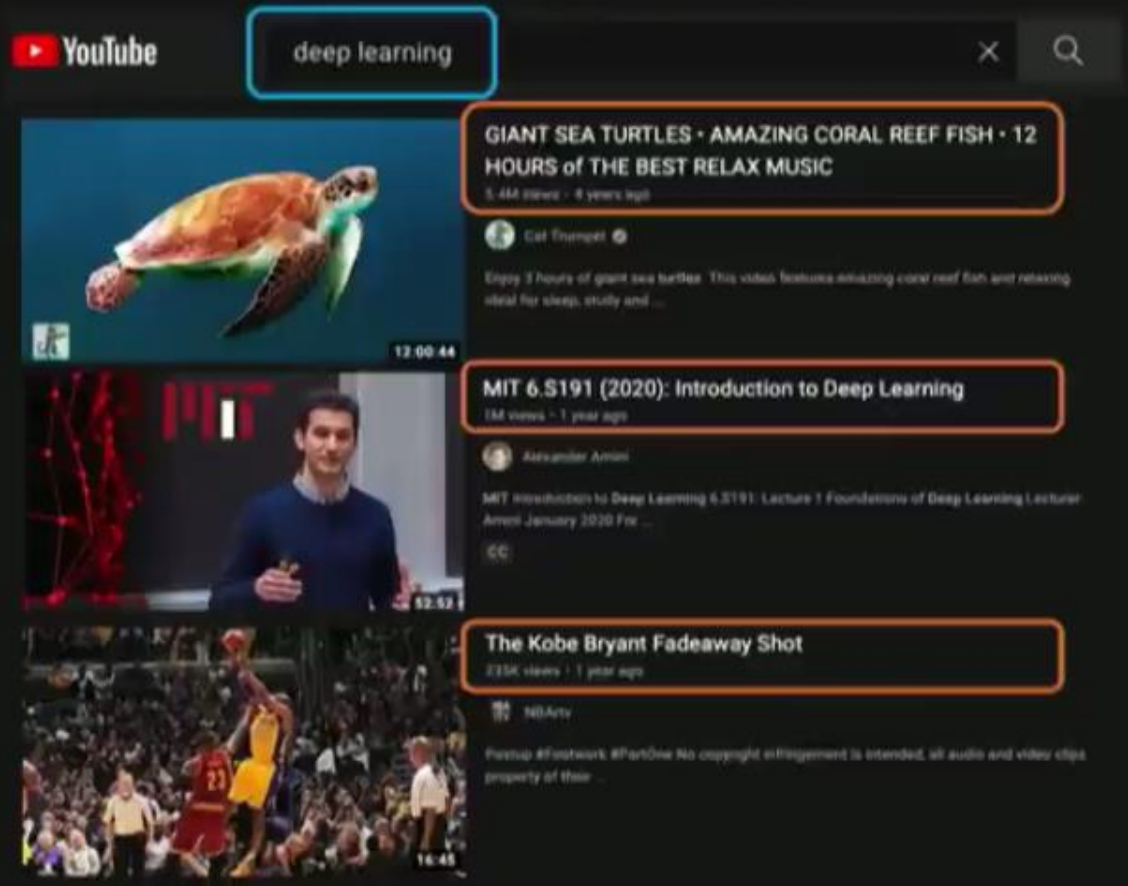
2. Extract the features with high attention

Similar to a search problem!

Understanding Attention with Search



Understanding Attention with Search



Query (Q)

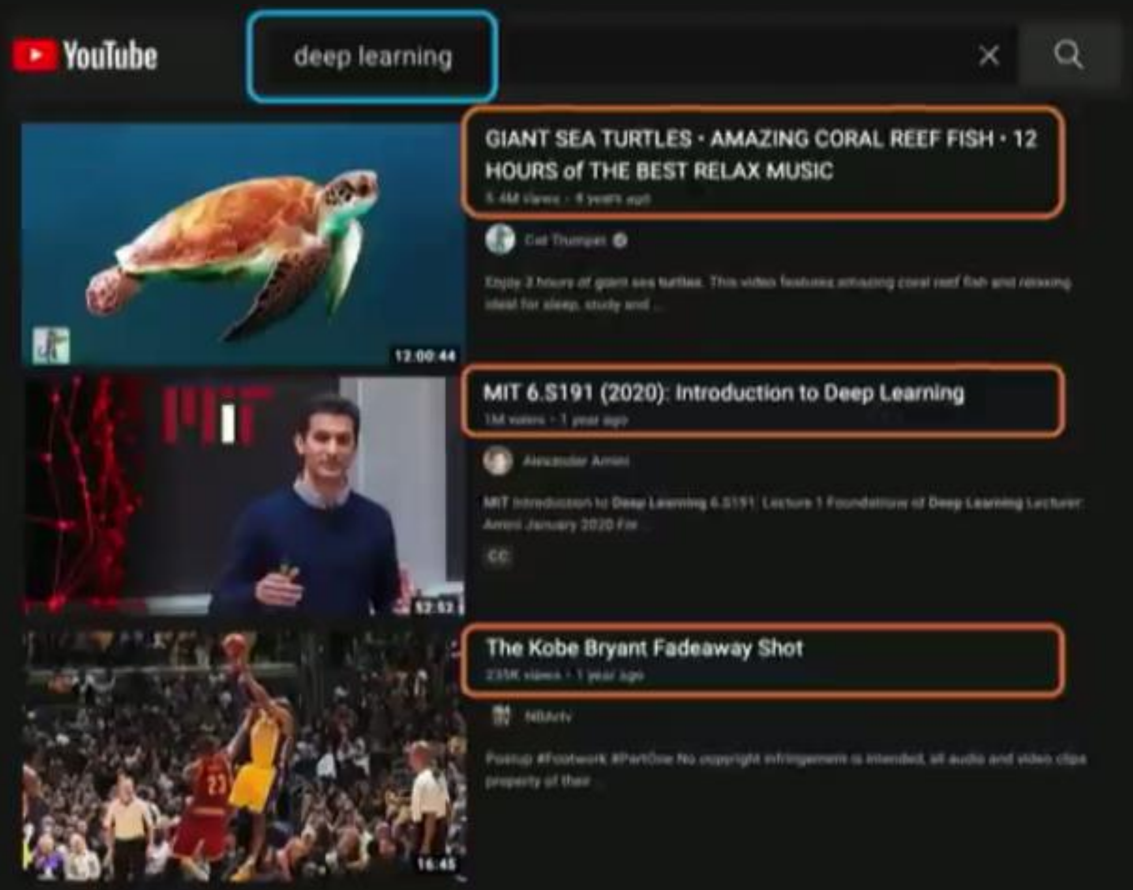
Key (K_1)

Key (K_2)

Key (K_3)

1. **Compute attention mask:** how similar is each key to the desired query?

Understanding Attention with Search



Query (Q)

Key (K_1)

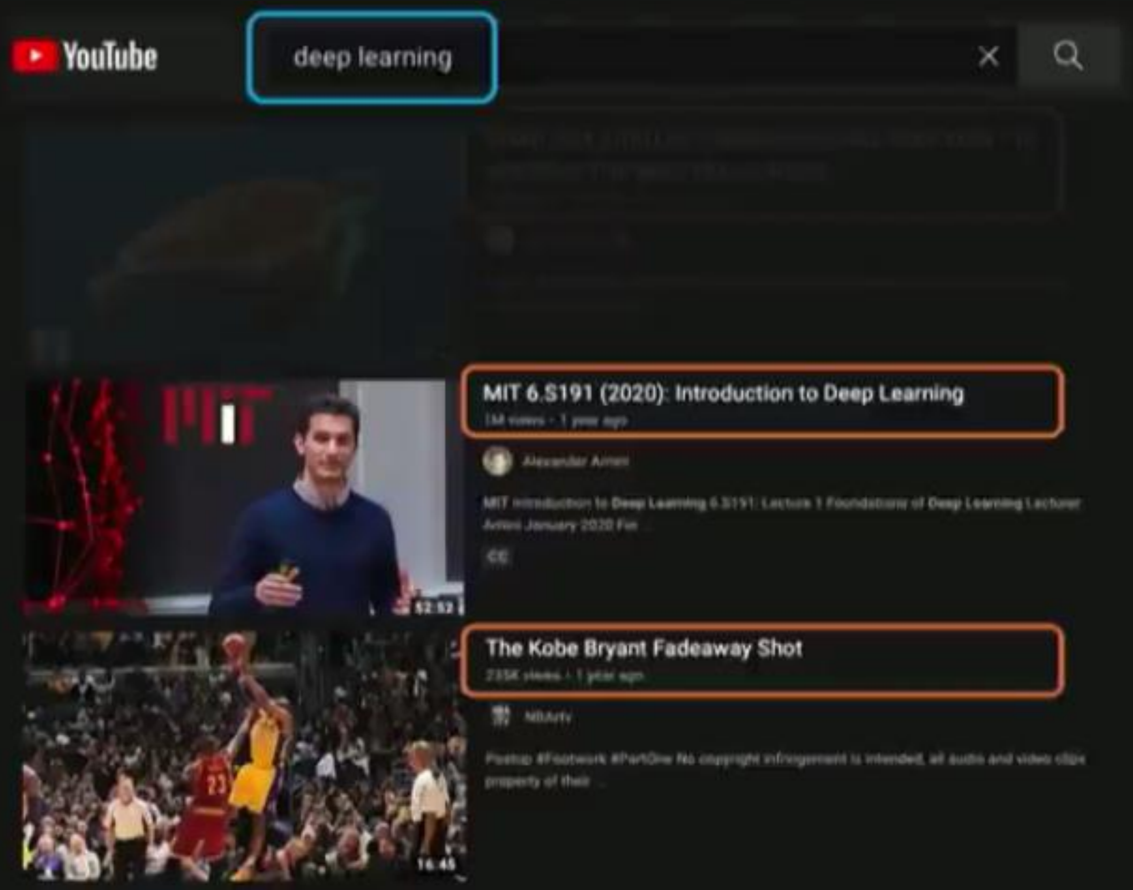
How similar is the key to the query?

Key (K_2)

Key (K_3)

1. **Compute attention mask:** how similar is each key to the desired query?

Understanding Attention with Search



Query (Q)

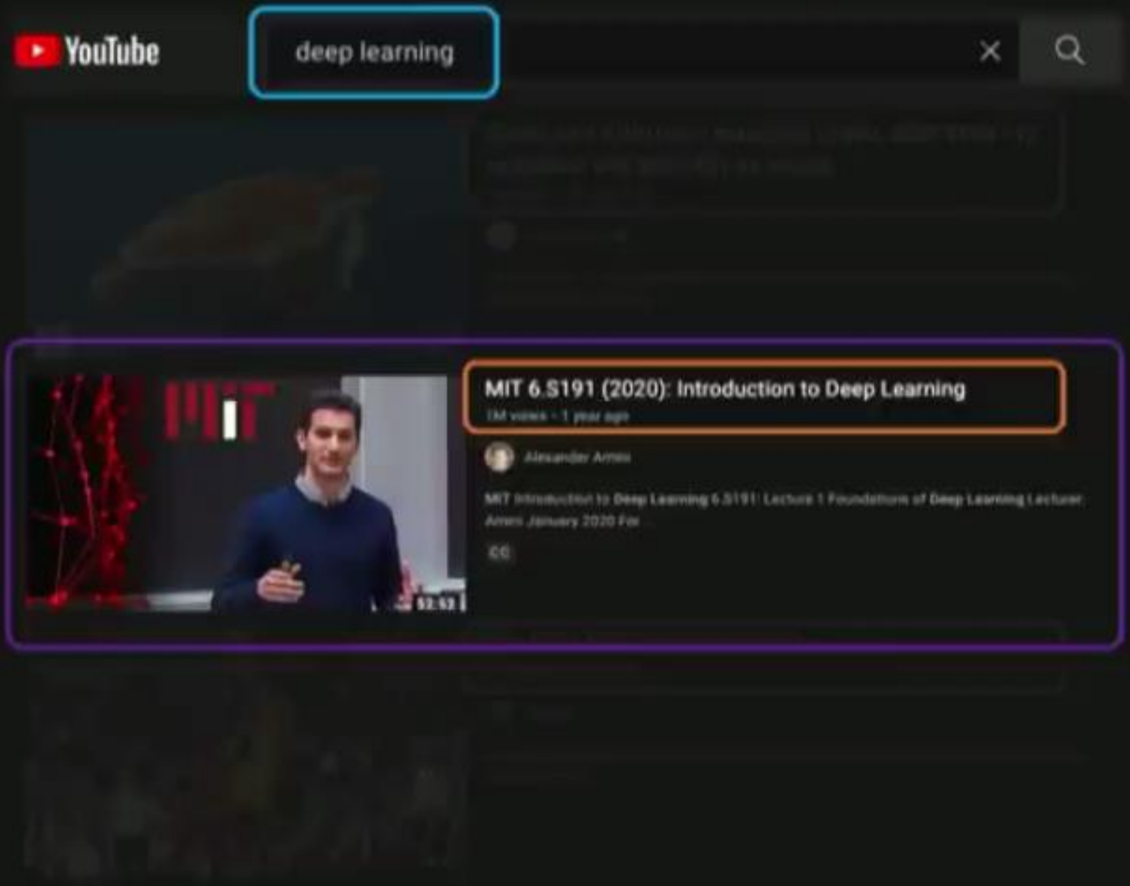
How similar is the key to the query?

Key (K_2)

Key (K_3)

1. **Compute attention mask:** how similar is each key to the desired query?

Understanding Attention with Search



Query (Q)

Key (K_2)

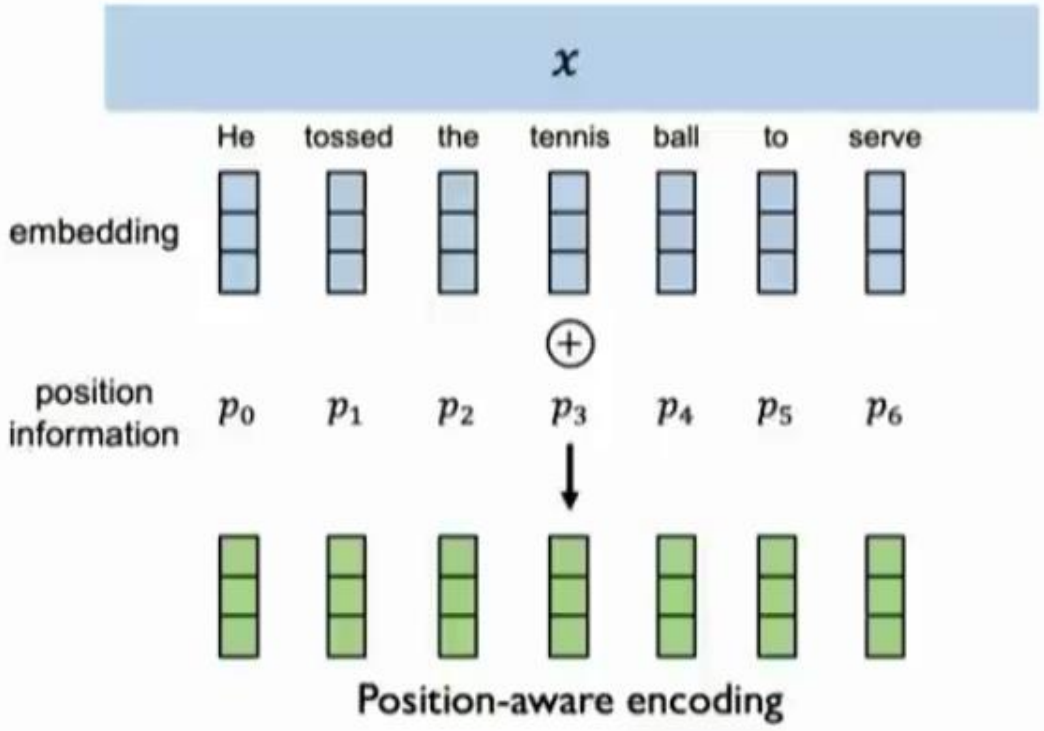
Value (V)

2. **Extract values based on attention:**
Return the values highest attention

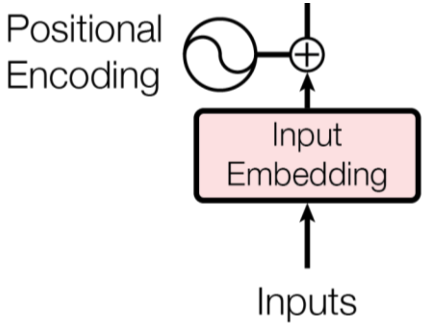
Learning Self-Attention

Goal: identify and attend to most important features in input.

