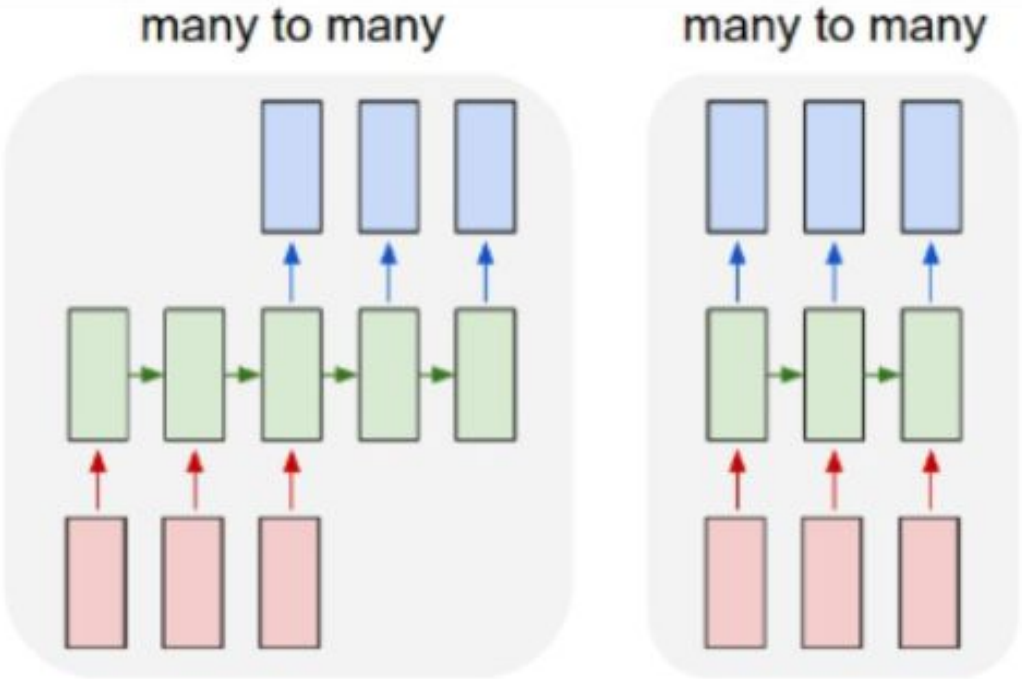
Transformers



Some suffer from vanishing gradient and inability to capture long dependencies

RNNs and LSTMs

Typical choices for time-series but miss parallelization

Dialogue Completer

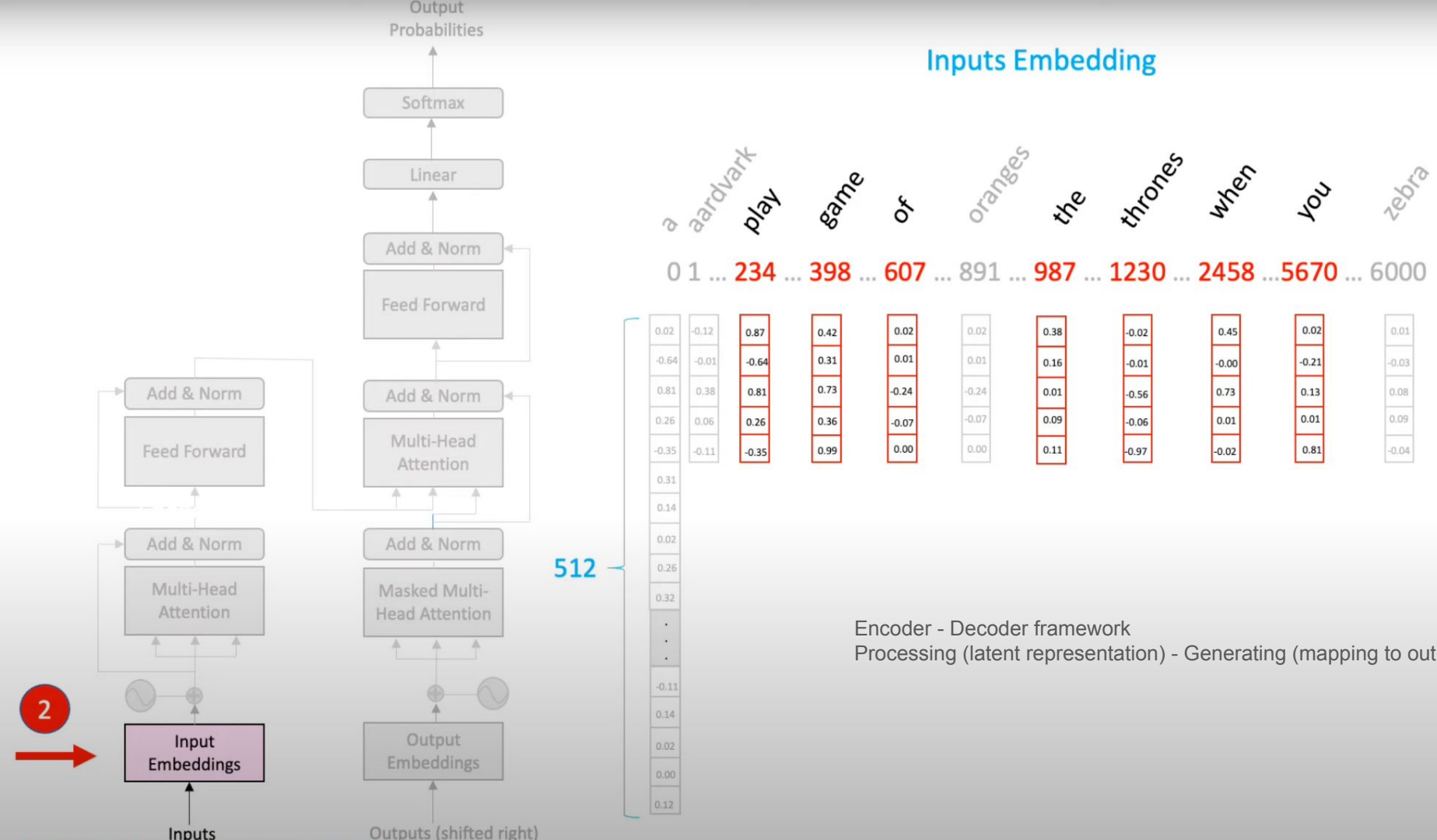
Transformers!

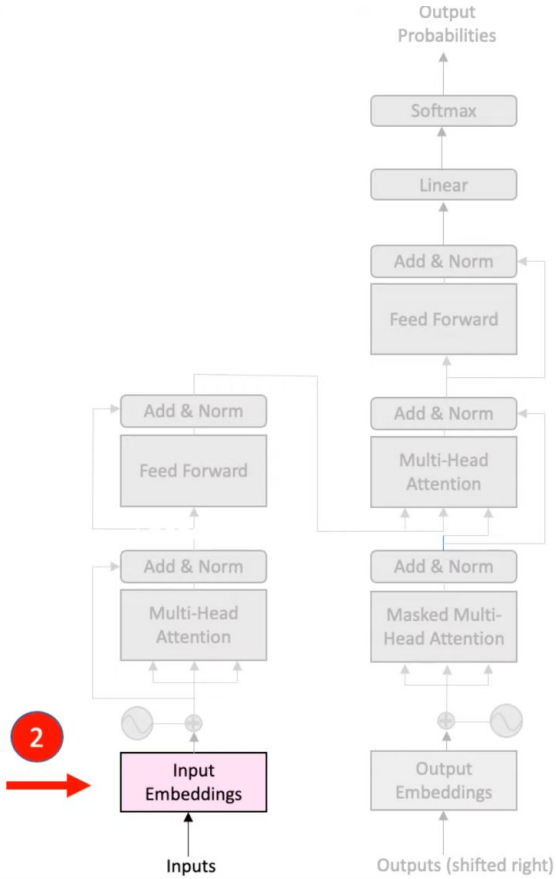
Input When you play the Game of Thrones …

Transformer Output ?

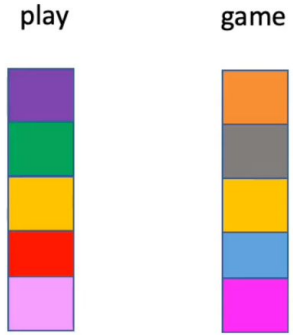
Training a Dialogue Completer

Dialogue PART 1	Dialogue PART 2
When you play the games of thrones	<start> you win or you die <end>
It is not our abilities that show who we truly are	<start> it is our choices <end>
Life happens where ever you are	<start> whether you make it or not <end>
All we have to do is decide what to do	<start> with the time that has been given to us <end>
There is some good in this world Mr. Frodo	<start> and it is worth fighting for <end>

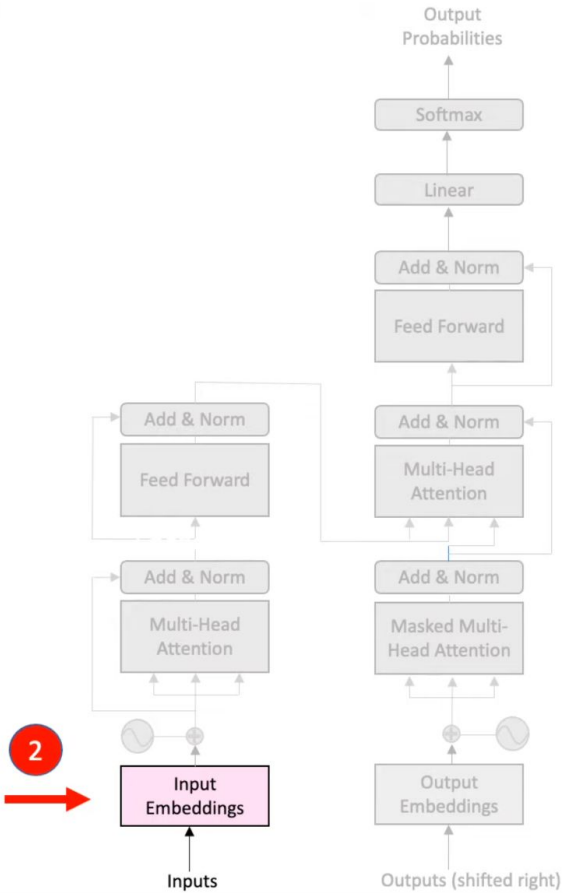

Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
2
Input Embeddings
Output Embeddings
Inputs
Outputs (shifted right)
Inputs Embedding
a aardvark play game of oranges the thrones when you zebra
0 1 ... 234 ... 398 ... 607 ... 891 ... 987 ... 1230 ... 2458 ...5670 ... 6000
512
Encoder - Decoder framework
Processing (latent representation) - Generating (mapping to output)