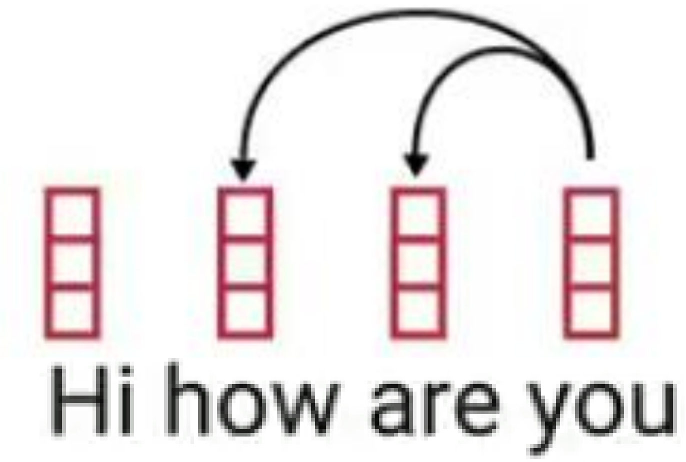
1. Encode **position** information
2. Extract query, key, value for search
3. Compute attention weighting
4. Extract features with high attention

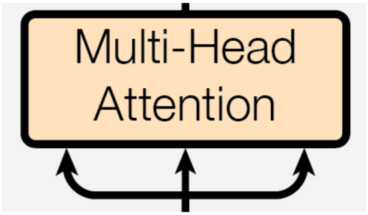


Data is fed in all at once! Need to encode position information to understand order.

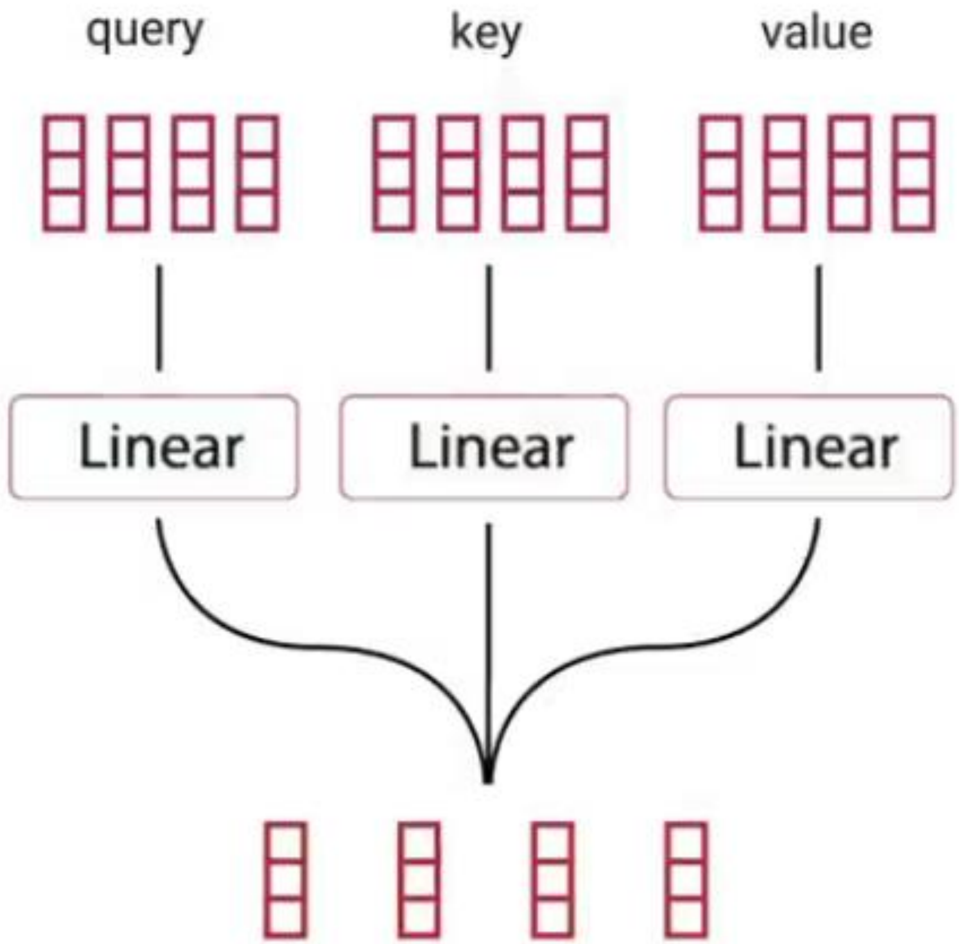


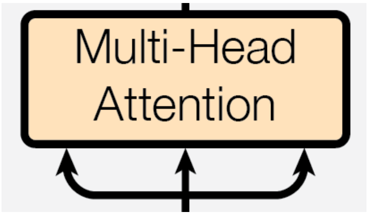
Learning Self-Attention



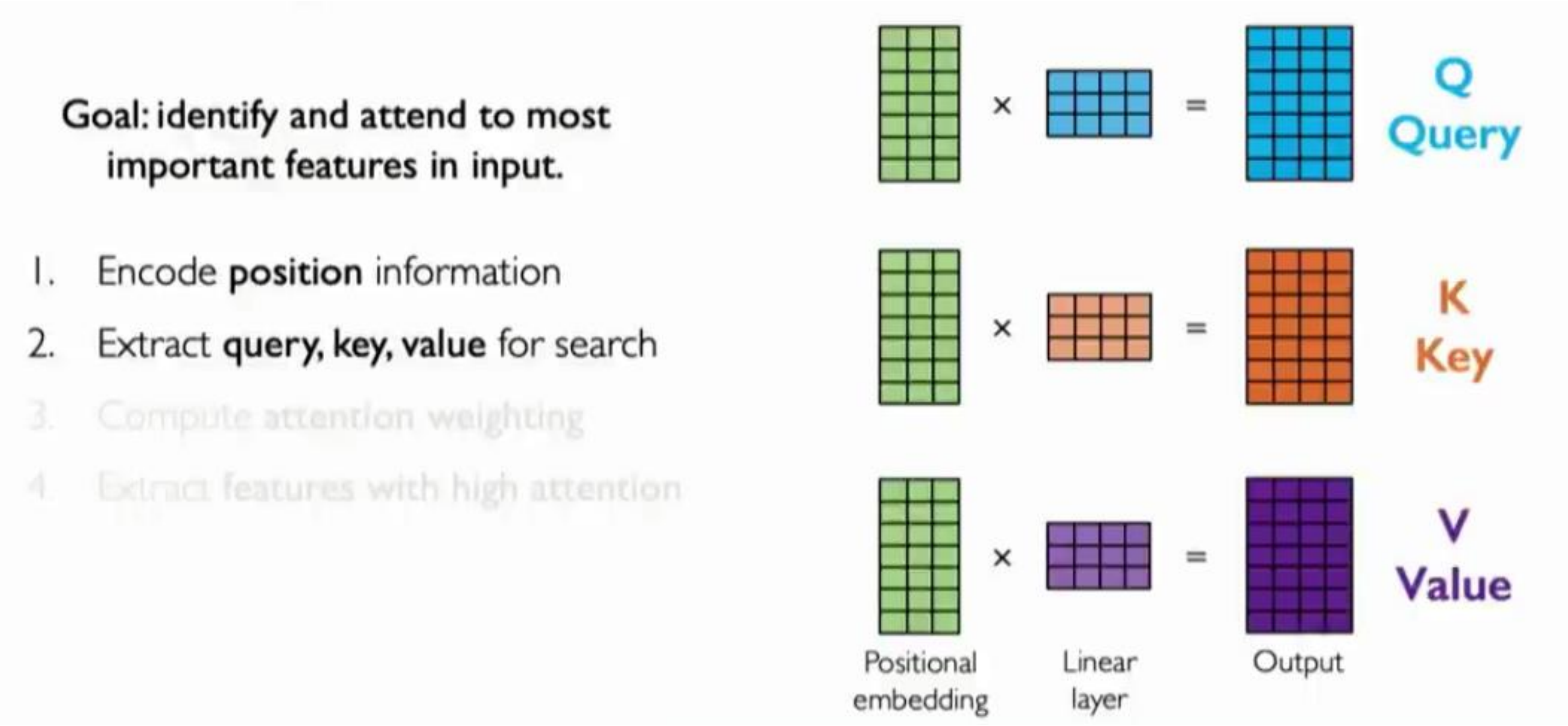


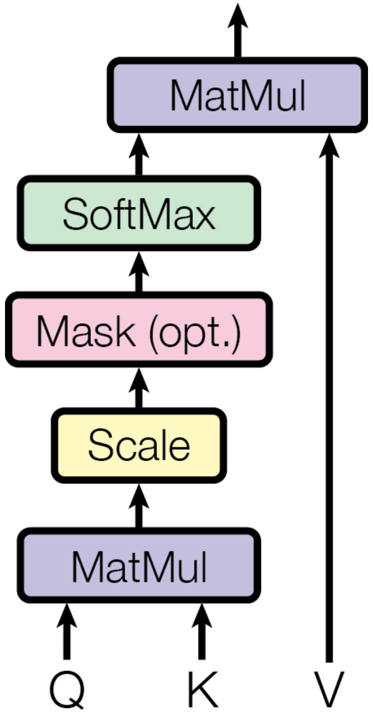
Learning Self-Attention





Learning Self-Attention





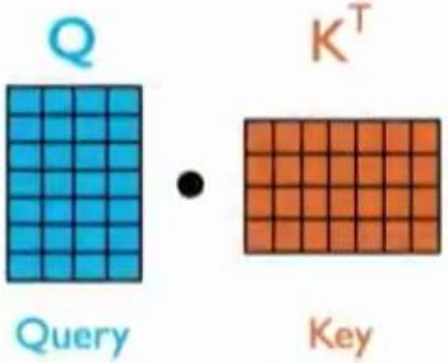
Learning Self-Attention

Goal: identify and attend to most important features in input.

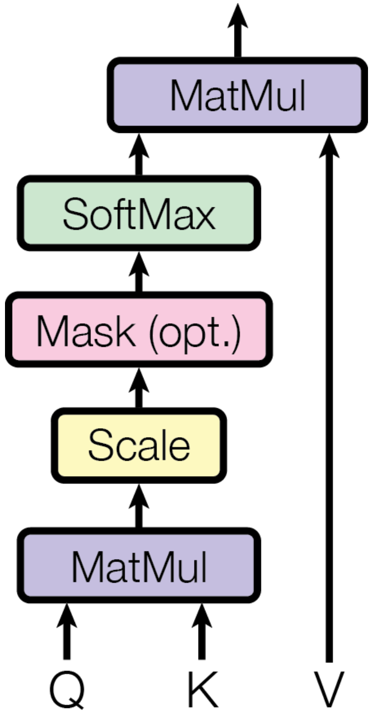
1. Encode **position** information
2. Extract **query, key, value** for search
3. Compute **attention weighting**
4. Extract features with high attention

Attention score: compute pairwise similarity between each query and key

How to compute similarity between two sets of features?

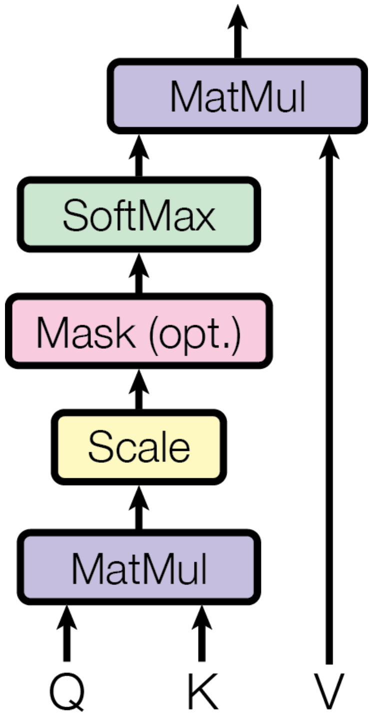


Also known as the "cosine similarity"



Learning Self-Attention

	Hi	how	are	you
Hi	98	27	10	12
how	27	89	31	67
are	10	31	91	54
you	12	67	54	92



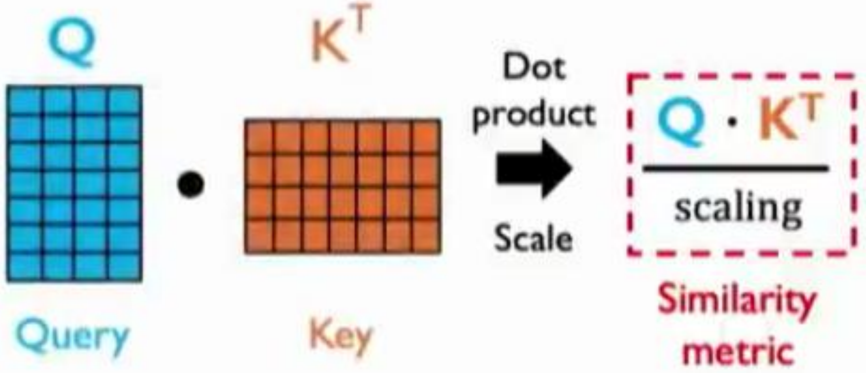
Learning Self-Attention

Goal: identify and attend to most important features in input.

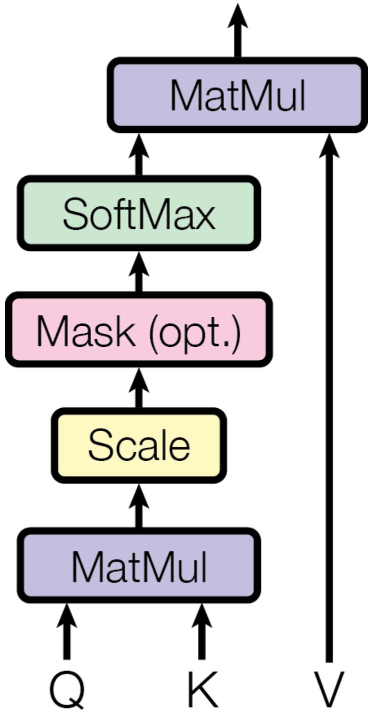
1. Encode **position** information
2. Extract **query, key, value** for search
3. Compute **attention weighting**
4. Extract features with high attention

Attention score: compute pairwise similarity between each query and key

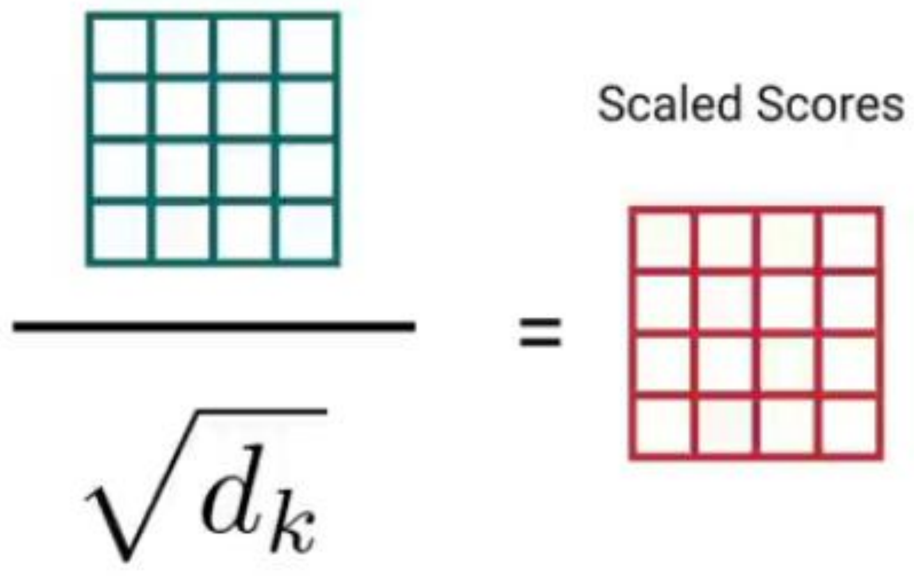
How to compute similarity between two sets of features?

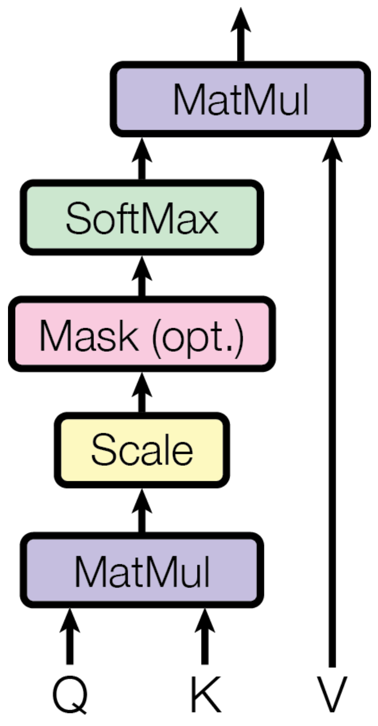


Also known as the "cosine similarity"



Learning Self-Attention

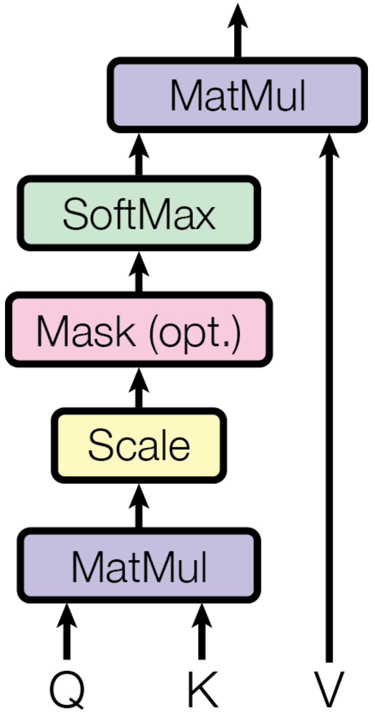




Learning Self-Attention

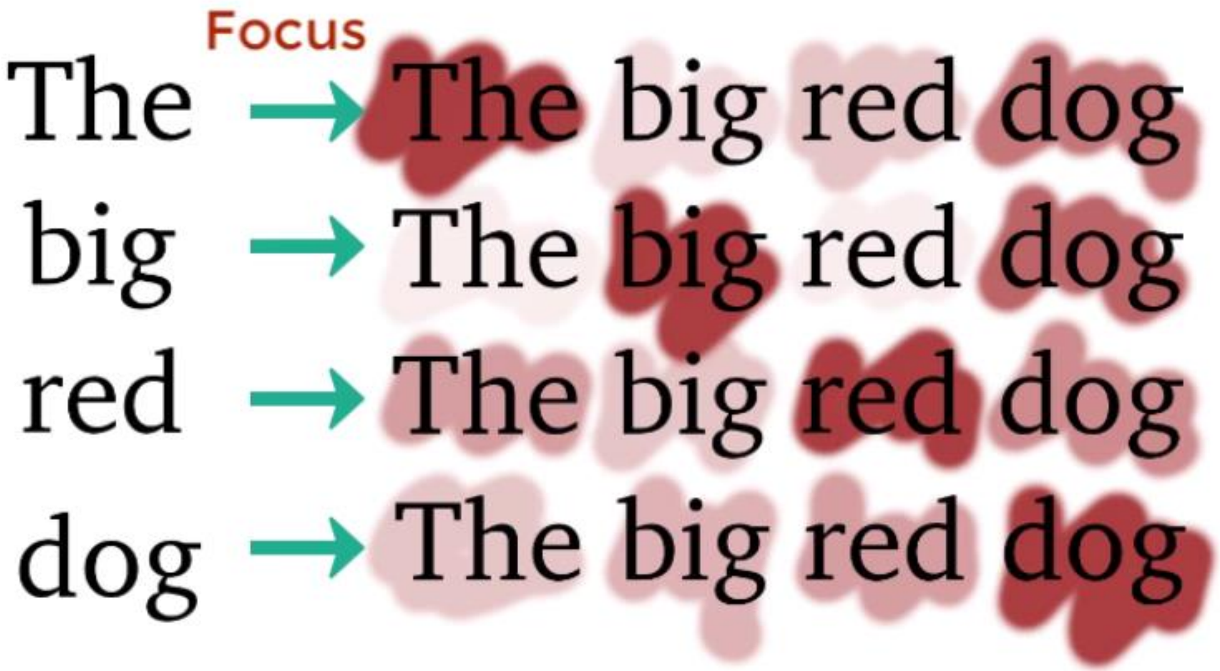
Softmax( ) =

	Hi	how	are	you
Hi	0.7	0.1	0.1	0.1
how	0.1	0.6	0.2	0.1
are	0.1	0.3	0.6	0.1
you	0.1	0.3	0.3	0.3



Self-Attention

Attention : What part of the input should we focus?



Attention Vectors

$[0.71 \quad 0.04 \quad 0.07 \quad 0.18]^T$

$[0.01 \quad 0.84 \quad 0.02 \quad 0.13]^T$

$[0.09 \quad 0.05 \quad 0.62 \quad 0.24]^T$

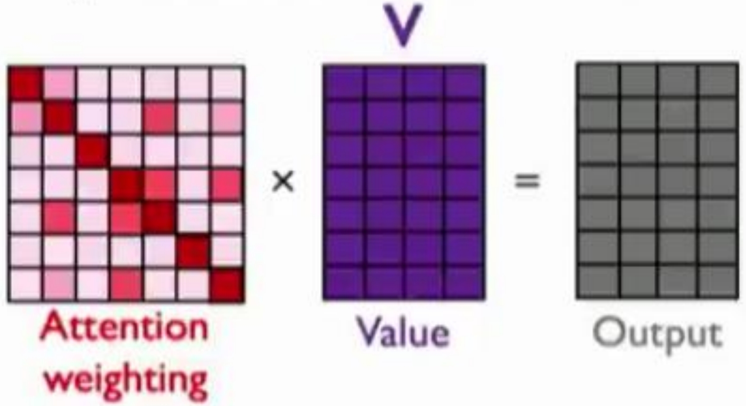
$[0.03 \quad 0.03 \quad 0.03 \quad 0.91]^T$

Learning Self-Attention

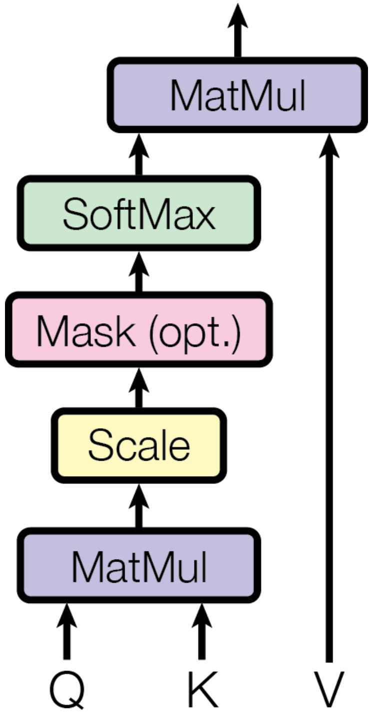
Goal: identify and attend to most important features in input.

1. Encode **position** information
2. Extract **query, key, value** for search
3. Compute **attention weighting**
4. Extract **features with high attention**

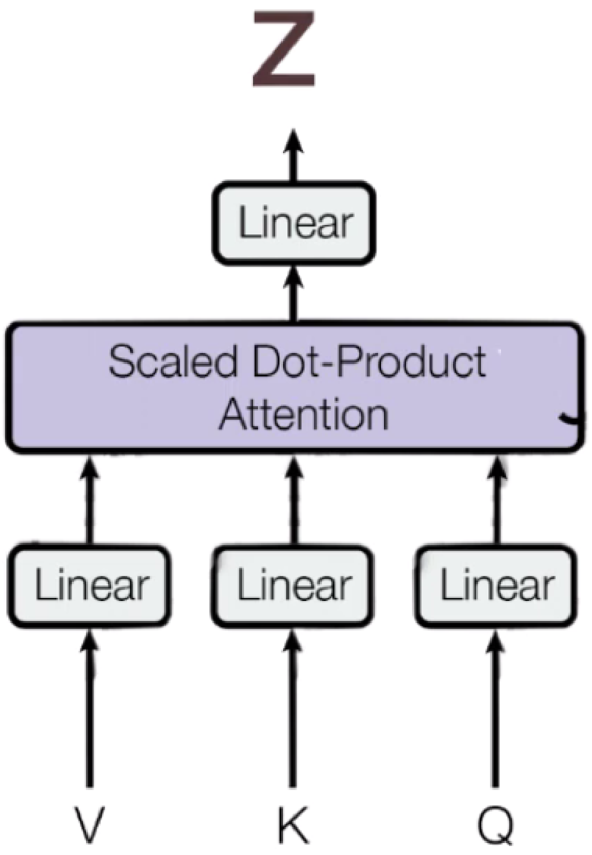
Last step: self-attend to extract features



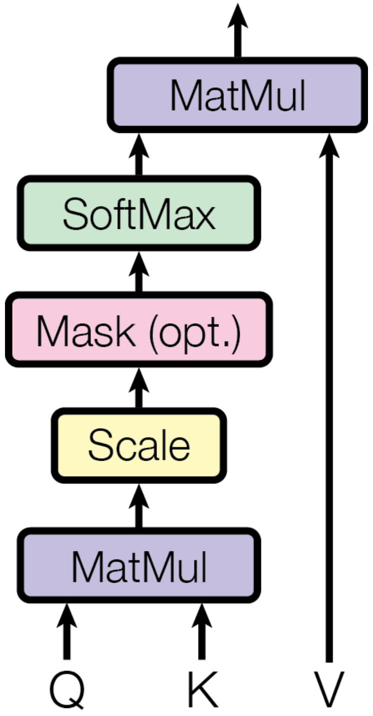
$$\underbrace{softmax\left(\frac{Q \cdot K^T}{scaling}\right)} \cdot \underbrace{V} = \underbrace{A(Q, K, V)}$$



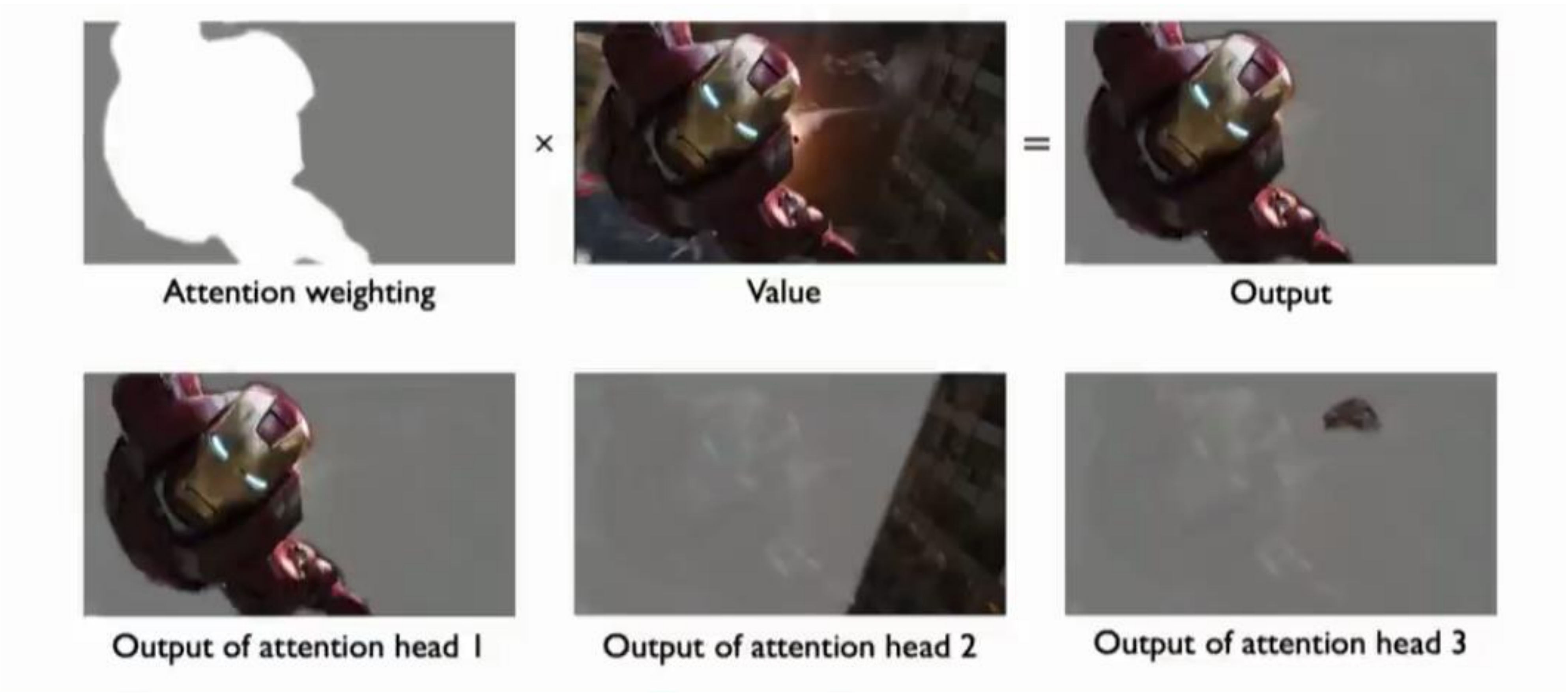
Self-Attention



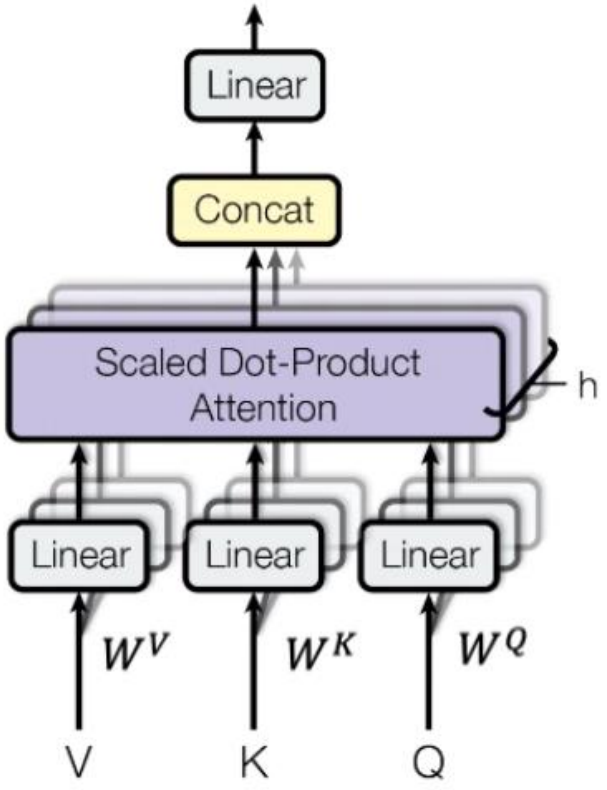
$$Z = softmax\left(\frac{Q.K^T}{\sqrt{Dimension\ of\ vector\ Q, K\ or\ V}}\right).V$$



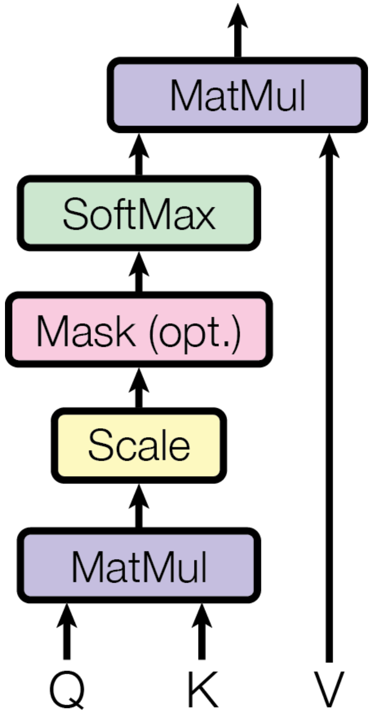
Multi-Head Self-Attention



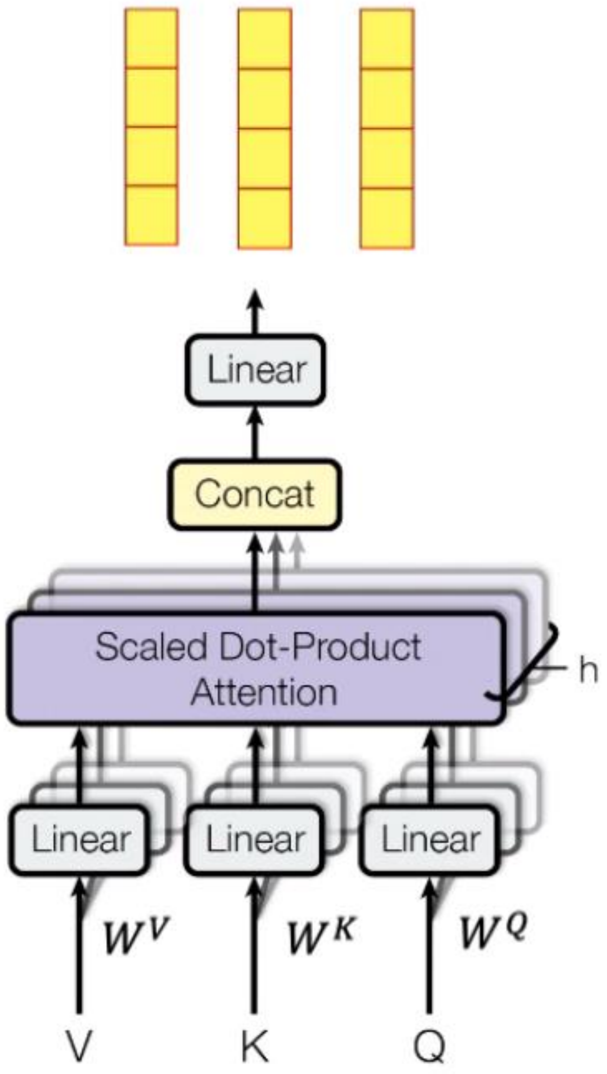
Multi-Head Self-Attention



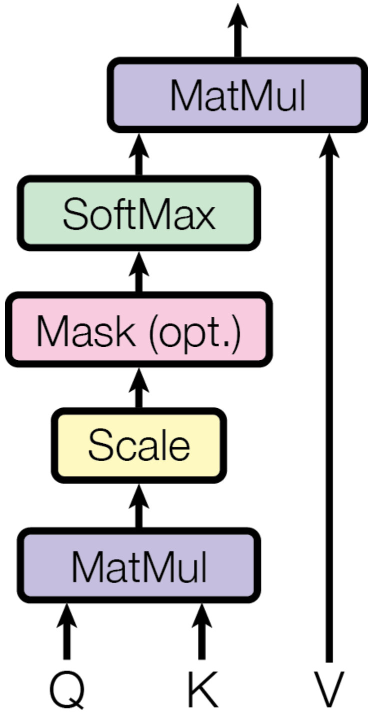
$$Z = softmax\left(\frac{Q.K^T}{\sqrt{Dimension\ of\ vector\ Q, K\ or\ V}}\right).V$$