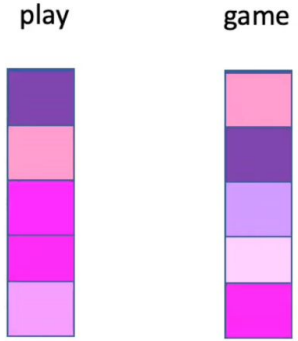
Inputs Embedding



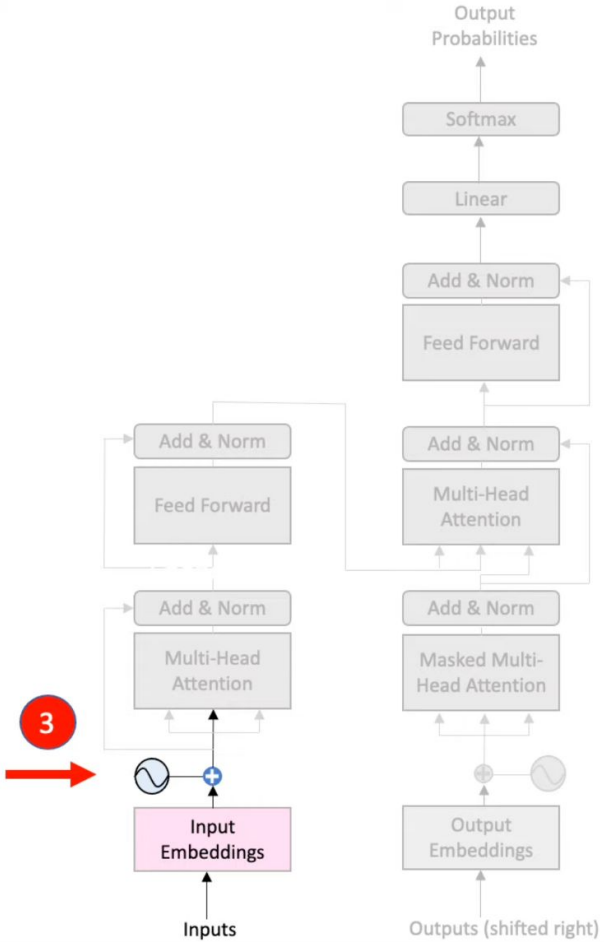
Embeddings can be randomly initialized



Inputs Embedding



Embeddings keep changing



Position Embeddings (Intuition)

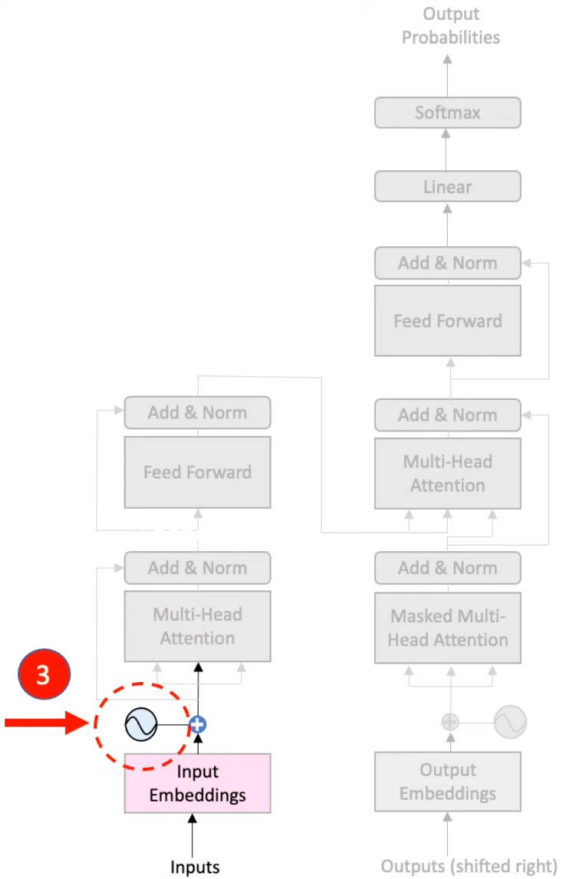
Here is why order matters

Even though she did not win the award, she was satisfied.

Even though she did win the award, she was not satisfied.

Why are we adding position embeddings only in transformers?

Position embedding = 1/(N-1) ??



Position Embeddings

Sentence 1

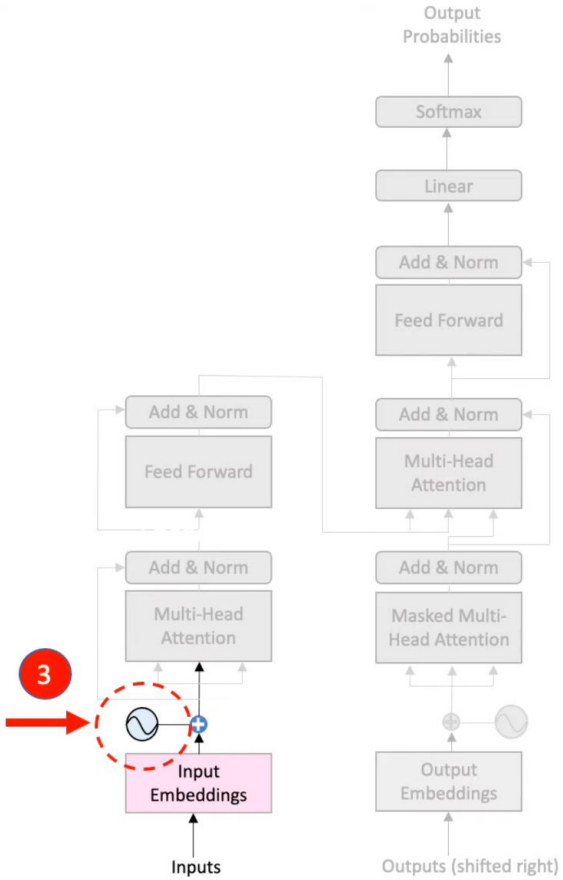
N=4

e_0	p_0	e_1	p_1	e_2	p_2	e_3	p_3
0.42	0	0.87	0.33	0.02	0.66	0.02	1
0.31	0	-0.64	0.33	0.01	0.66	0.01	1
0.73	0	0.81	0.33	-0.24	0.66	-0.24	1
0.36	0	0.26	0.33	-0.07	0.66	-0.07	1
0.99	0	-0.35	0.33	0.00	0.66	0.00	1

Sentence 2

N=5

e_0	p_0	e_1	p_1	e_2	p_2	e_4	p_4
0.42	0	0.87	0.20	0.02	0.40	0.02	1
0.31	0	-0.64	0.20	0.01	0.40	0.01	1
0.73	0	0.81	0.20	-0.24	0.40	-0.24	1
0.36	0	0.26	0.20	-0.07	0.40	-0.07	1
0.99	0	-0.35	0.20	0.00	0.40	0.00	1



Position Embeddings

What about frequencies?

e_0	p_0	e_1	p_2	e_2	p_1	e_3	p_3
0.42	?	0.87	?	0.02	?	0.02	?
0.31	?	-0.64	?	0.01	?	0.01	?
0.73	?	0.81	?	-0.24	?	-0.24	?
0.36	?	0.26	?	-0.07	?	-0.07	?
0.99	?	-0.35	?	0.00	?	0.00	?

Vaswani et al 2017

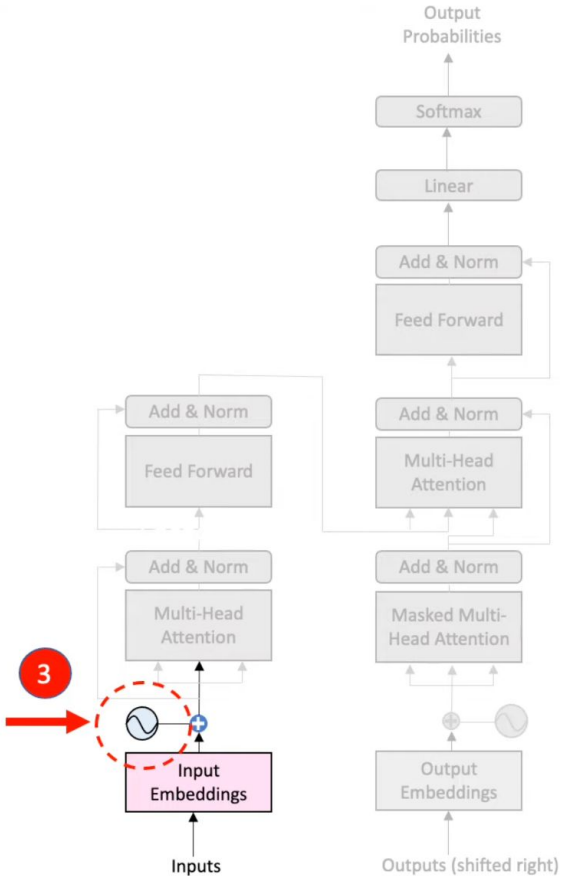
$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

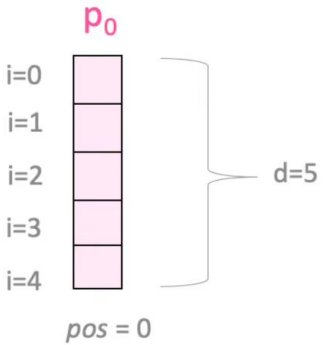
Encode sentences longer than N used in training

Something with fixed range of output (not much dependant on N)

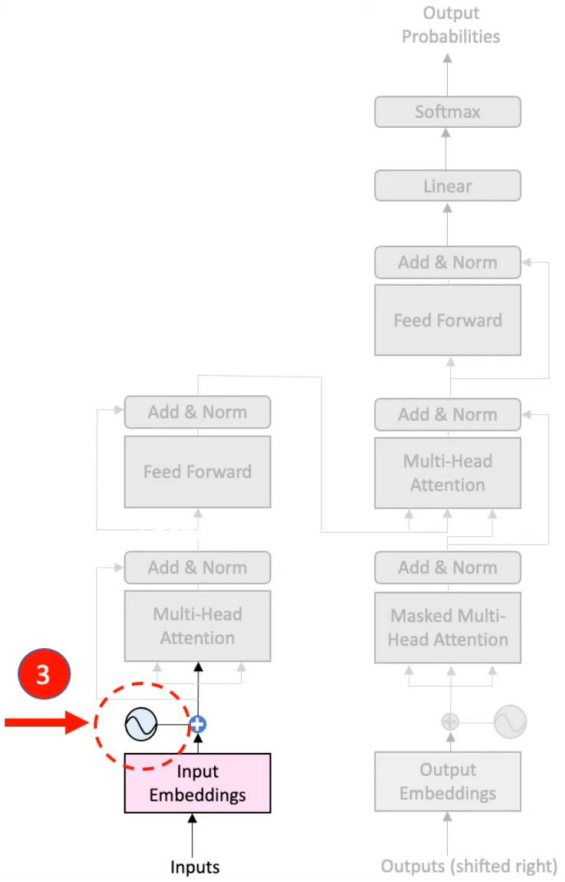
Relative position matters!

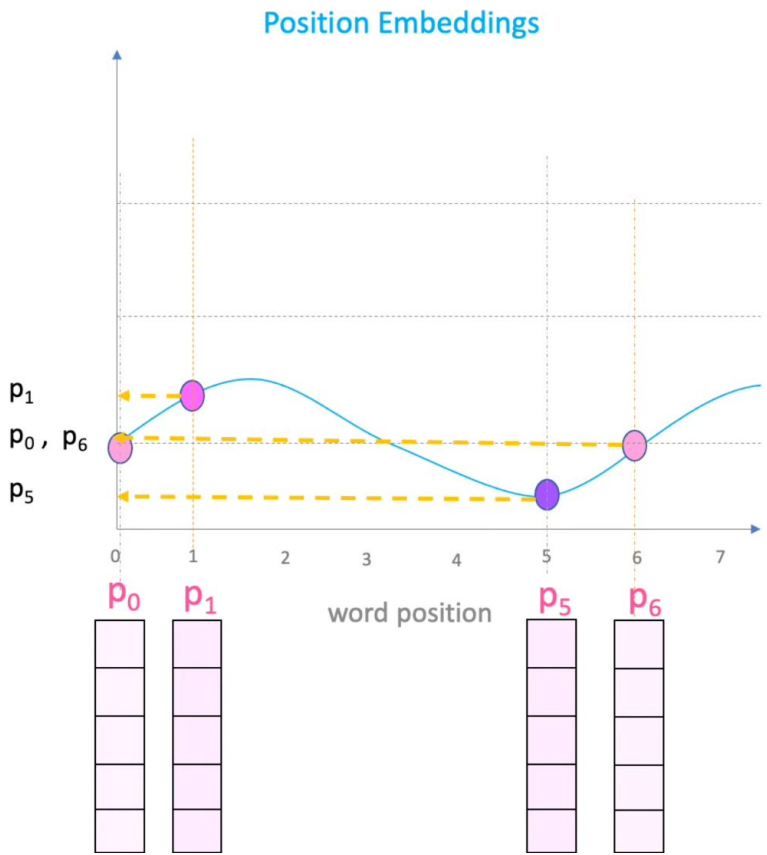


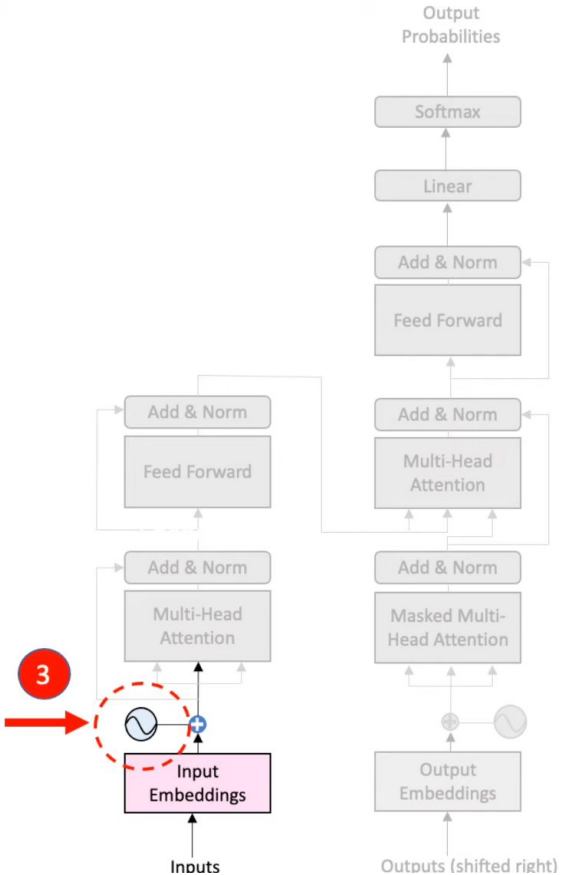
Position Embeddings

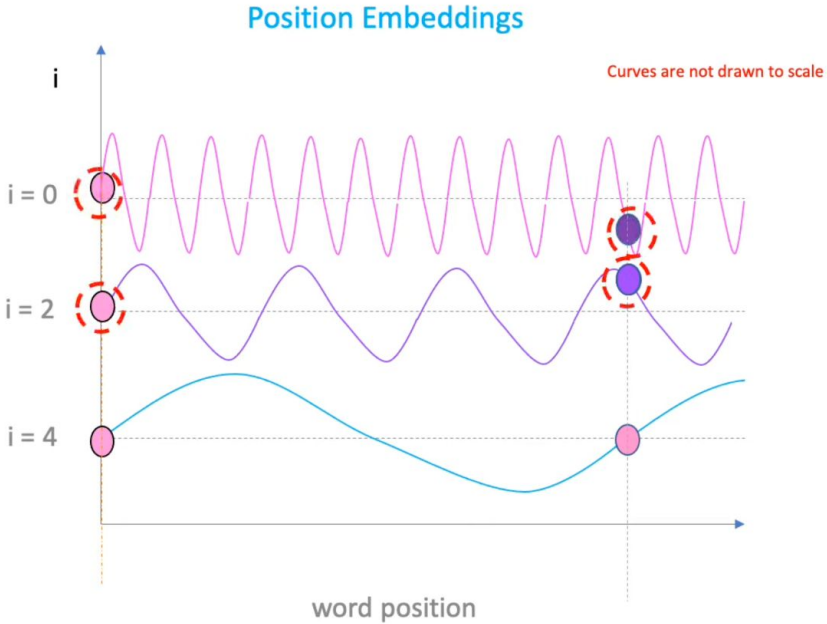


$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Output Embeddings
Outputs (shifted right)
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Input Embeddings
Inputs
3


Position Embeddings
p1
p0 , p6
p5
0
1
2
3
4
5
6
7
word position
p0
p1
p5
p6





$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

i

She faced her enemies and whispered - DRACARYS

Which GOT character is referred here?

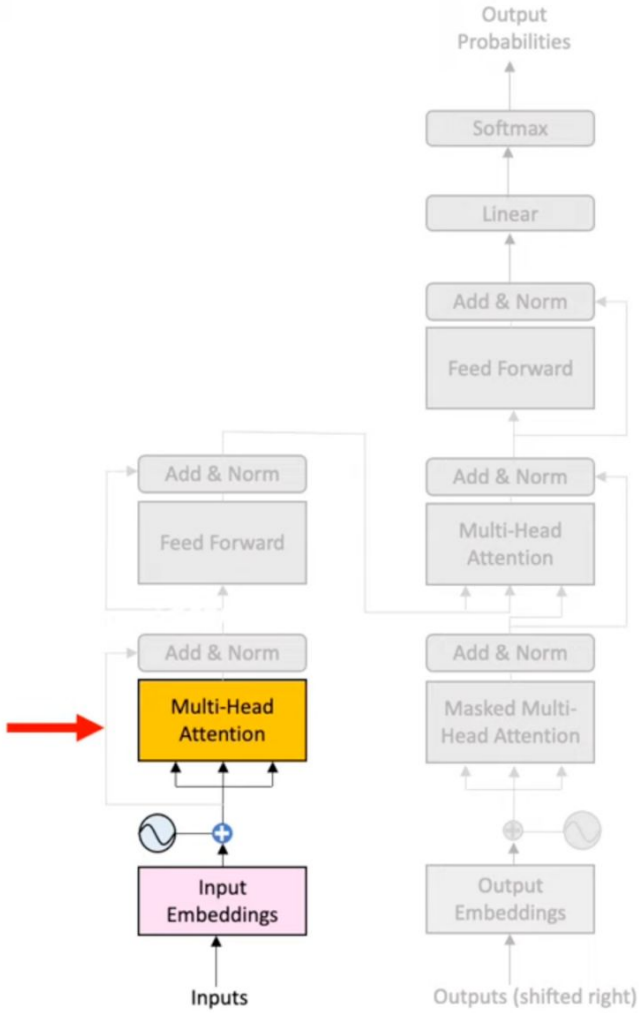
Why do you think so?

She faced here enemies and whispered - **Dracarys**

Daenerys Targaryen

Simple idea: Our mind focuses on some words to make a decision!

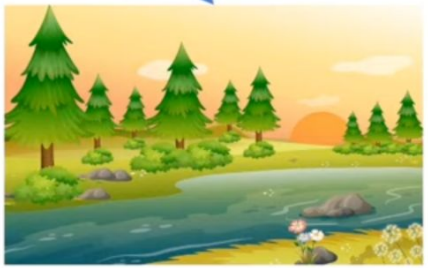
Attention!



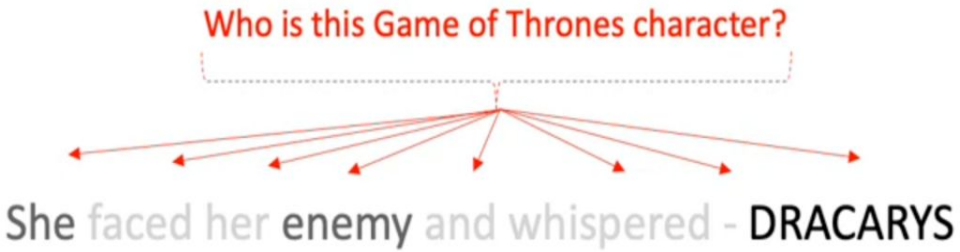
Multi-Head Attention

Self-Attention

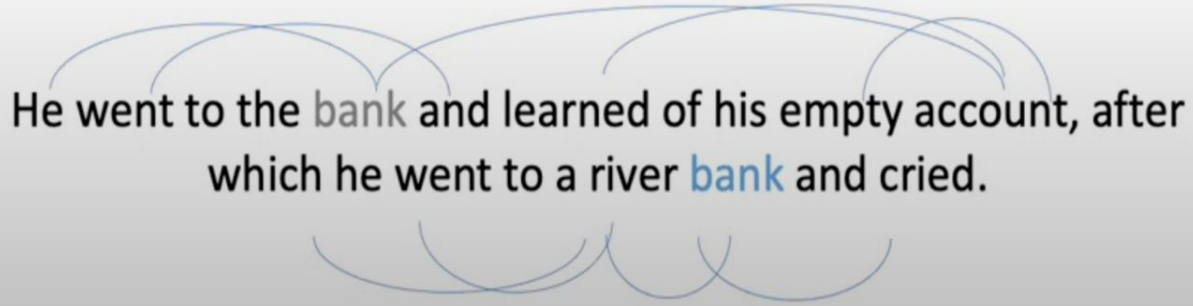
He went to the bank and learned of his empty account, after which he went to a river bank and cried.