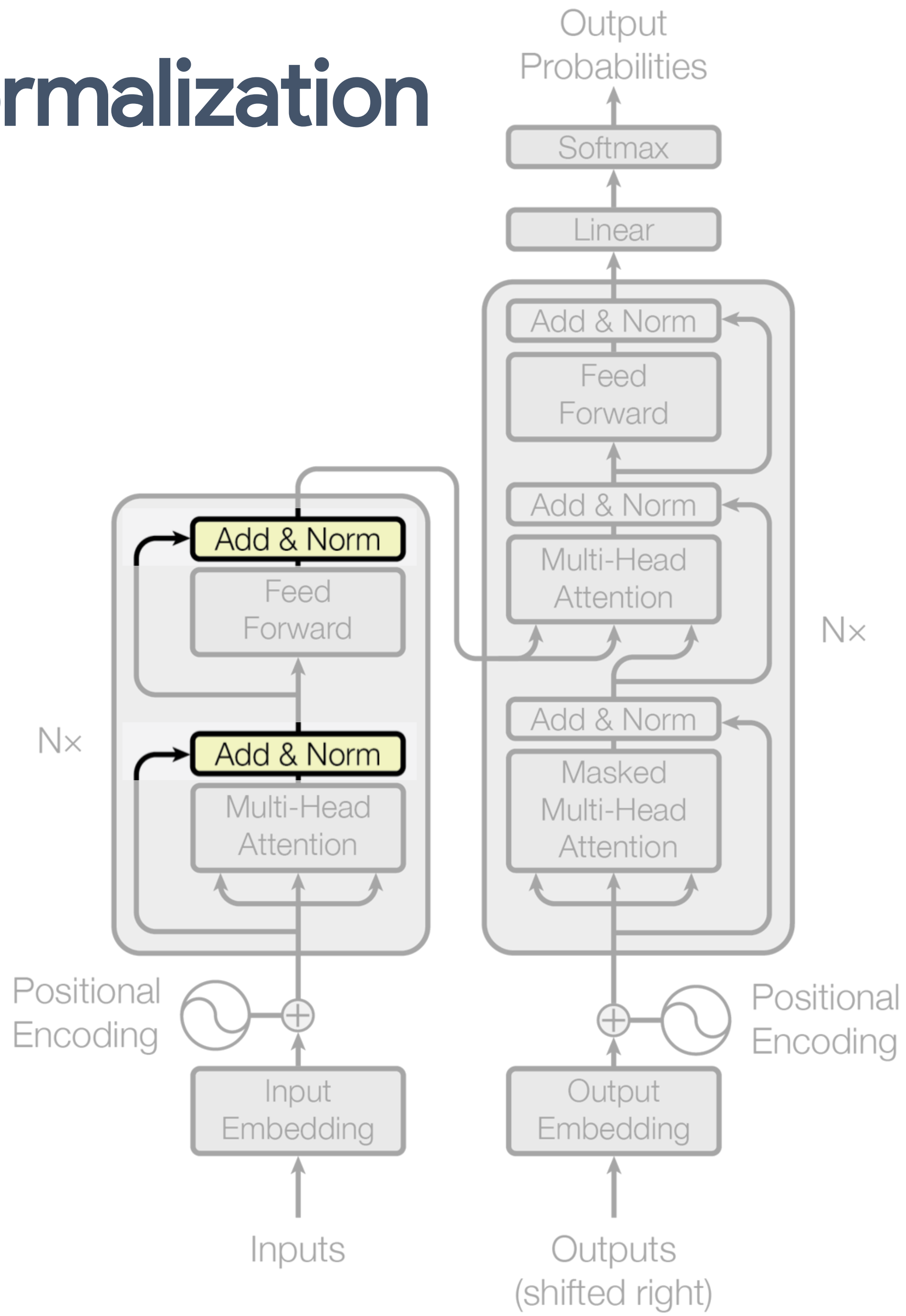
Multi-Head Self-Attention



$$Z = softmax\left(\frac{Q.K^T}{\sqrt{Dimension\ of\ vector\ Q, K\ or\ V}}\right).V$$

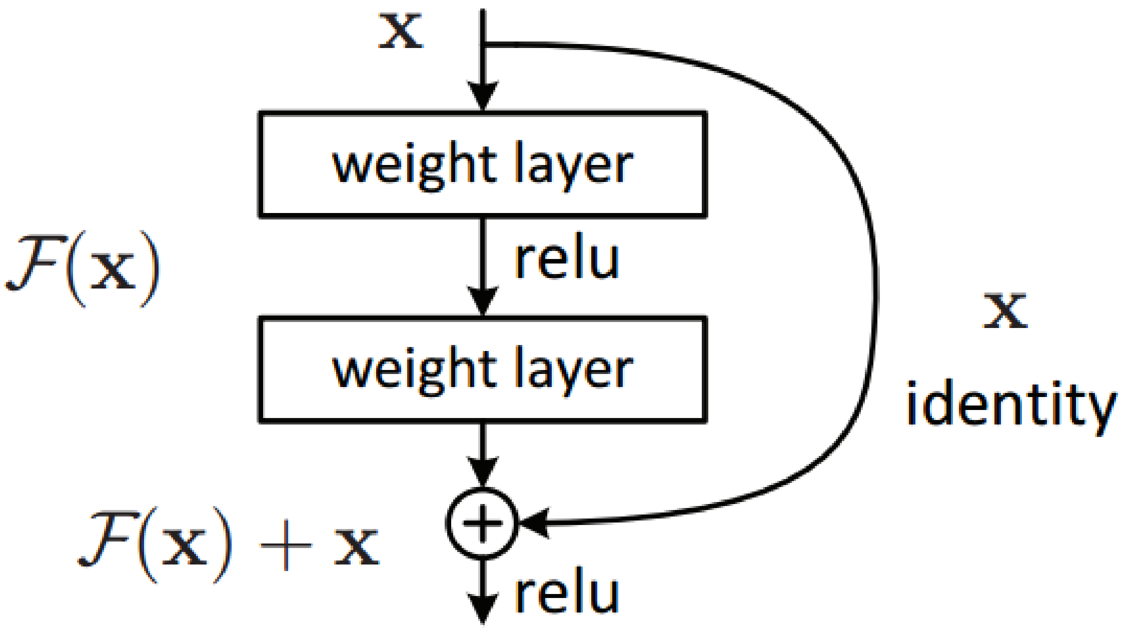


Residual, Add, Normalization

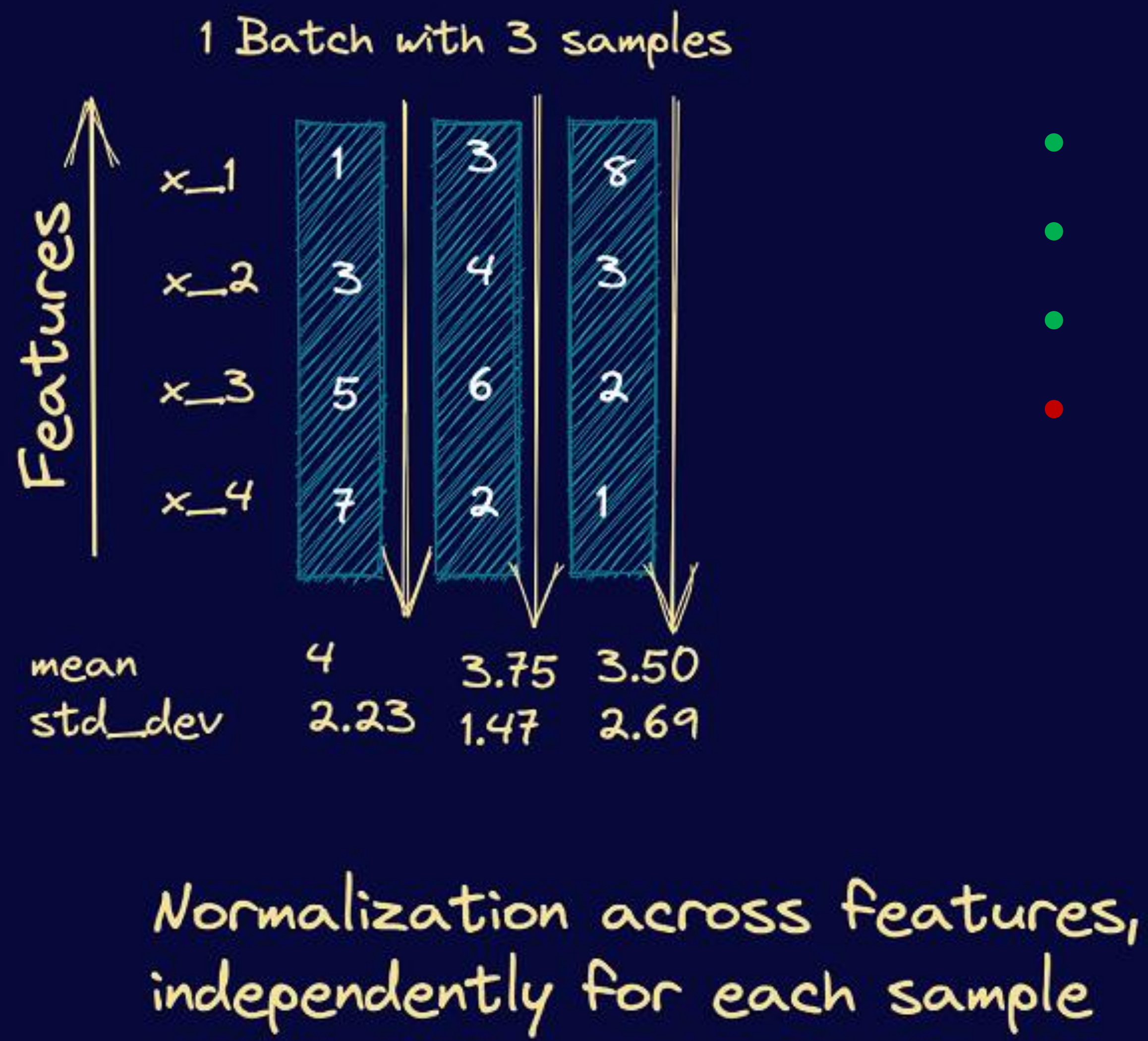


Residual Layer

By explicitly learning the residual mapping, the network can focus on learning the fine-grained details or changes needed to refine the input, rather than trying to learn the complete transformation from scratch.

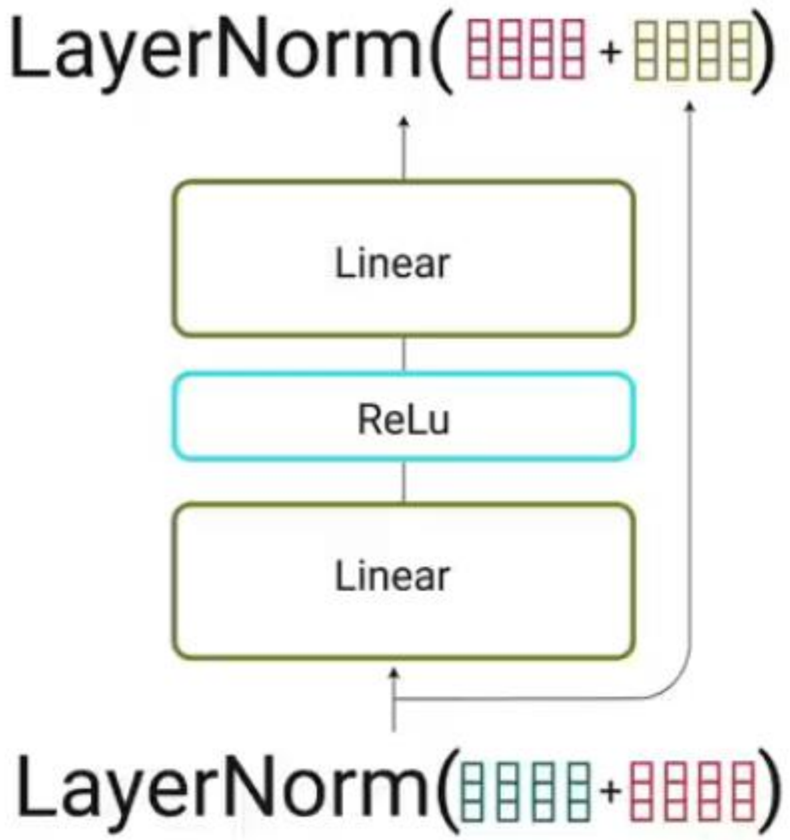


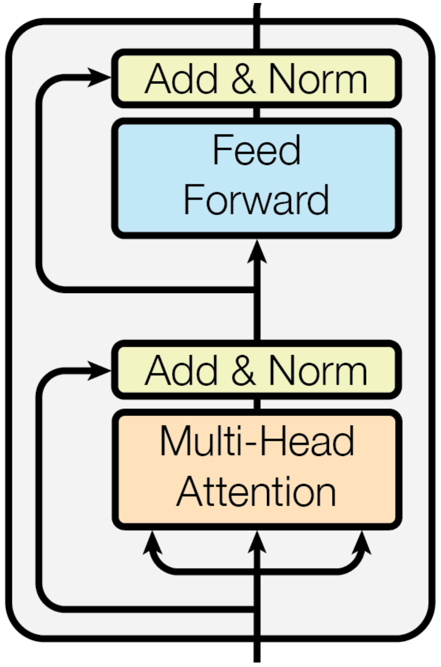
Layer Normalization



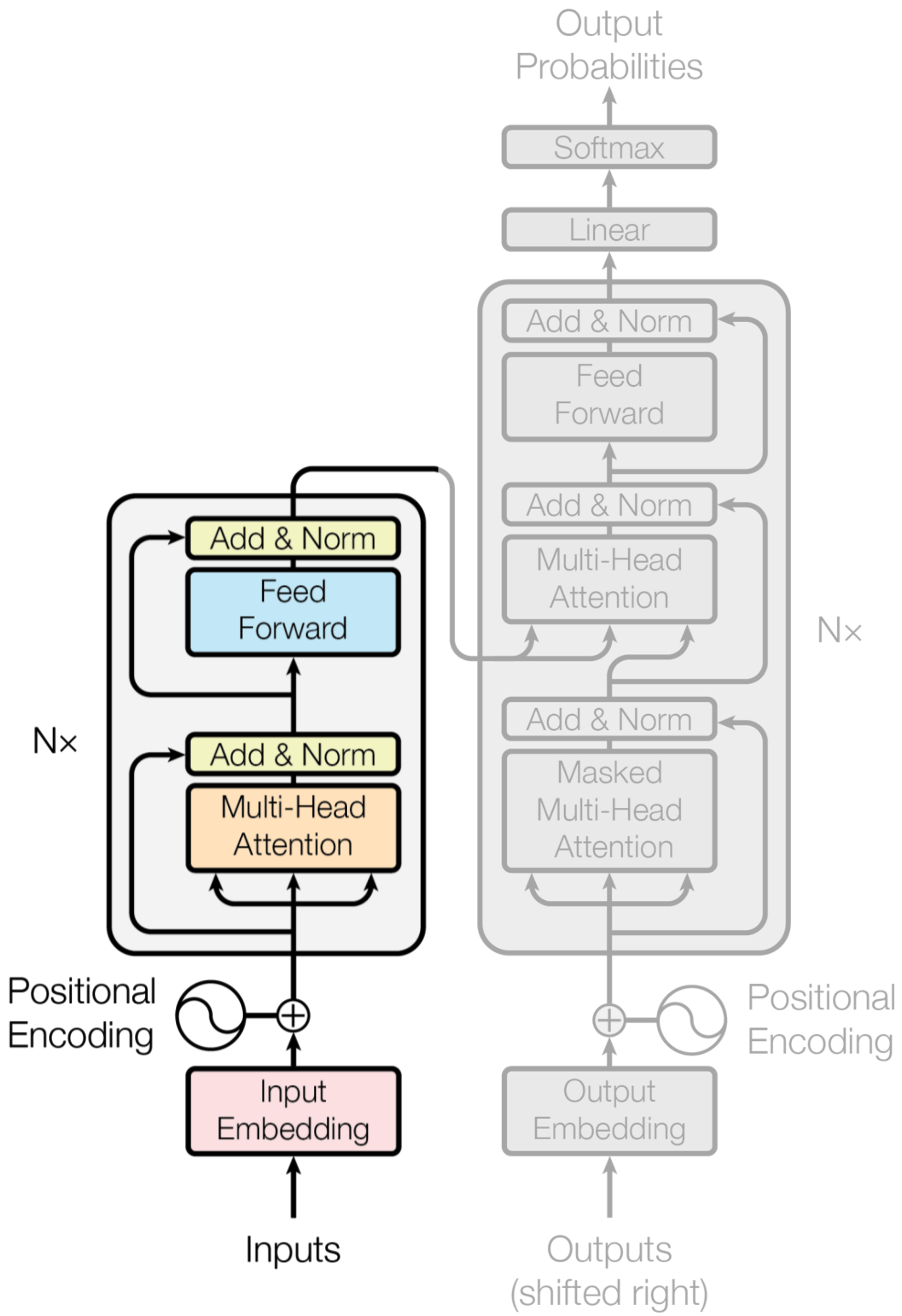
- can deal with sequences
- any batch number works
- can parallelize
- cannot work well with CNN