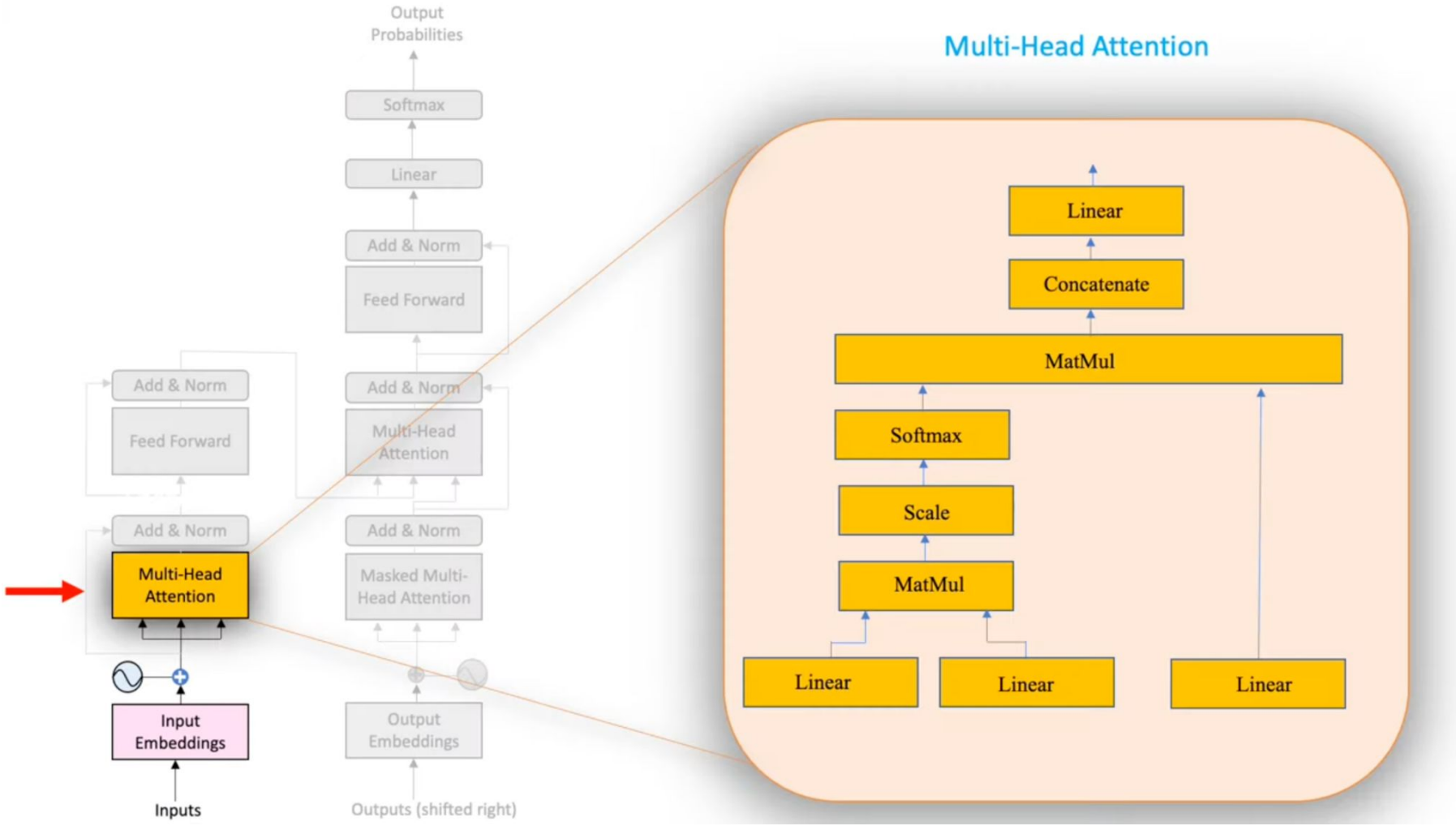


Simple-Attention

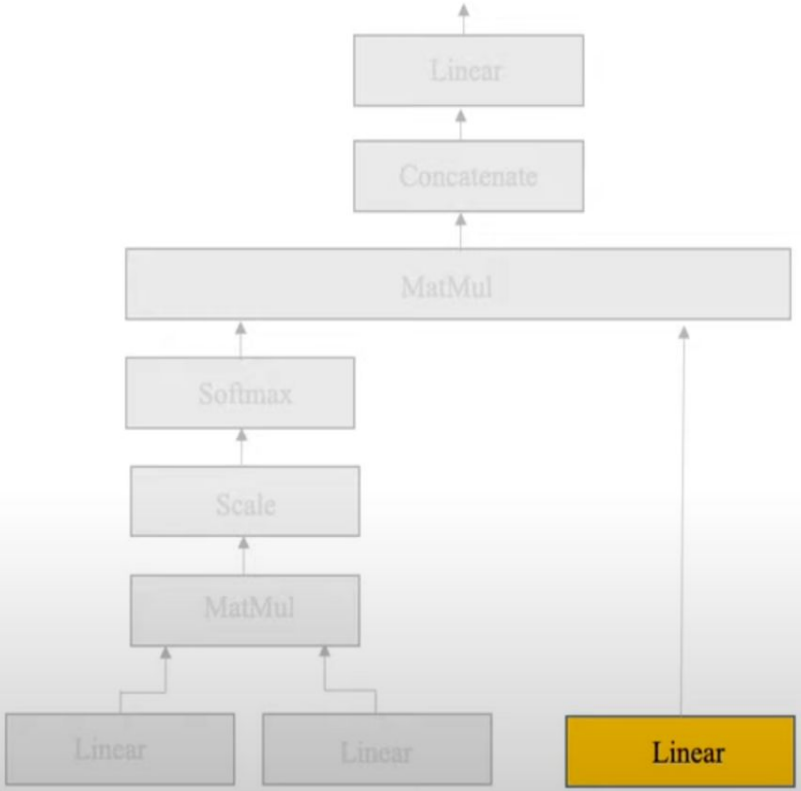


Self-Attention




Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Input Embeddings
Output Embeddings
Inputs
Outputs (shifted right)
Multi-Head Attention
Linear
Concatenate
MatMul
Softmax
Scale
MatMul
Linear
Linear
Linear

Multi-Head Attention

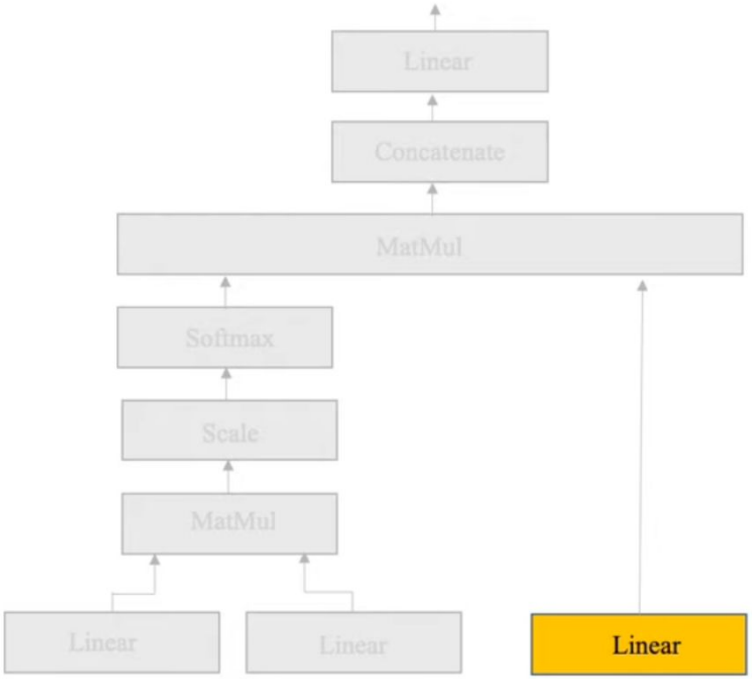


Linear Layer

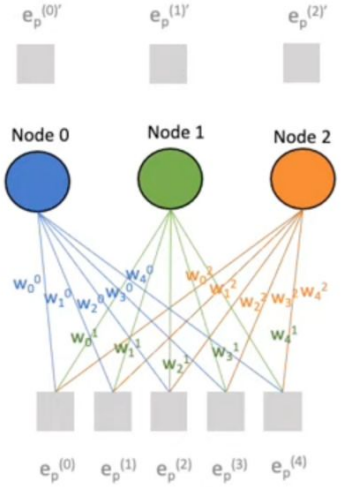
i) Mapping inputs onto the outputs

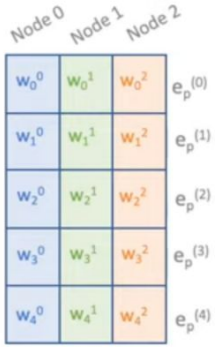
ii) Changing matrix/vector dimensions

Multi-Head Attention

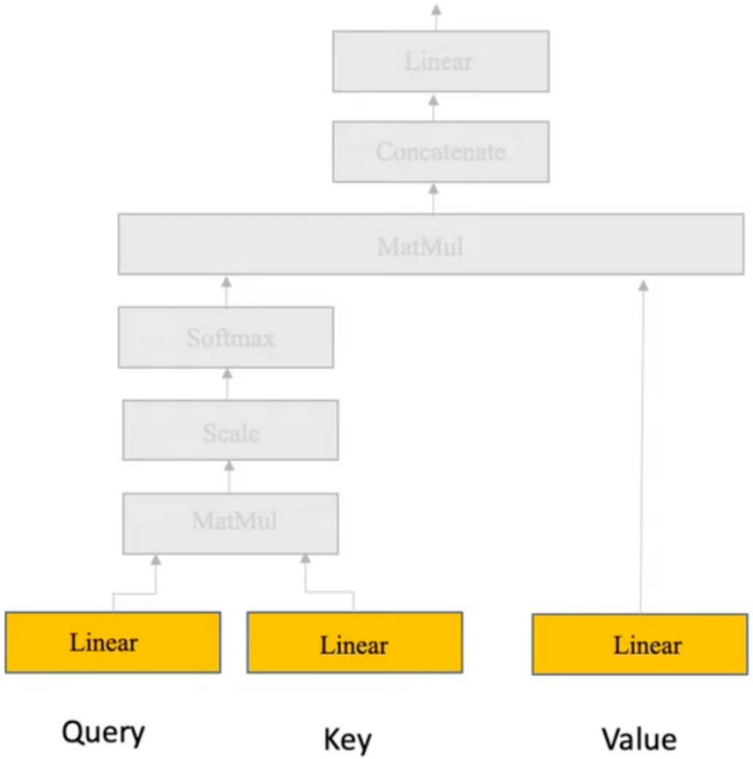


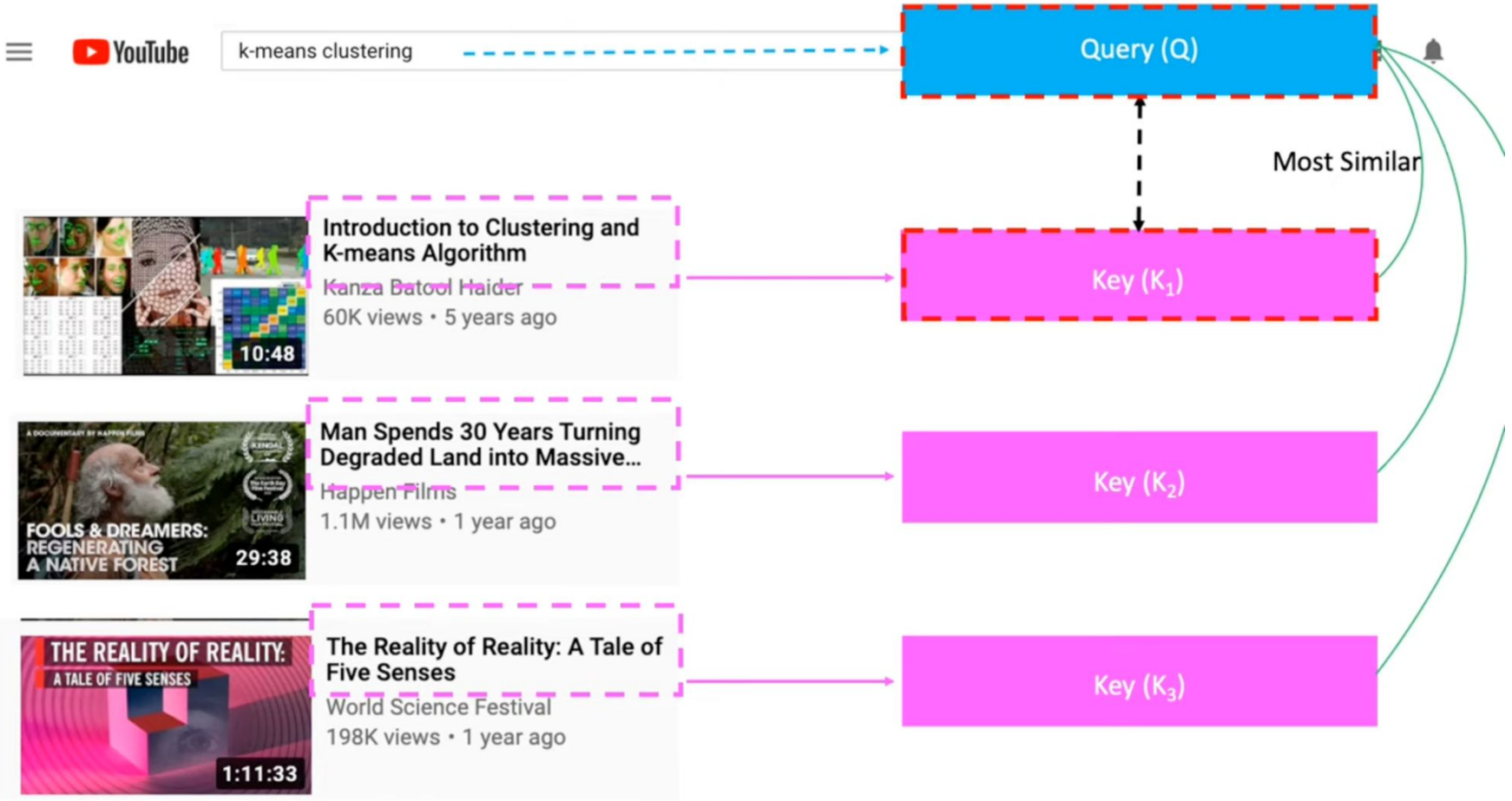
Linear Layer

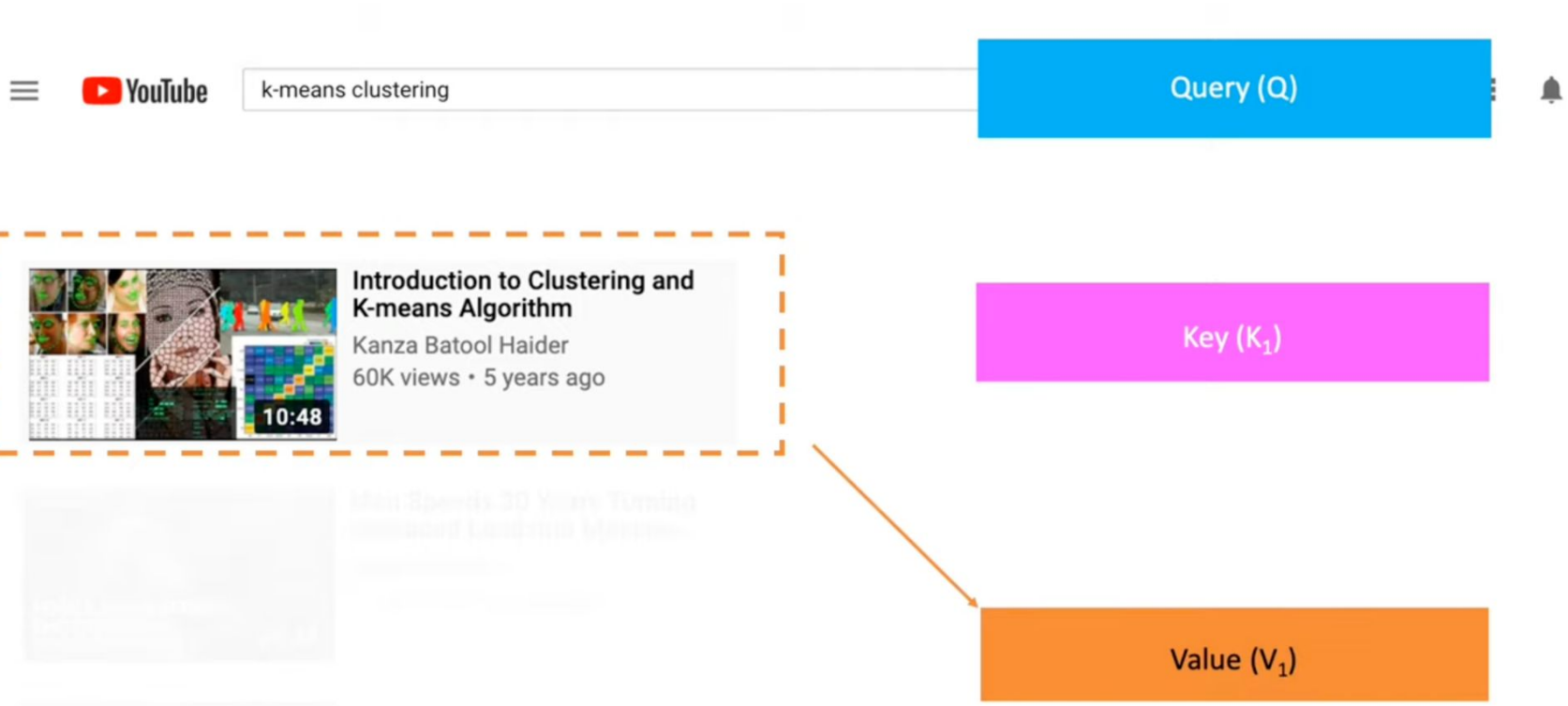




Multi-Head Attention




YouTube
k-means clustering
Query (Q)
Most Similar
10:48
Introduction to Clustering and K-means Algorithm
Kanza Batool Haider
60K views • 5 years ago
Key (K_1)
FOOLS & DREAMERS: REGENERATING A NATIVE FOREST
29:38
Man Spends 30 Years Turning Degraded Land into Massive...
Happen Films
1.1M views • 1 year ago
Key (K_2)
THE REALITY OF REALITY: A TALE OF FIVE SENSES
1:11:33
The Reality of Reality: A Tale of Five Senses
World Science Festival
198K views • 1 year ago
Key (K_3)


YouTube
k-means clustering
Query (Q)
Introduction to Clustering and K-means Algorithm
Kanza Batool Haider
60K views • 5 years ago
10:48
Key (K_1)
Value (V_1)

Cosine Similarity

$$\text{Cos}\,(A, B) = \frac{A \,.\, B}{|A|\,|B|}$$

Similarity b/w Vectors

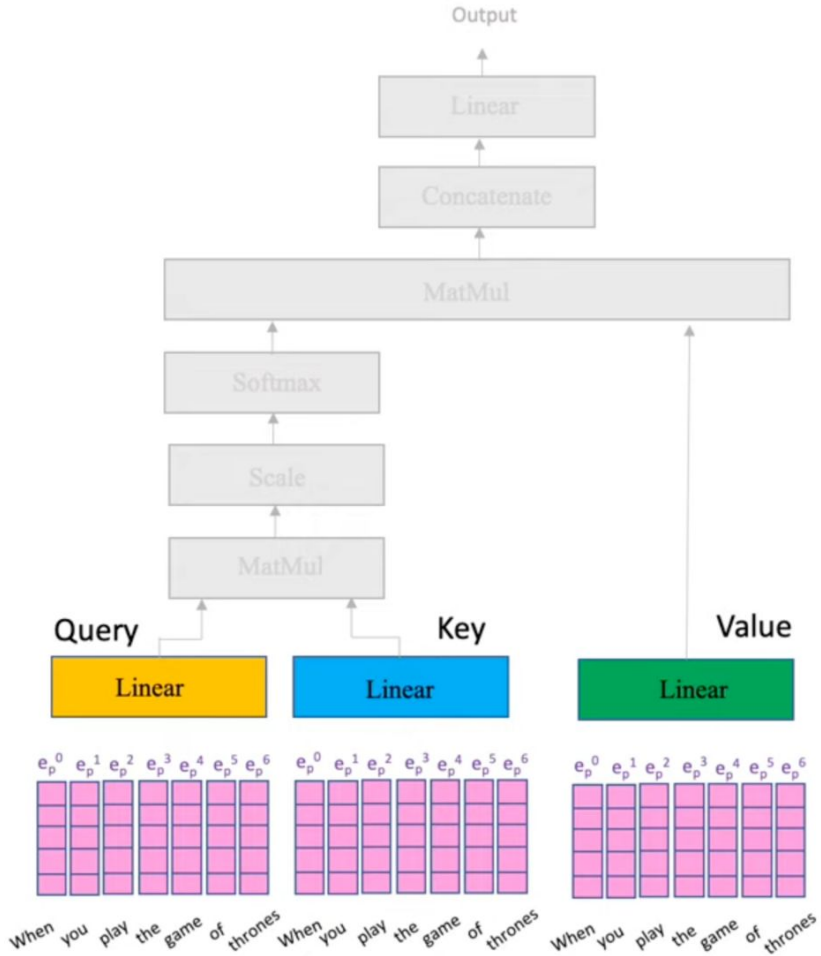
$$\text{similarity}\,(A, B) = \frac{A \,.\, B}{scaling}$$