Add & Layer Normalization & Feed-Forward





Encoder


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

English-French Translation

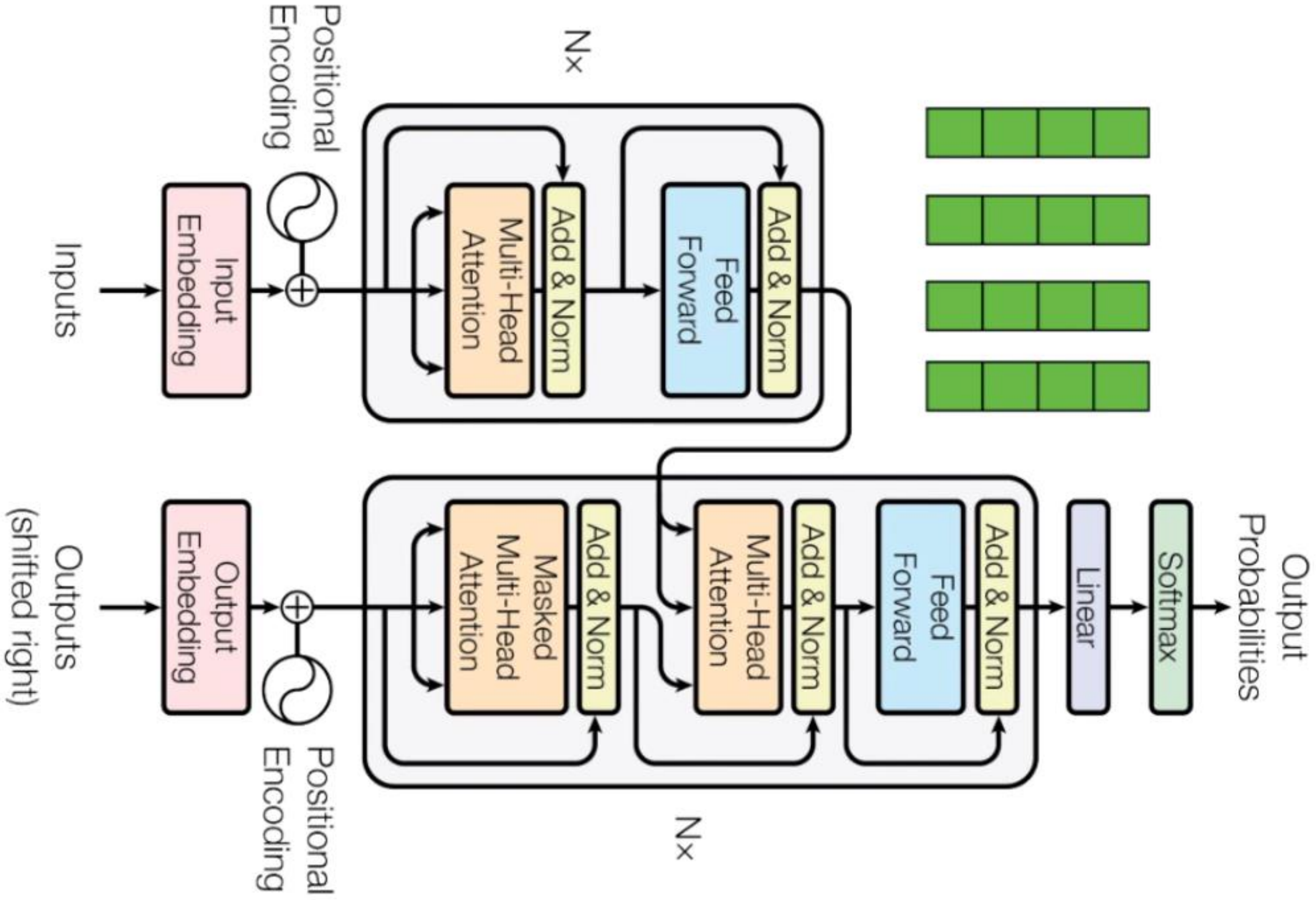
The big red dog
Le gros chien rouge

The

big

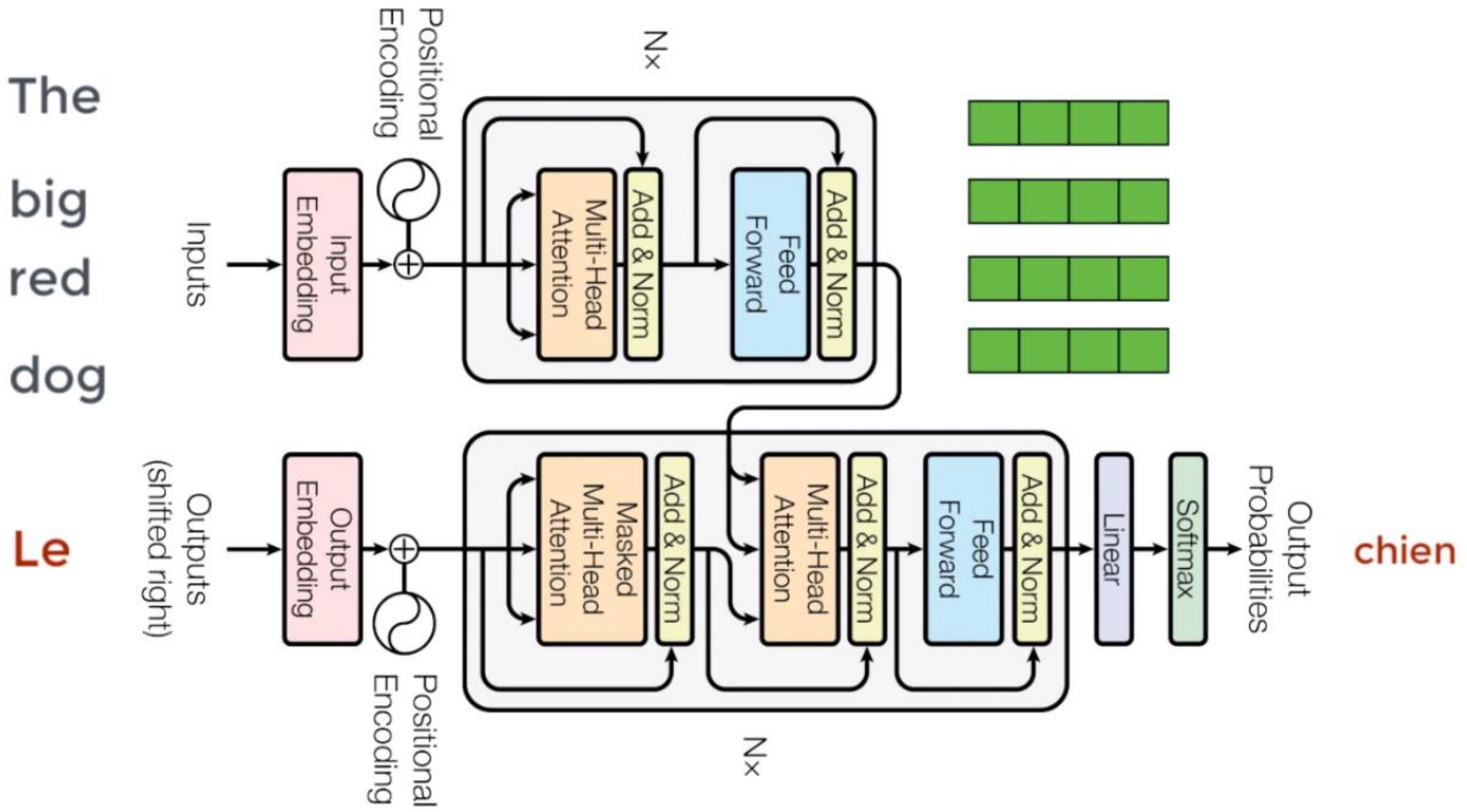
red

dog

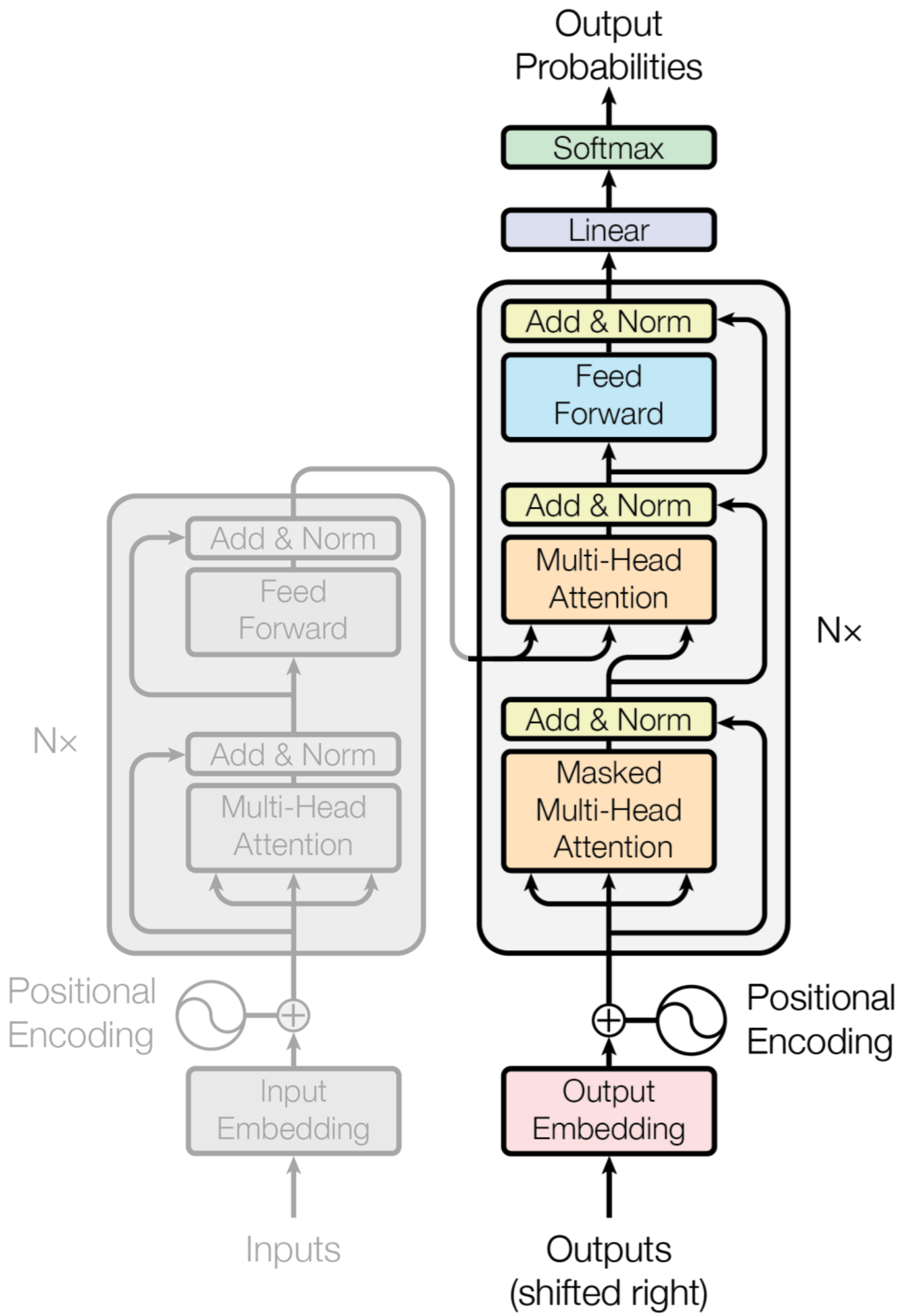


English-French Translation

The big red dog
Le gros chien rouge

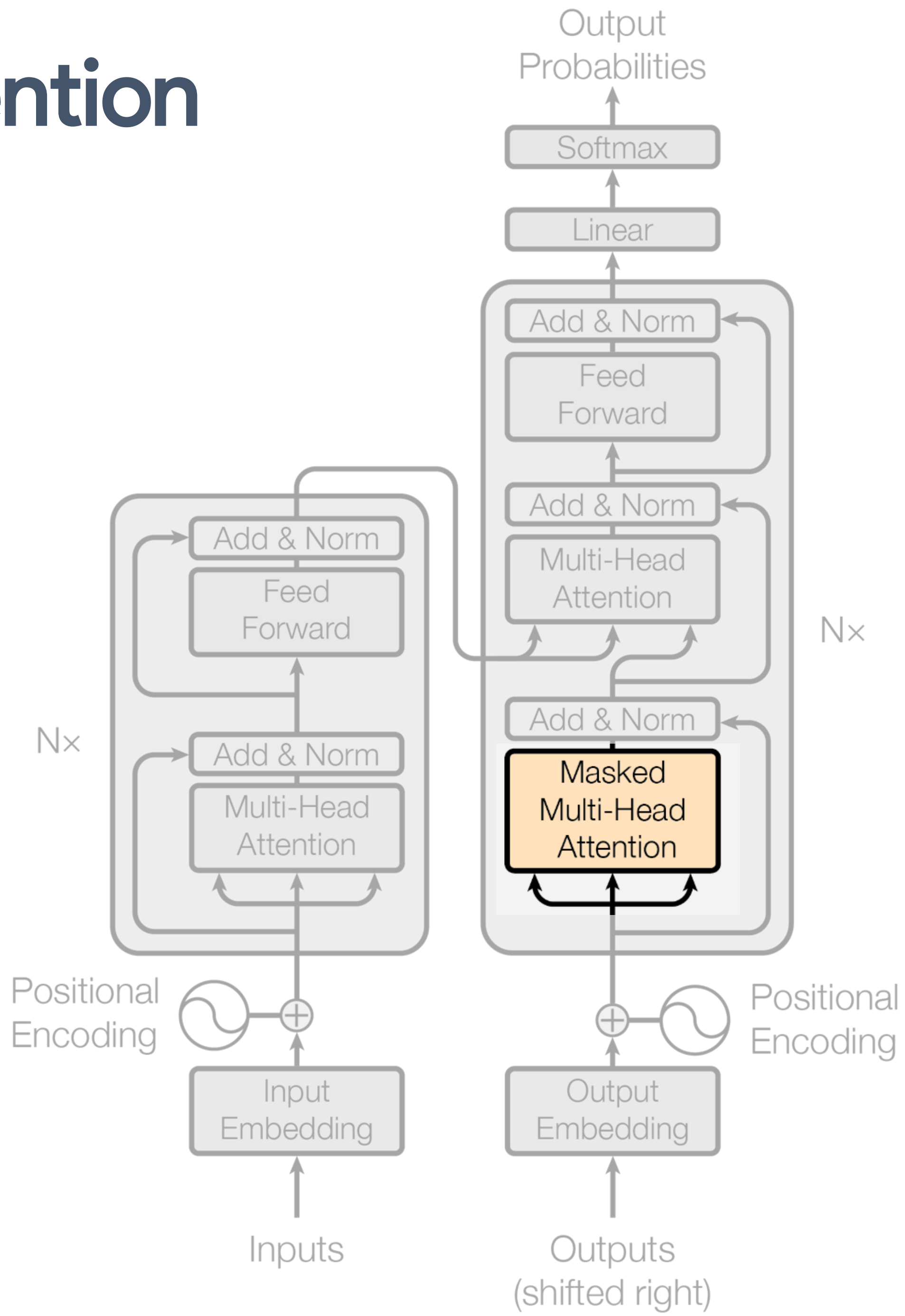


Decoder

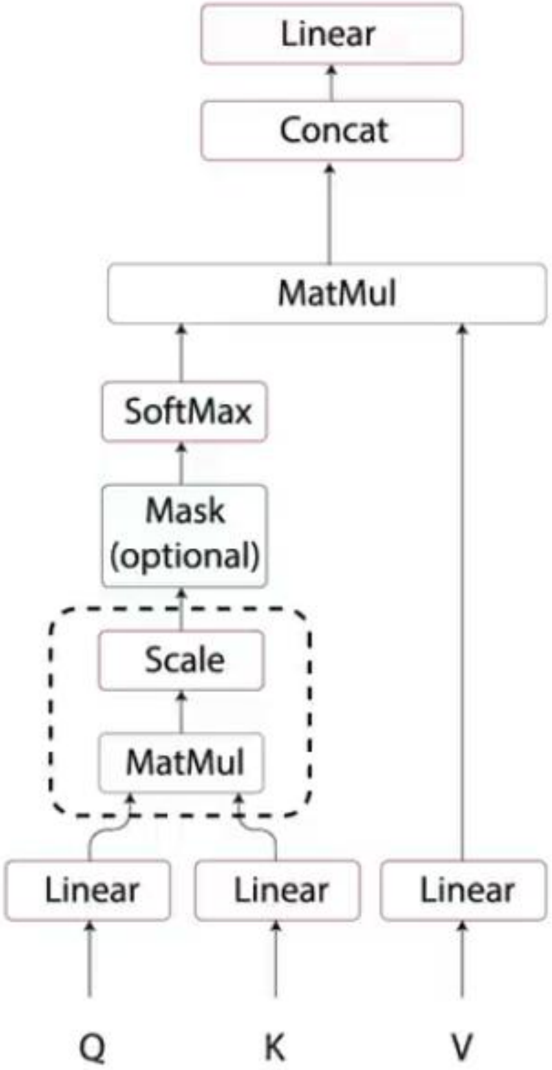


Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Add & Norm
Masked
Multi-Head
Attention
Positional
Encoding
Output
Embedding
Outputs
(shifted right)
Add & Norm
Feed
Forward
Nx
Add & Norm
Multi-Head
Attention
Positional
Encoding
Input
Embedding
Inputs