Similarity b/w Matrices

$$\text{similarity } (A, B) = \frac{A \,.\, B^{T}}{scaling}$$

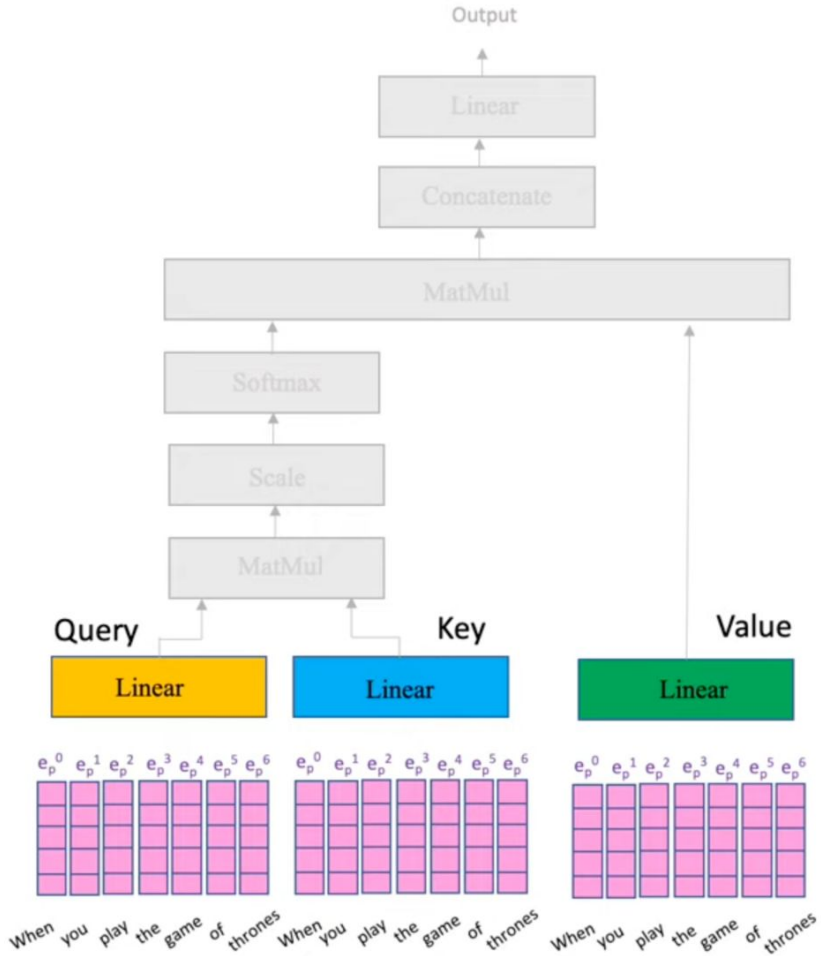
Similarity b/w Query and Key

$$\text{similarity}(Q, K) = \frac{Q \cdot K^T}{scaling}$$



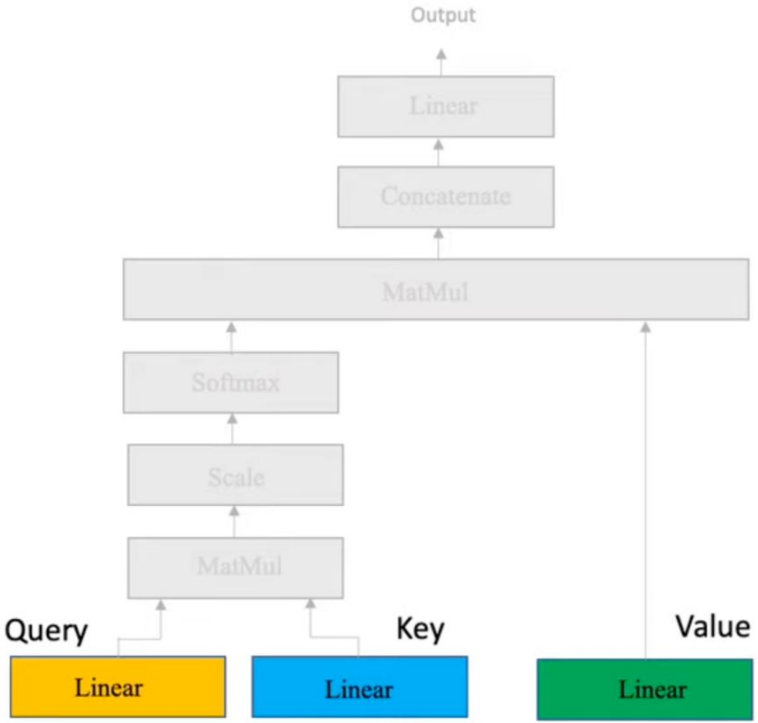
Multi-Head Attention

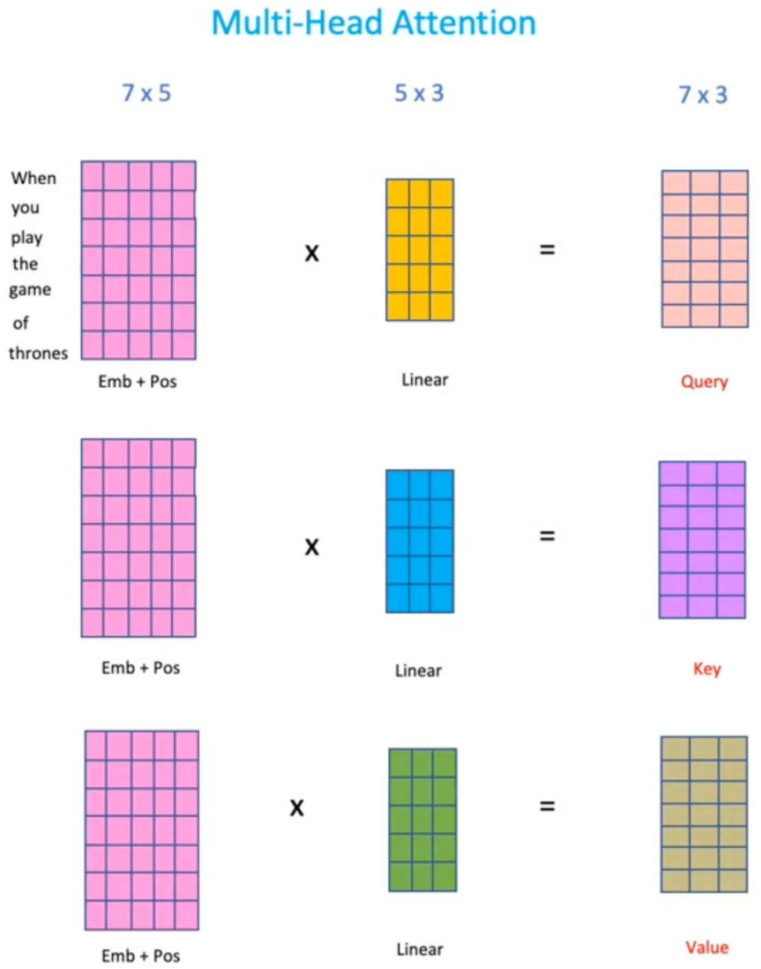
Self-Attention

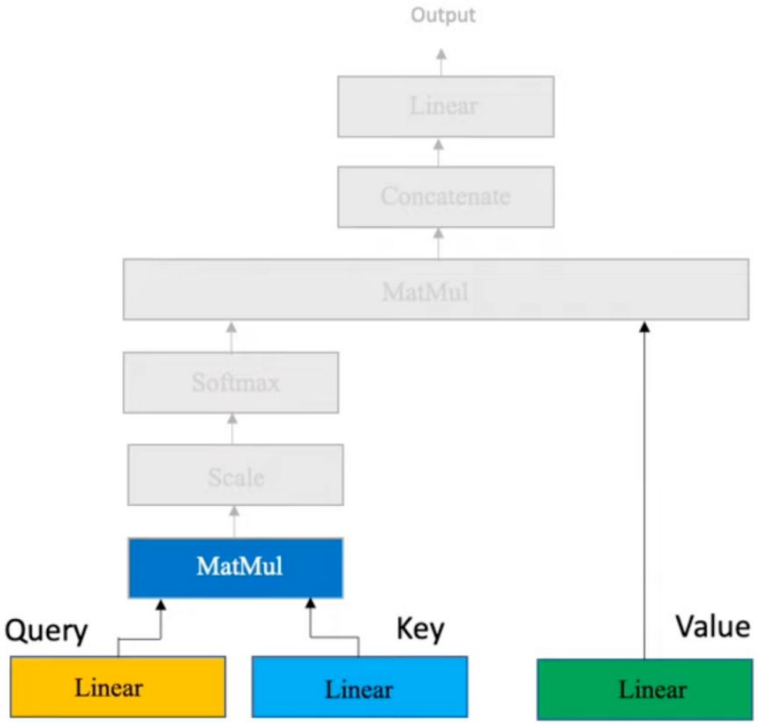


Multi-Head Attention

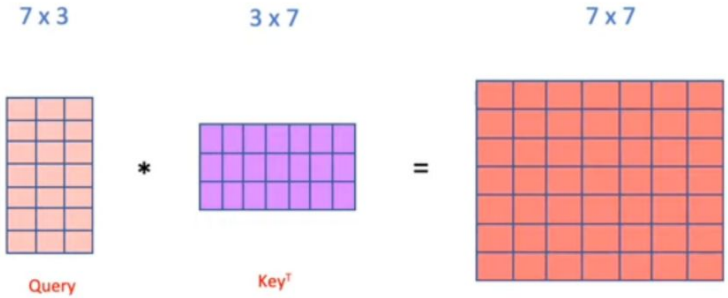
Self-Attention

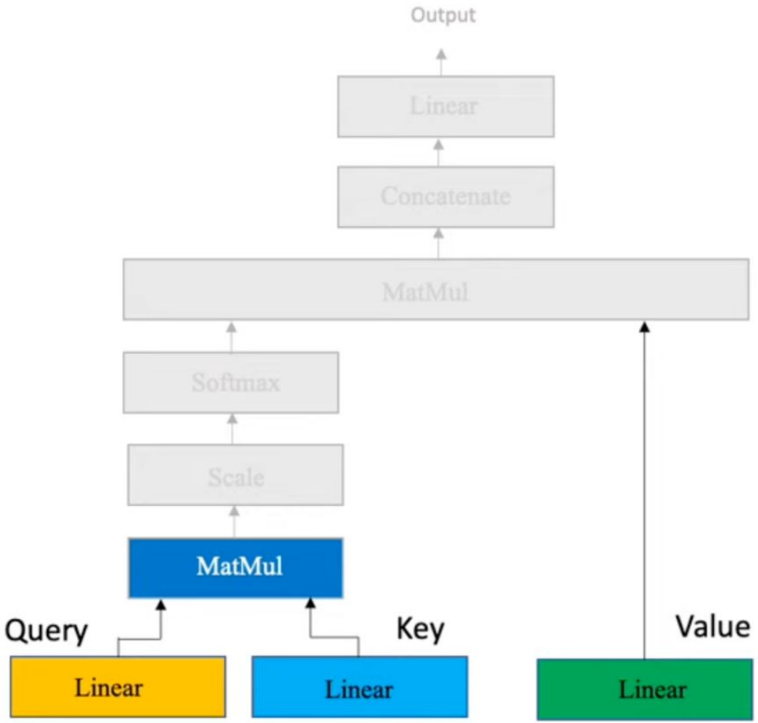

Output
Linear
Concatenate
MatMul
Softmax
Scale
MatMul
Query
Key
Value
Linear
Linear
Linear


Multi-Head Attention
7 x 5
5 x 3
7 x 3
When
you
play
the
game
of
thrones
Emb + Pos
x
Linear
=
Query
Emb + Pos
x
Linear
=
Key
Emb + Pos
x
Linear
=
Value



Multi-Head Attention



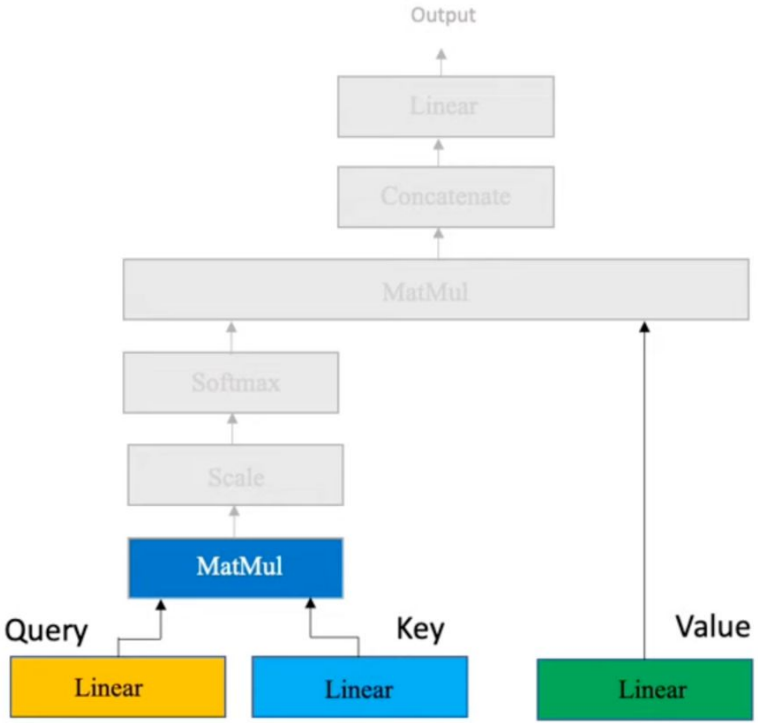


Multi-Head Attention

7 x 7

	When	you	play	the	game	of	thrones
When	89	48	41	36	35	40	19
you	67	91	11	92	17	99	11
play	91	10	11	11	12	41	98
the	11	96	28	12	98	11	00
game	76	11	91	24	12	12	12
of	11	29	77	78	22	93	13
thrones	11	87	12	12	13	98	19