Masked Self-Attention

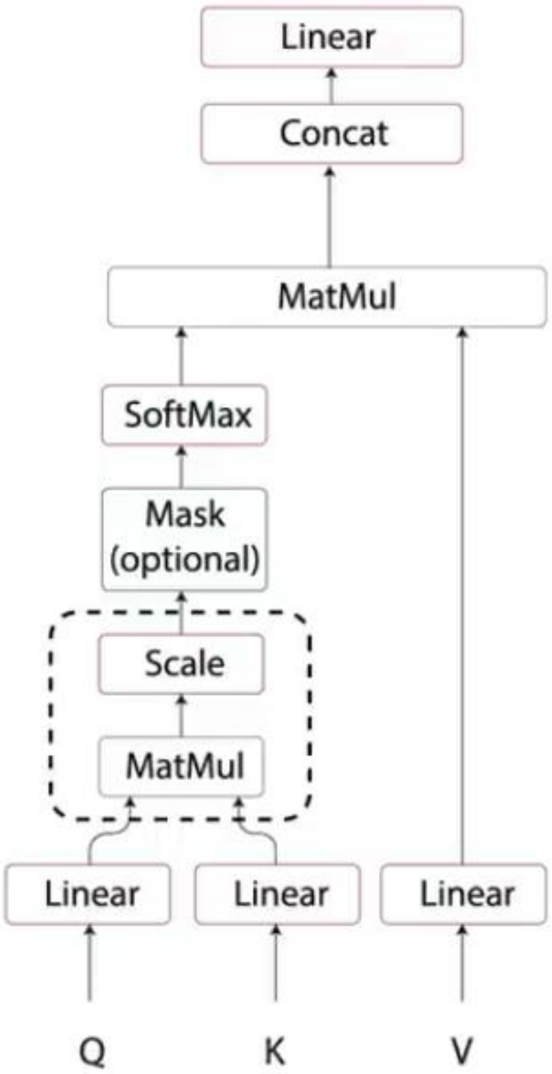


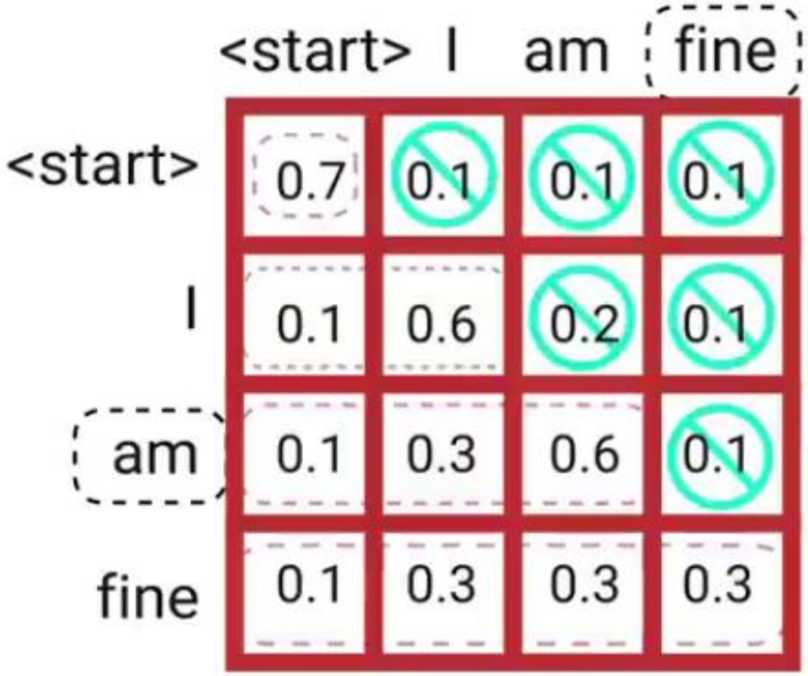
Masked Self-Attention



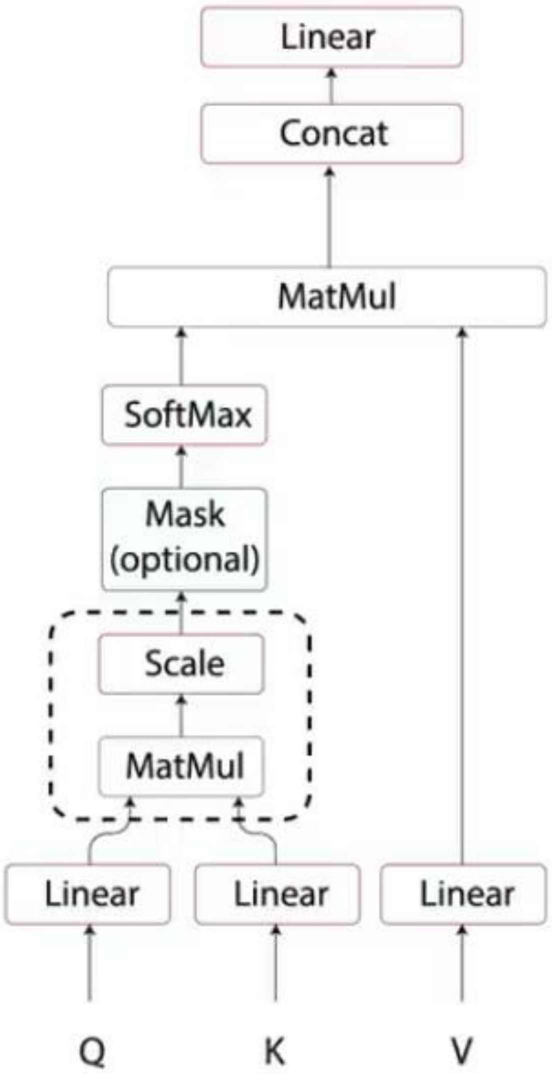
	<start>	I	am	fine
<start>	0.7	0.1	0.1	0.1
I	0.1	0.6	0.2	0.1
am	0.1	0.3	0.6	0.1
fine	0.1	0.3	0.3	0.3

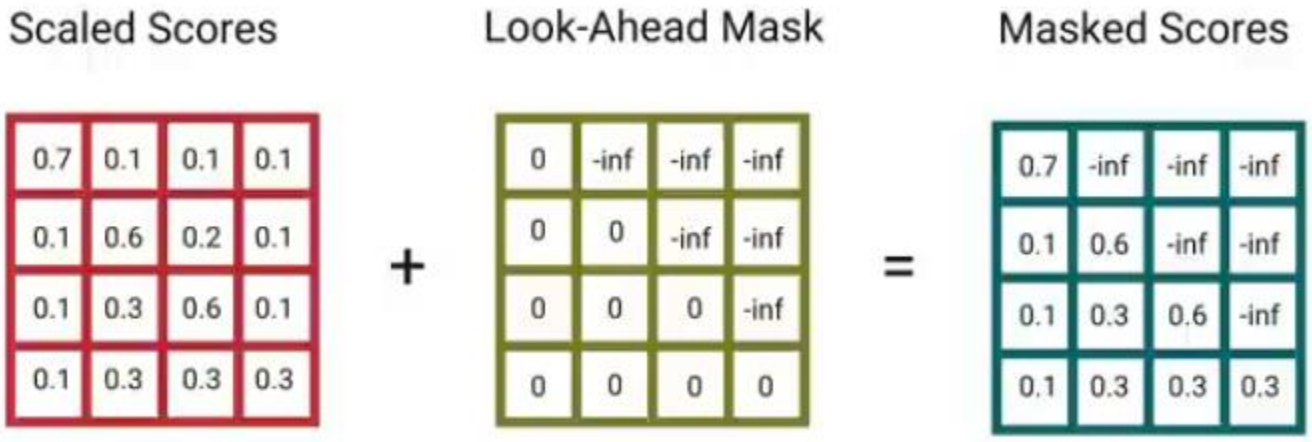
Masked Self-Attention



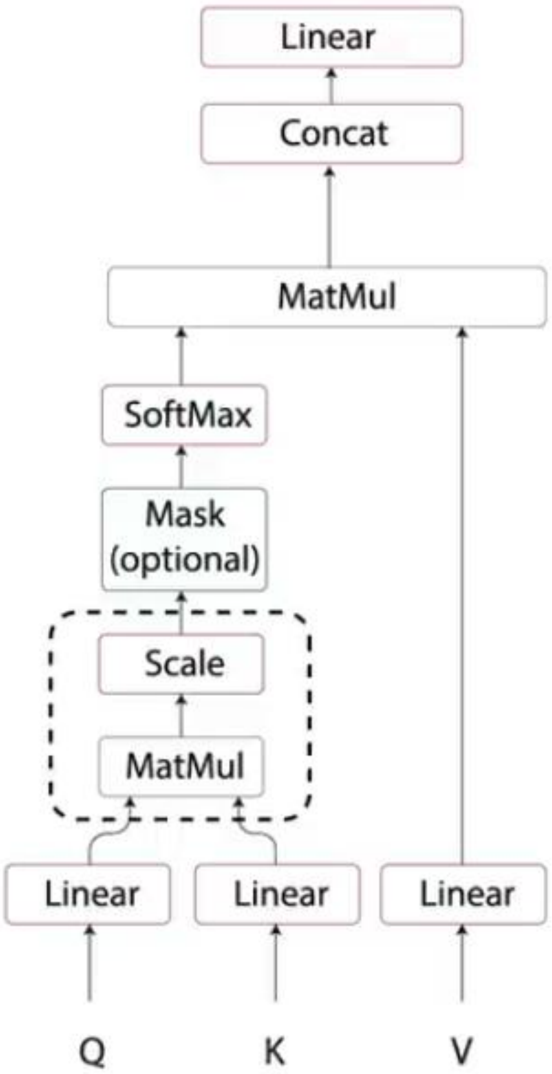


Masked Self-Attention



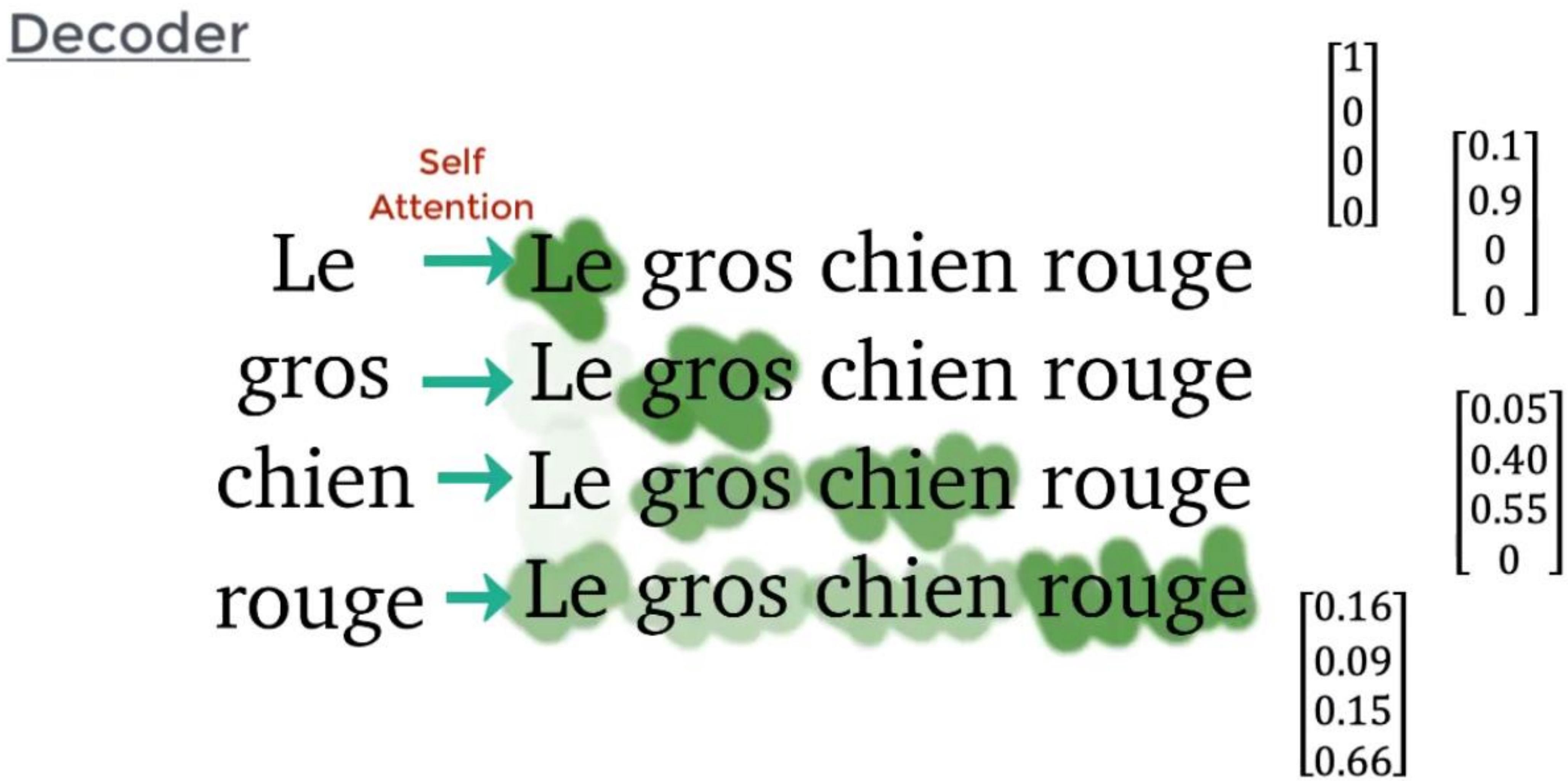


Masked Self-Attention

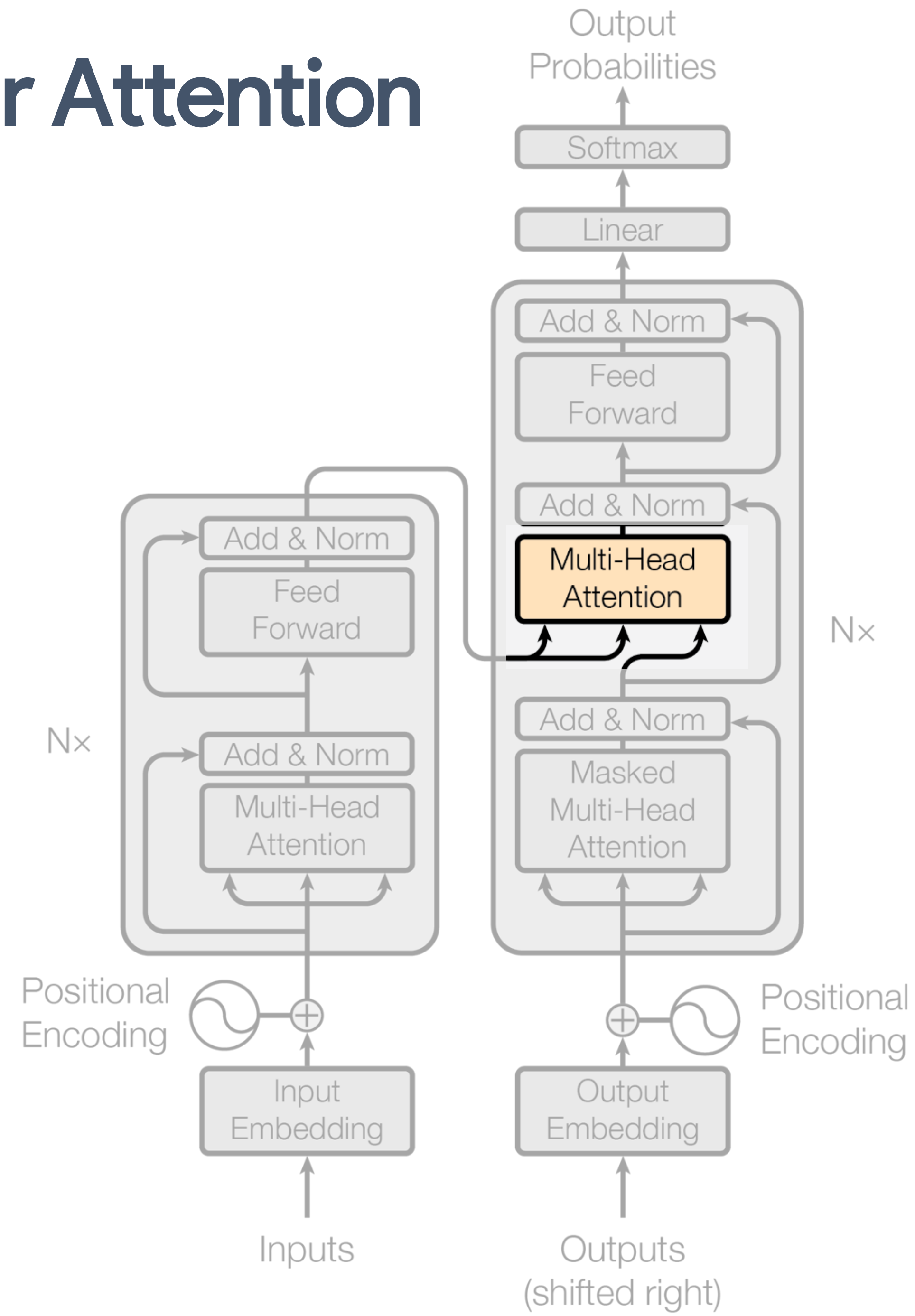


$$\mathrm{Softmax}\left(\begin{bmatrix} 0.7 & -inf & -inf & -inf \\ 0.1 & 0.6 & -inf & -inf \\ 0.1 & 0.3 & 0.6 & -inf \\ 0.1 & 0.3 & 0.3 & 0.3 \end{bmatrix}\right) = \begin{array}{cc} & \begin{matrix} \text{<start>} & \text{I} & \text{am} & \text{fine} \end{matrix} \\ \begin{matrix} \text{<start>} \\ \text{I} \\ \text{am} \\ \text{fine} \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.37 & 0.62 & 0 & 0 \\ 0.26 & 0.31 & 0.43 & 0 \\ 0.21 & 0.26 & 0.26 & 0.26 \end{bmatrix} \end{array}$$

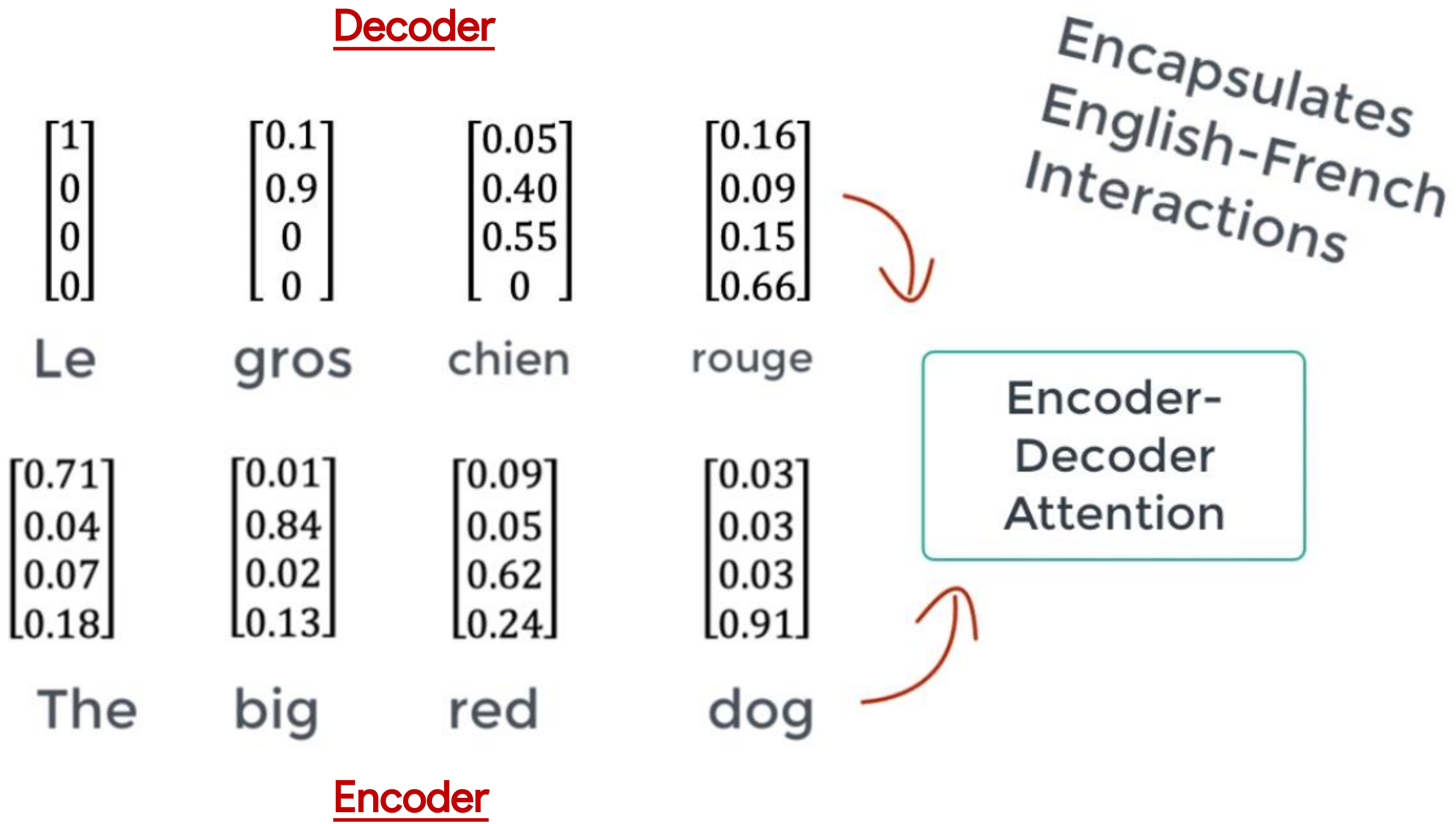
Masked Self-Attention



Encoder-Decoder Attention

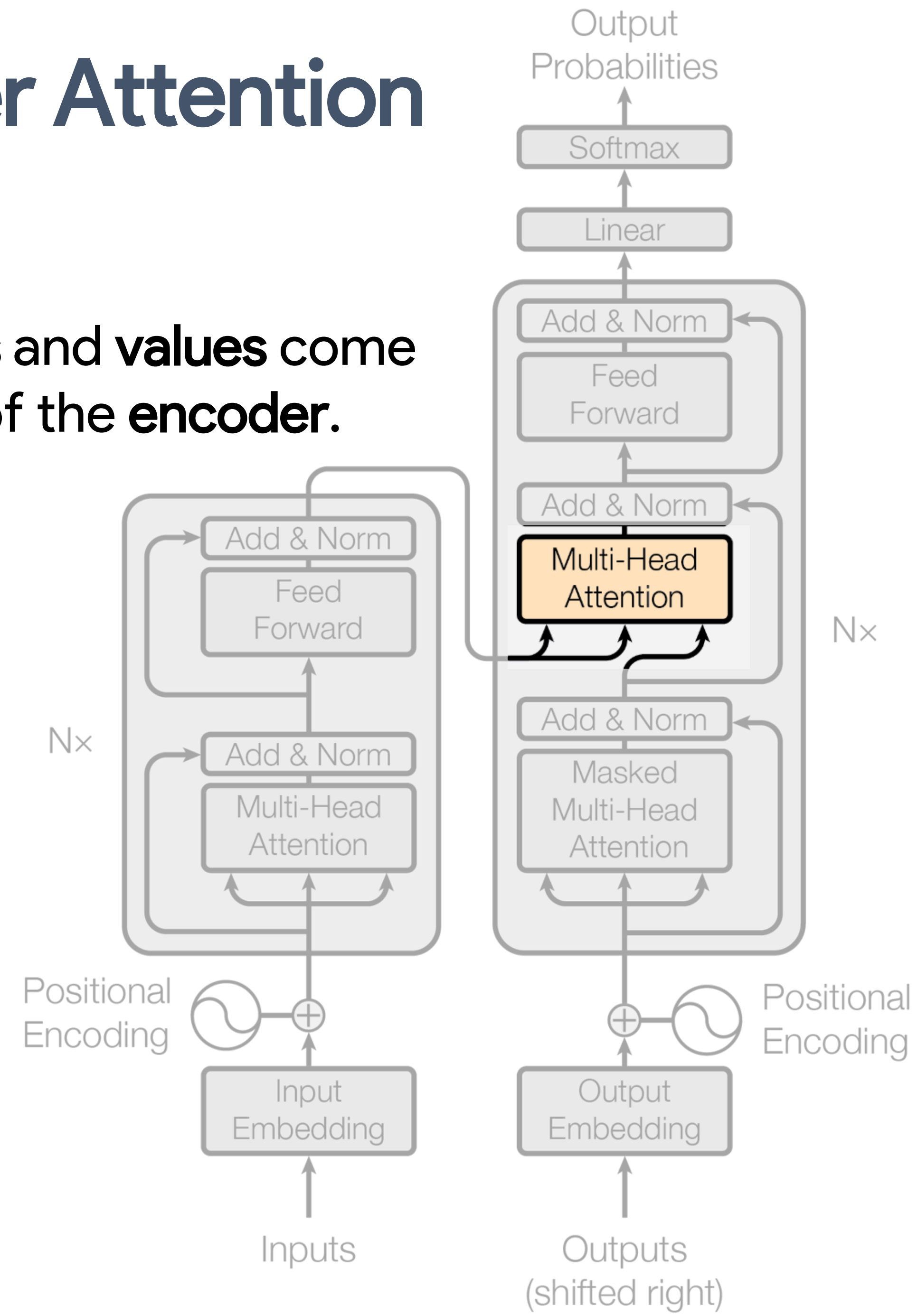


Encoder-Decoder Attention



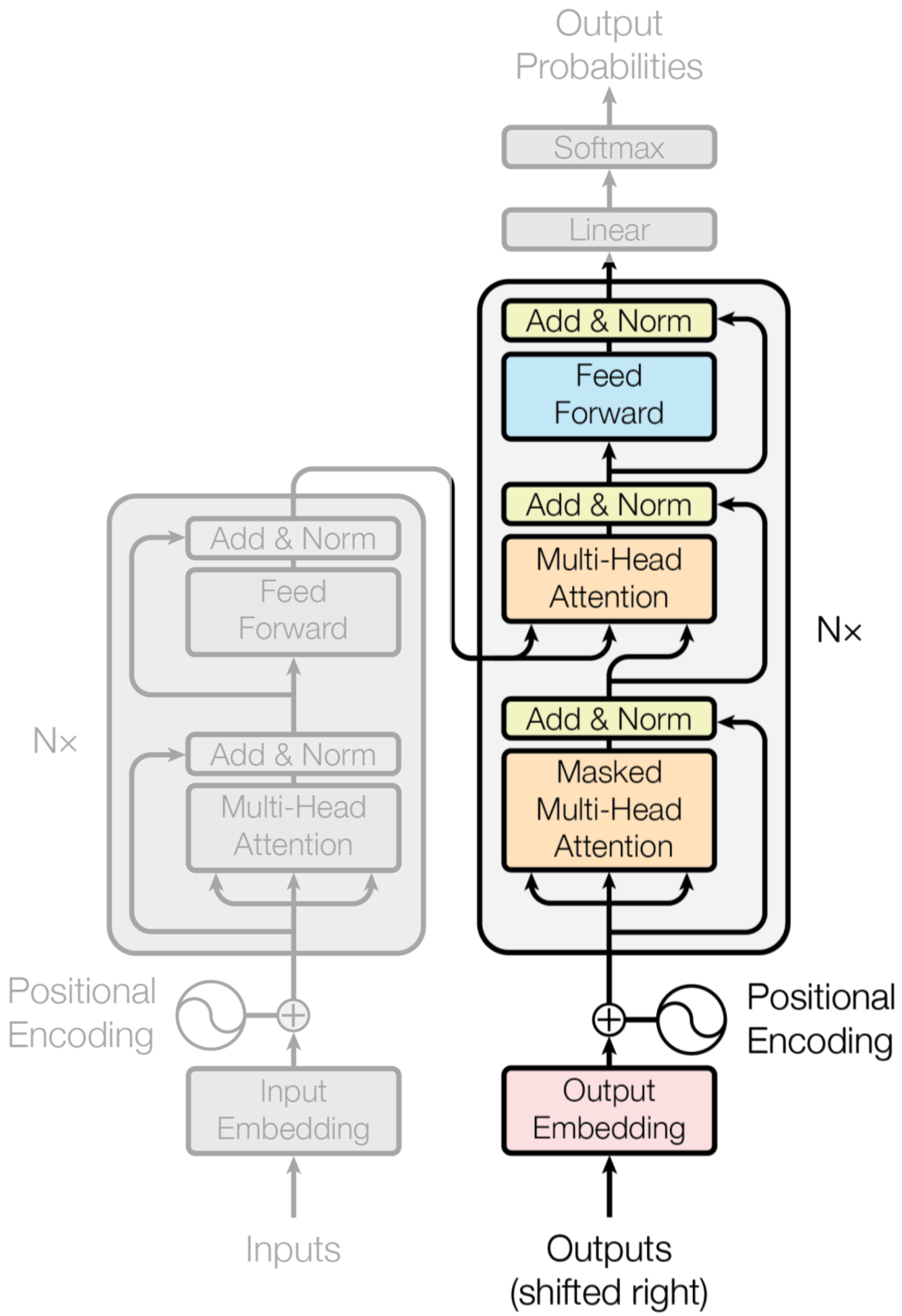
Encoder-Decoder Attention

The memory **keys** and **values** come from the output of the **encoder**.



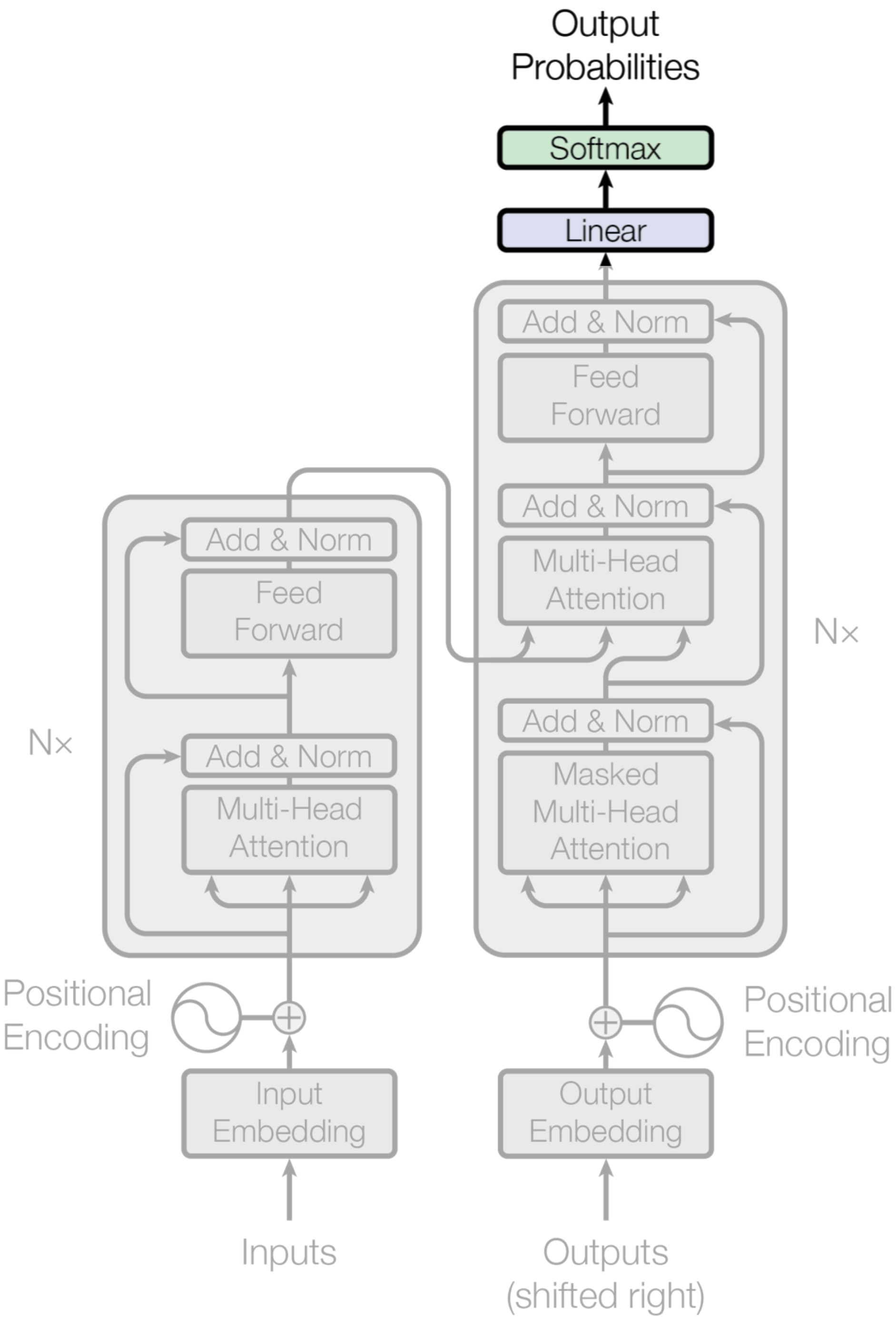
The **queries** come from the previous **decoder** layer.