Initially



Multi-Head Attention

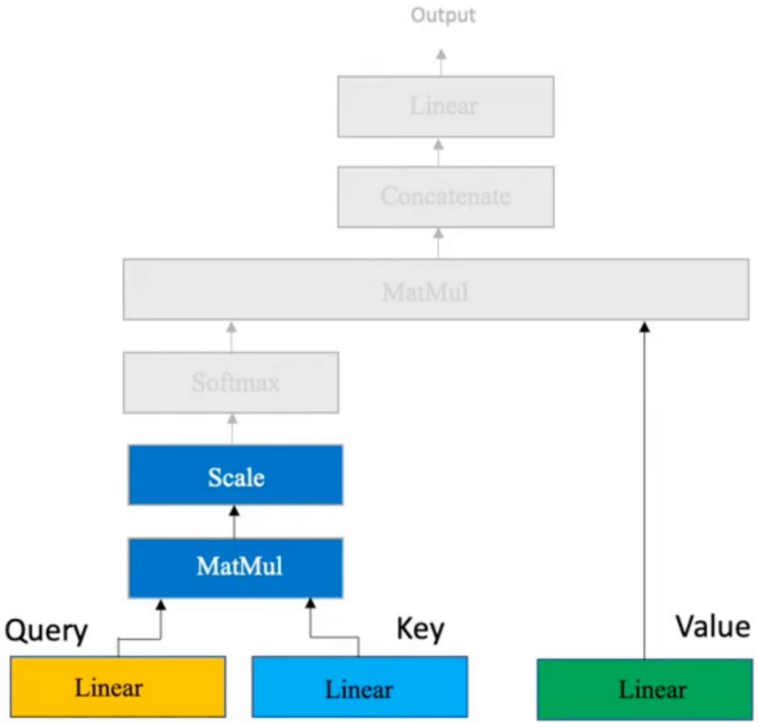
7 x 7

	When	you	play	the	game	of	thrones
When	89	20	41	10	55	78	59
you	90	98	81	22	87	15	32
play	29	81	95	10	90	30	92
the	10	22	67	12	88	40	89
game	22	70	90	56	98	44	80
of	10	15	30	40	44	44	59
thrones	59	72	92	90	13	59	99

After training

Captures relationships between tokens without thinking about their order

Unlike RNNs, transformers capture both local and global dependencies between tokens

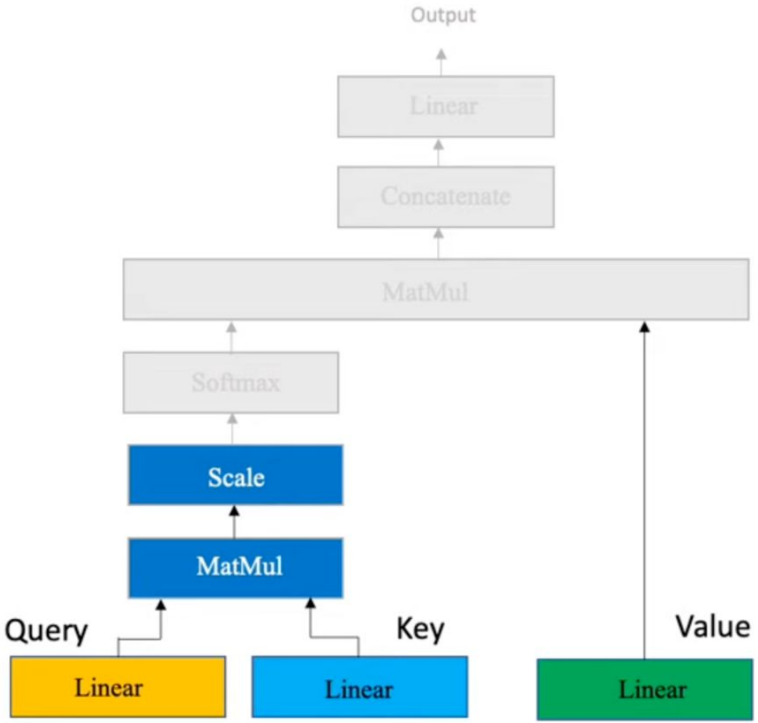


Multi-Head Attention

7×7

	When	you	play	the	game	of	thrones
When	89	20	41	10	55	10	59
you	20	90	81	22	70	15	72
play	41	81	95	10	90	30	92
the	10	22	10	92	88	40	89
game	55	70	90	88	98	44	87
of	10	15	30	40	44	85	59
thrones	59	72	92	90	95	59	99

$$\frac{}{\sqrt{d_k}}$$



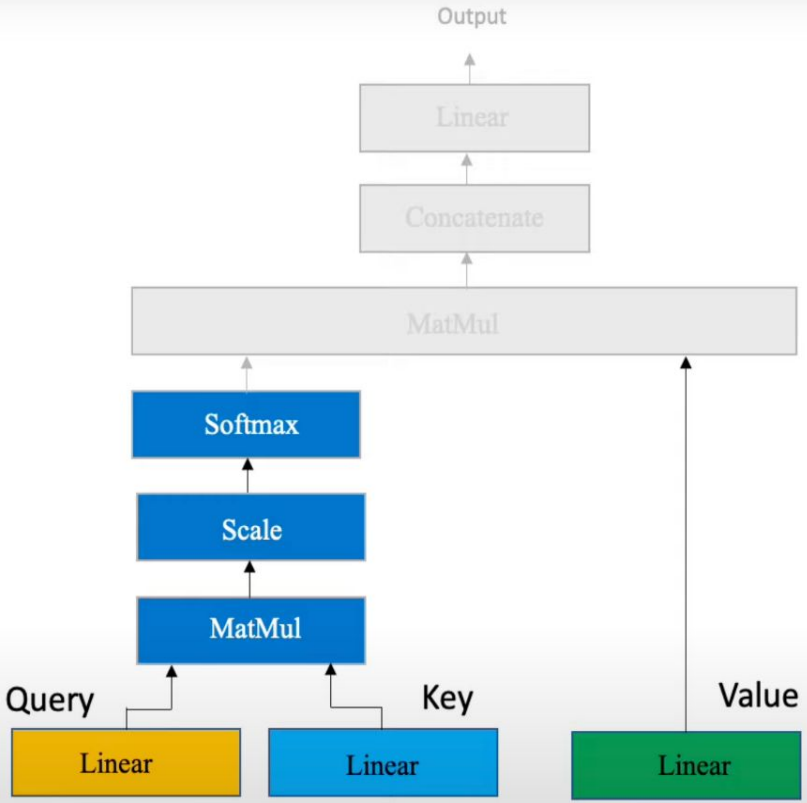
Multi-Head Attention

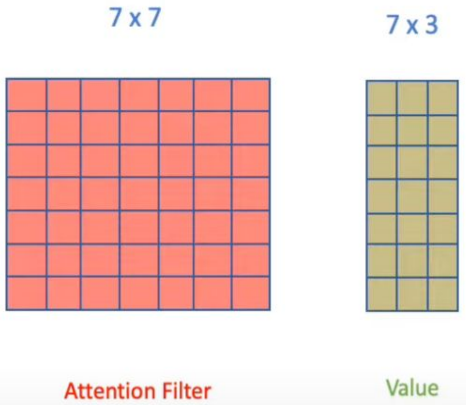
7 x 7

	When	you	play	the	game	of	thrones
When	89	20	41	10	55	10	59
you	20	90	81	22	70	15	72
play	41	81	95	10	90	30	92
the	10	22	10	92	88	40	89
game	55	70	90	88	98	44	87
of	10	15	30	40	44	85	59
thrones	59	72	92	90	95	59	99