Decoder



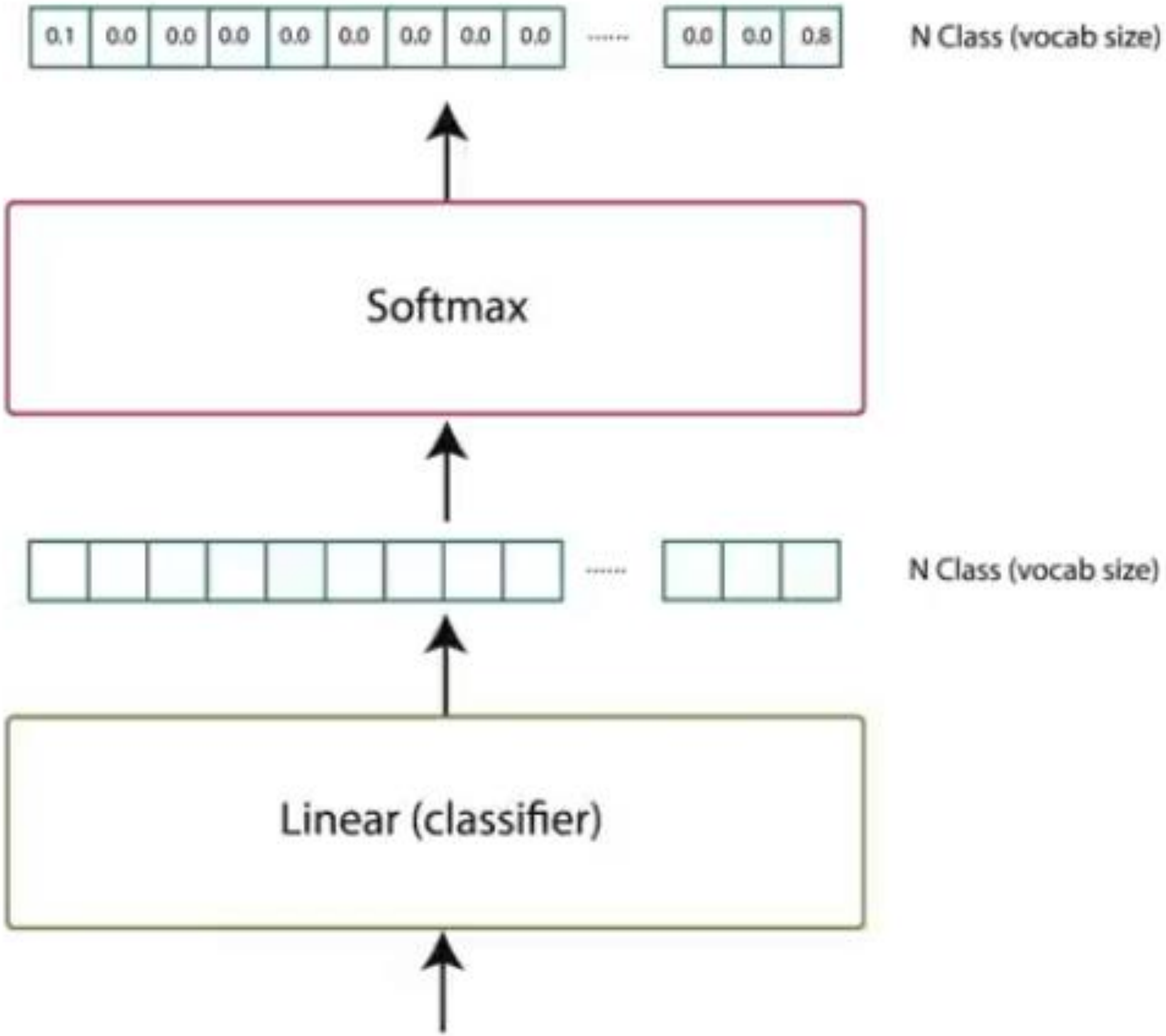
Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
N×
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
N×
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

Decoder



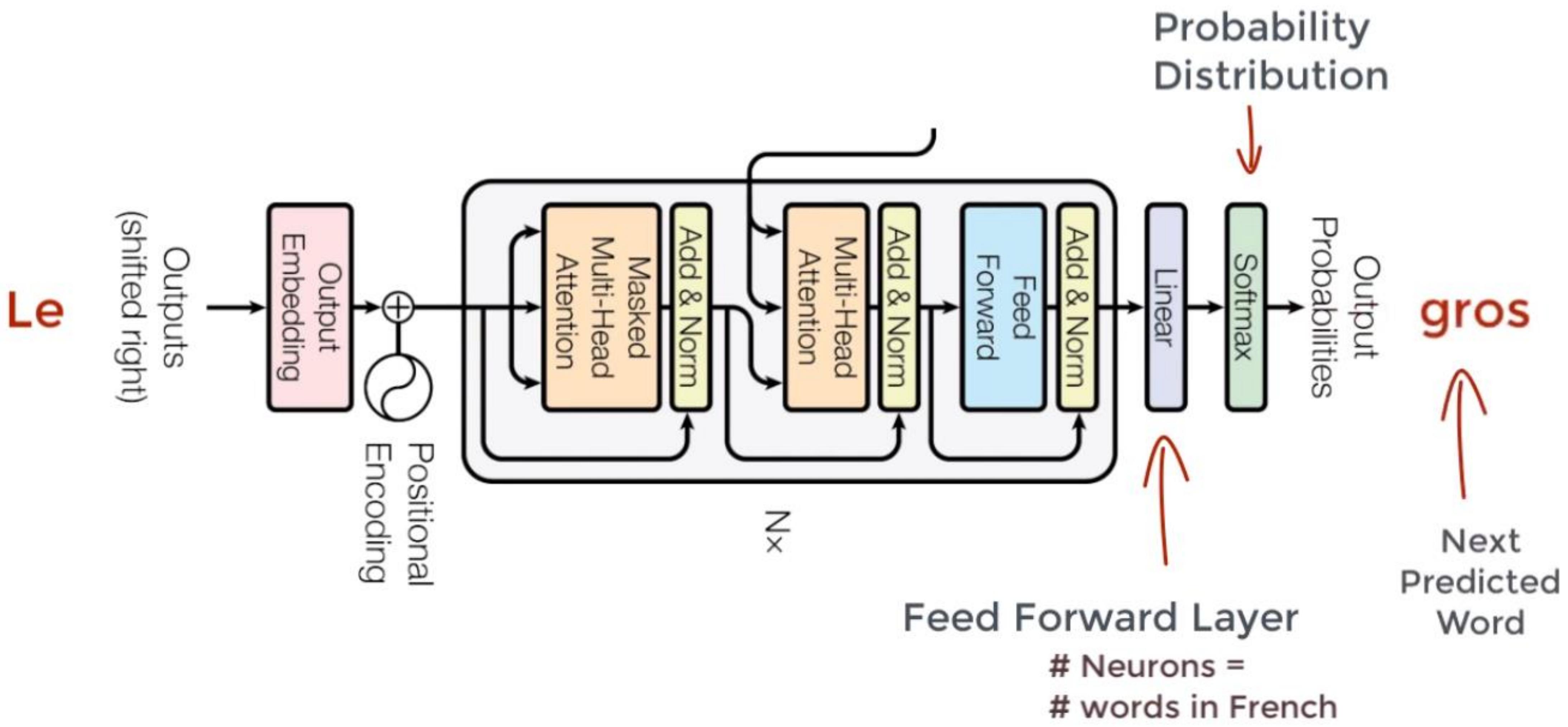
Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
Nx
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)

Decoder


0.1
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.8
N Class (vocab size)
Softmax
N Class (vocab size)
Linear (classifier)

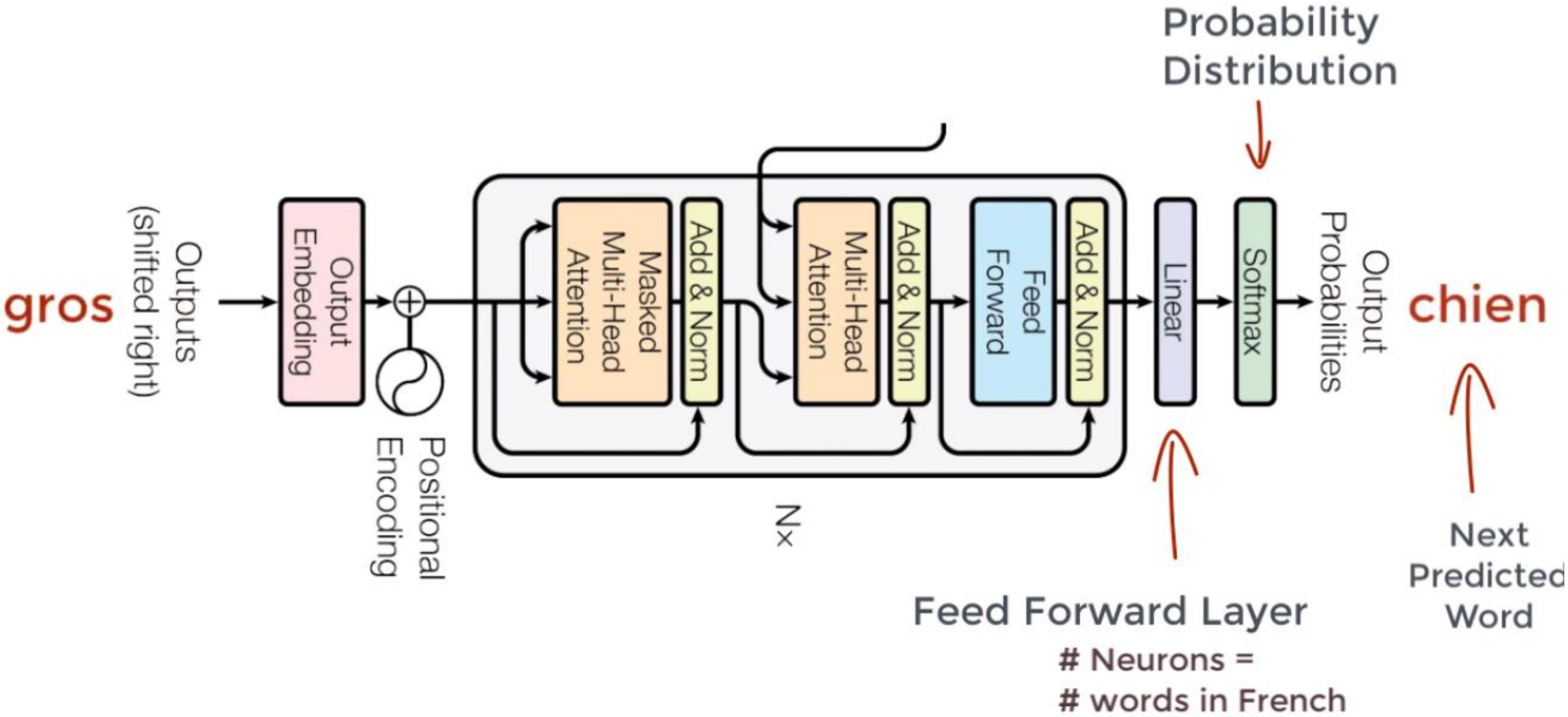
Decoder

The big red dog
Le gros chien rouge



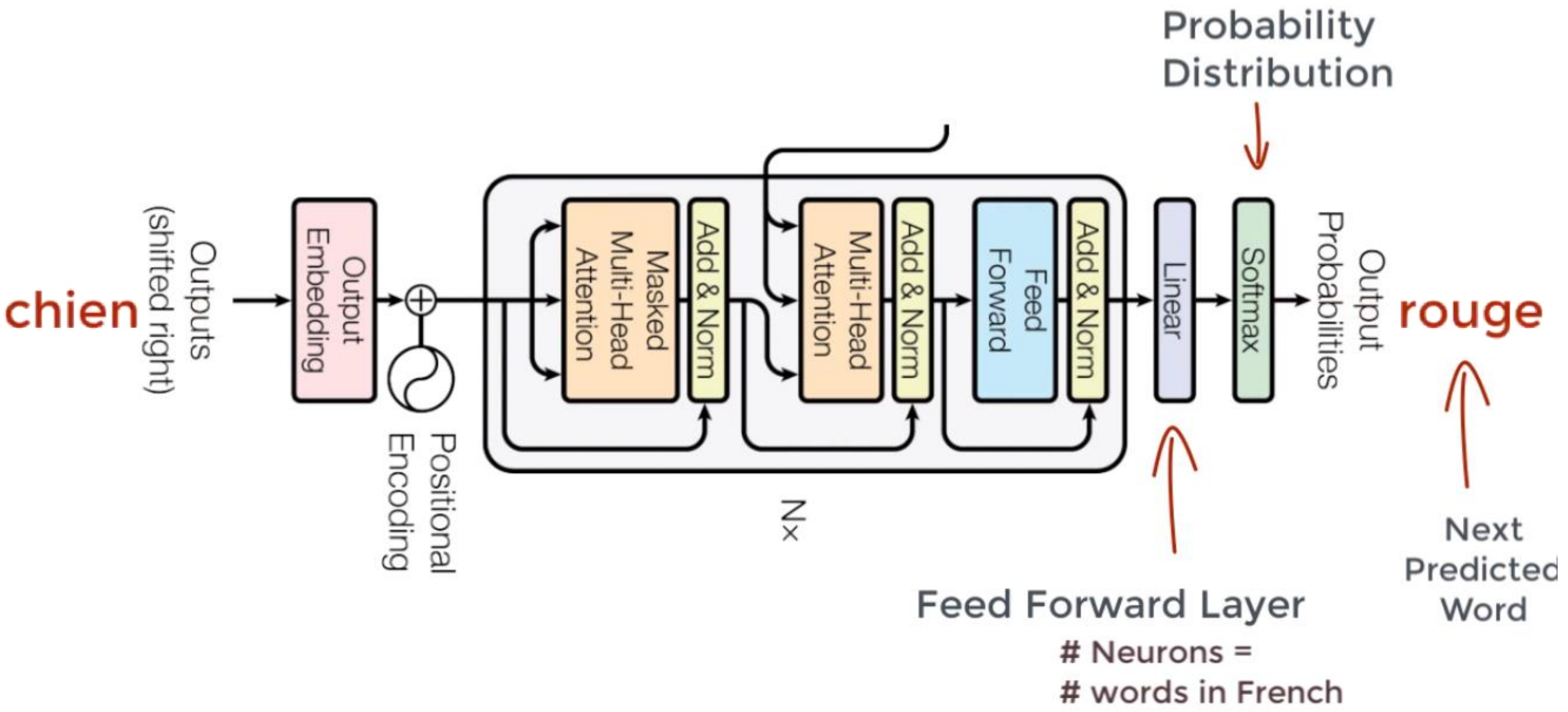
Decoder

The big red dog
Le gros chien rouge



Decoder

The	big	red	dog
Le	gros	chien	rouge



Transformers
Encoder

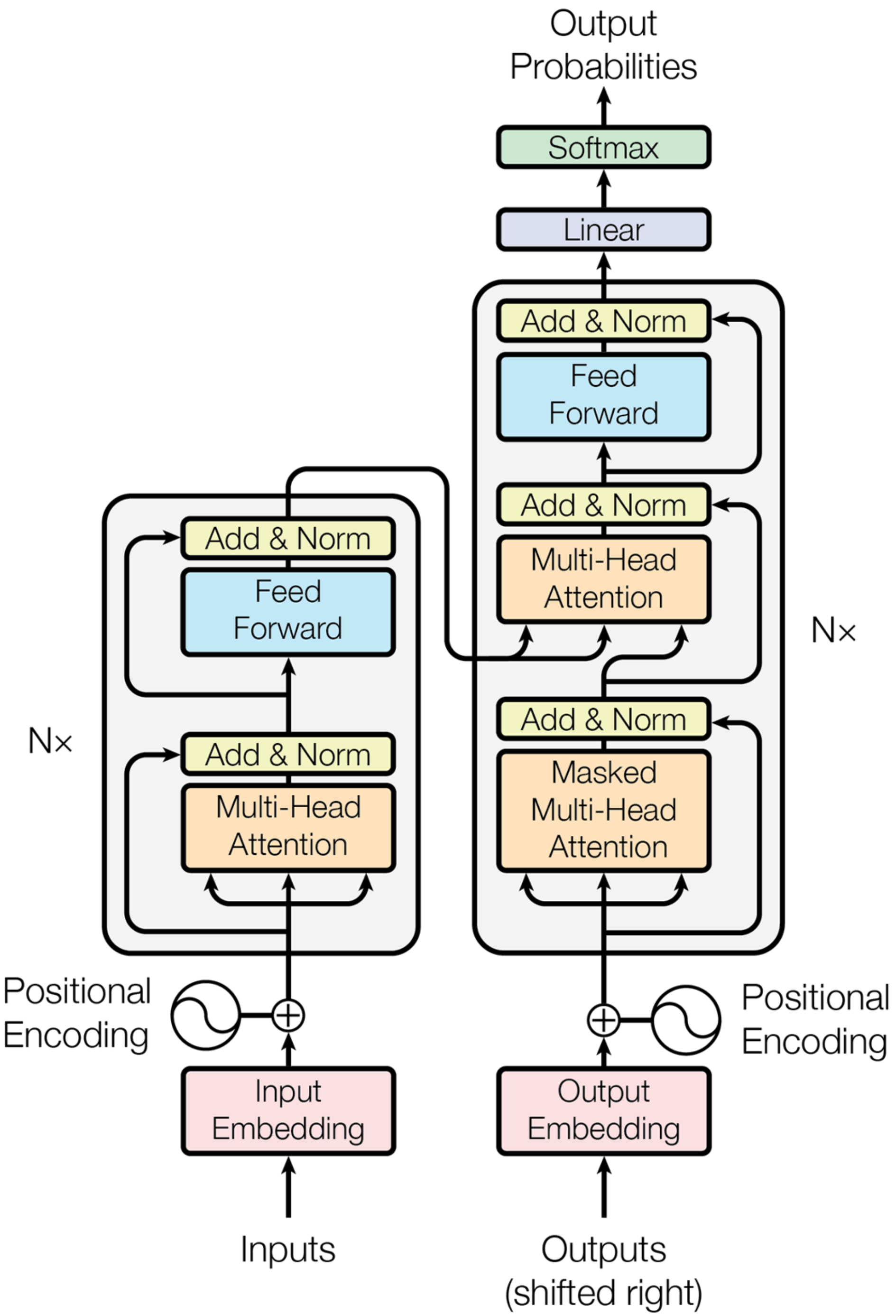
Transformers
Decoder

Results

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	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**	**41.0**	$2.3 \cdot 10^{19}$	

Thank you



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