$$\sqrt{7}$$

Multi-Head Attention





$$\text{softmax}\left(\frac{Q.K^T}{\sqrt{d_k}}\right) \quad V$$

Treat it as a feature extraction step

Intuition

Attention Filter

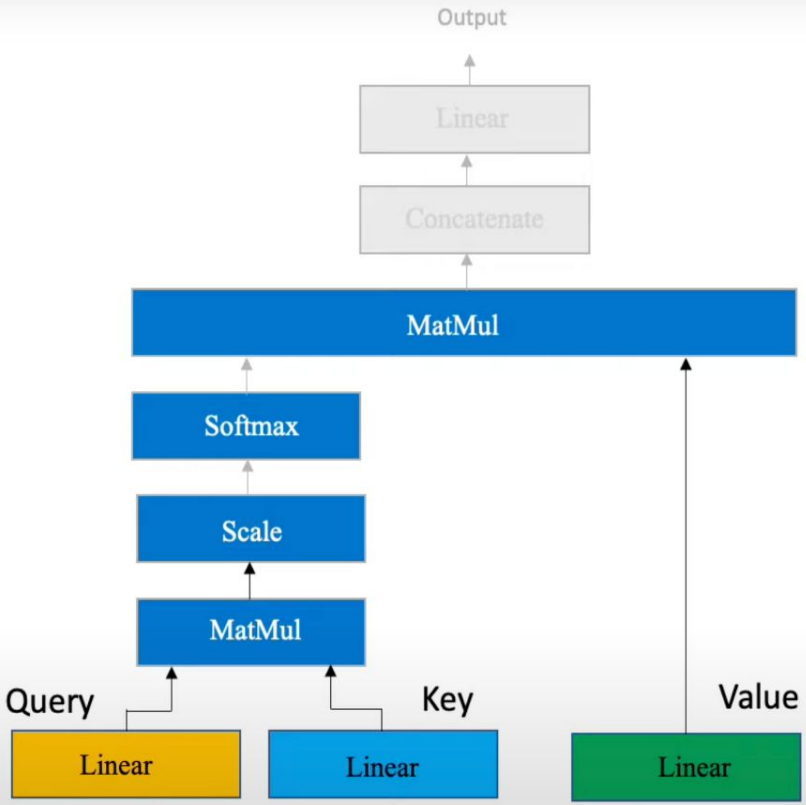
*

Original Image

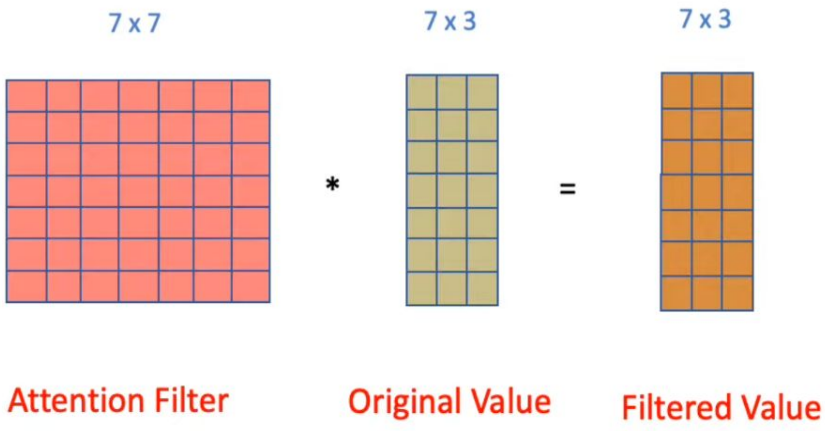
=

Filtered Image

Treat it as a feature extraction step



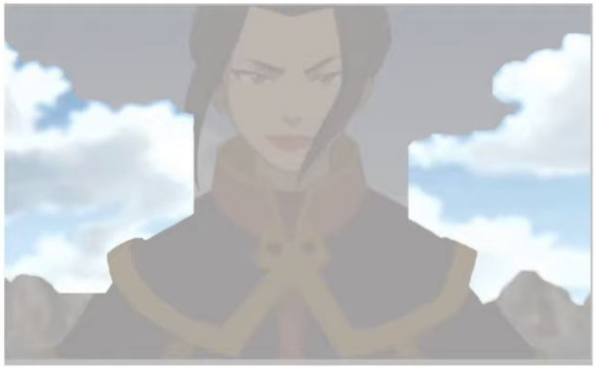
Multi-Head Attention



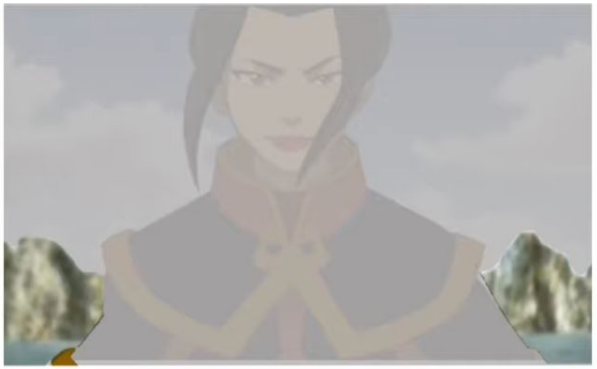
Intuition for Multi-head Attention

Attention Filter 1

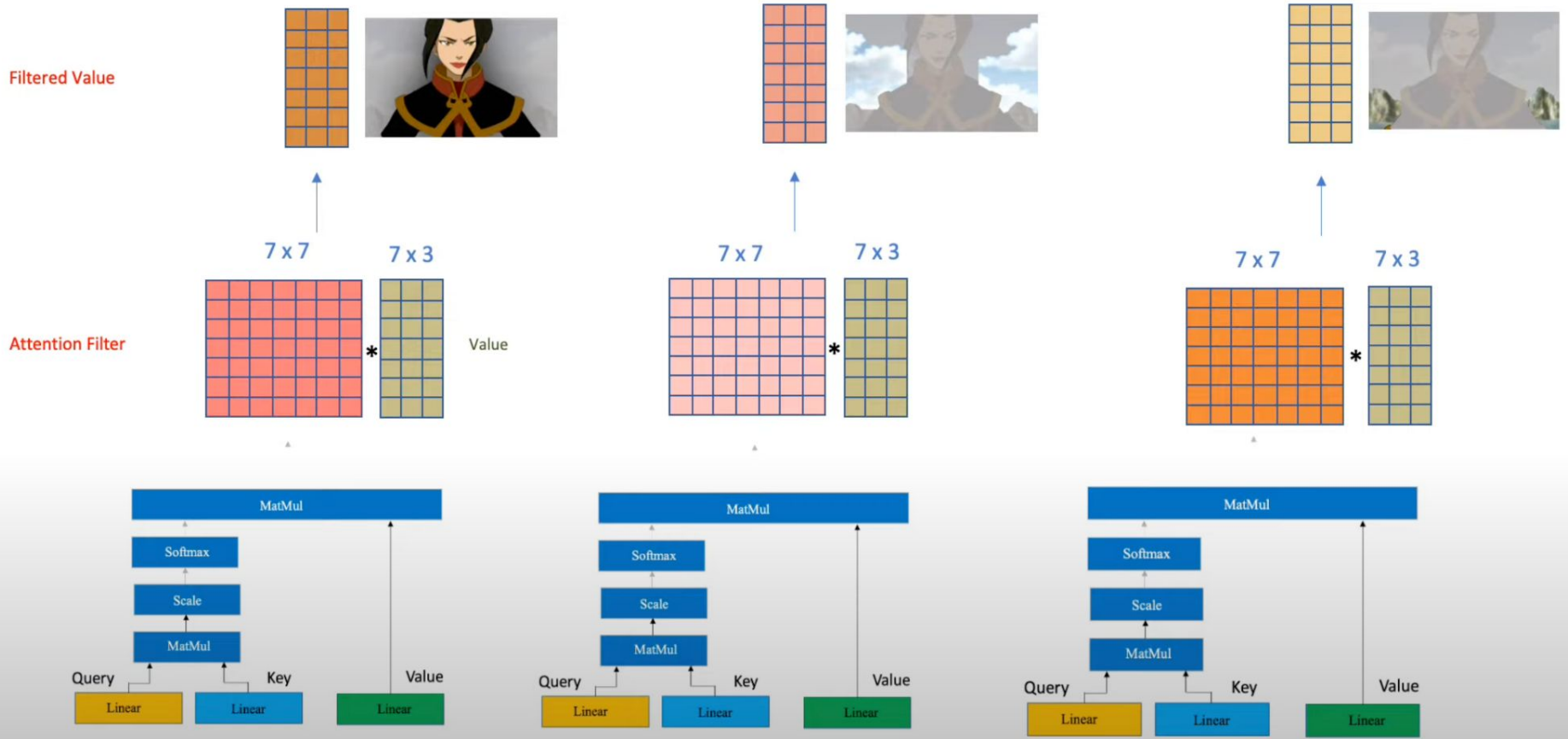
Attention Filter 2



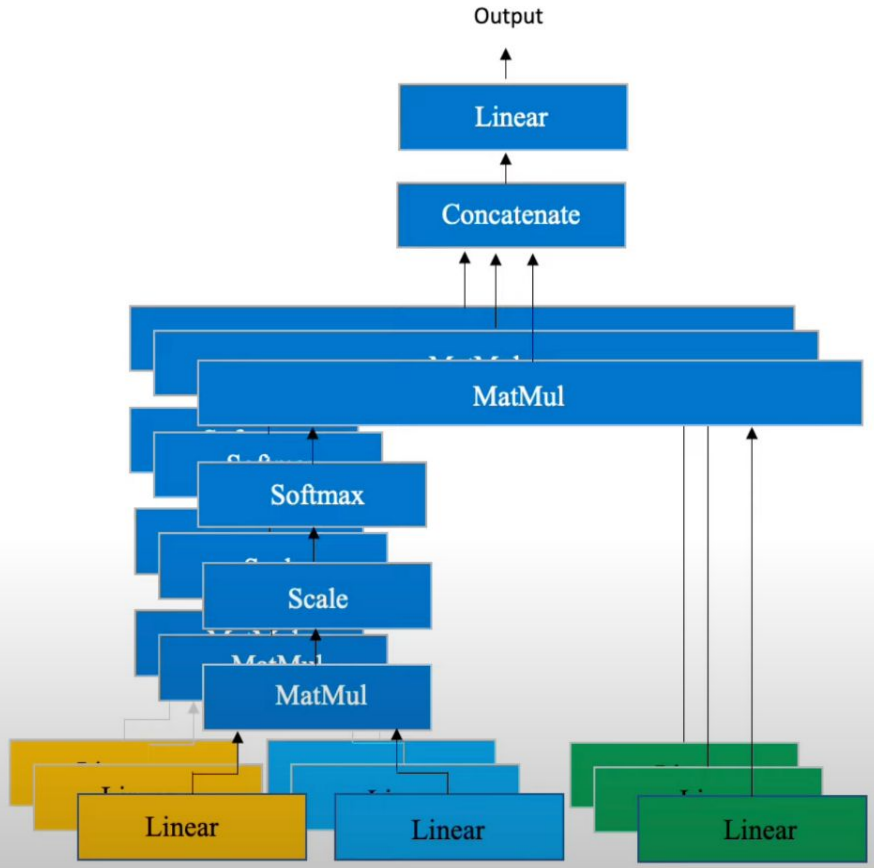
Attention Filter 3

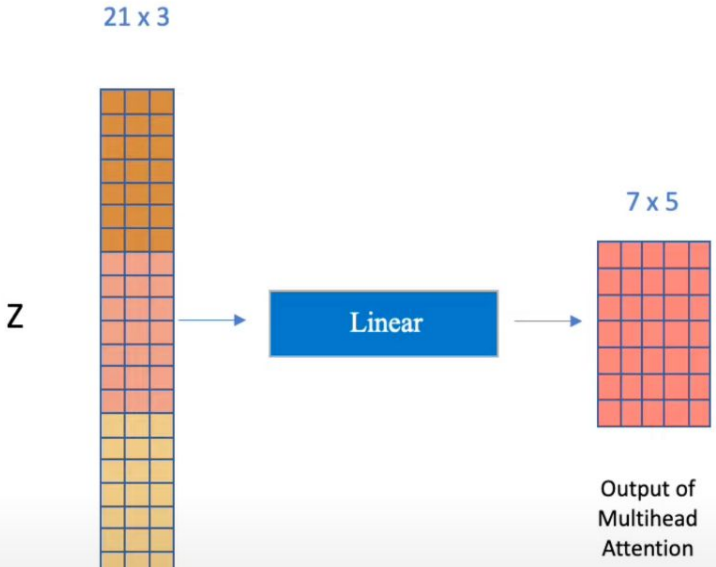


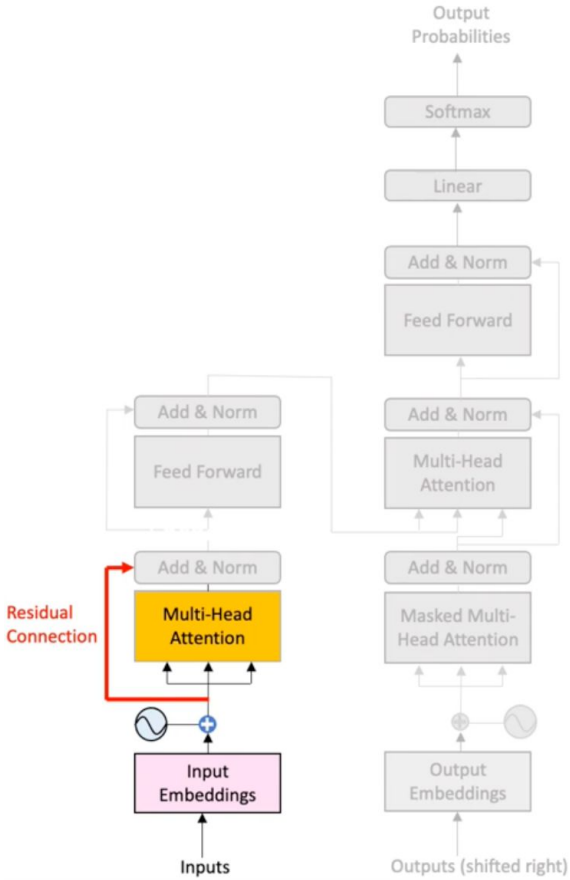
Multi-Head Attention



Multi-Head Attention

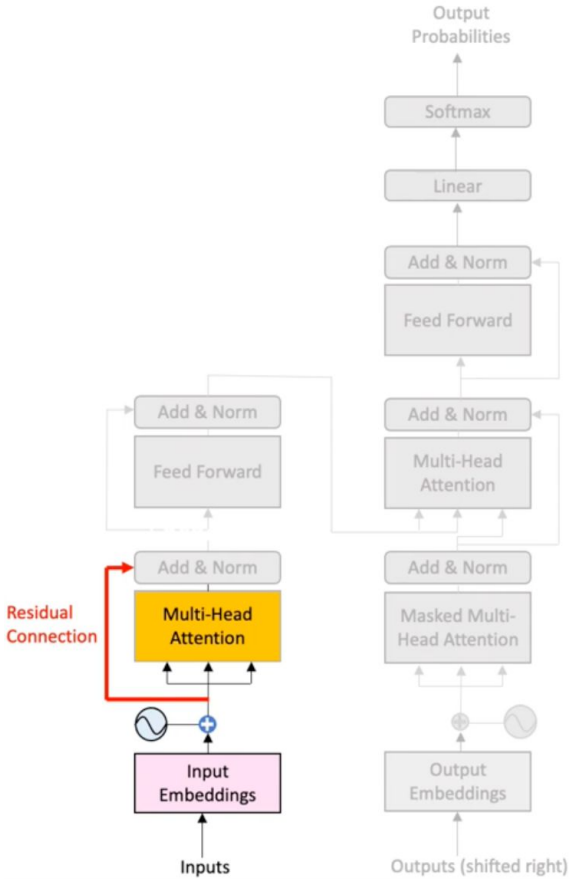






Why do we need residual connections? What's the purpose?

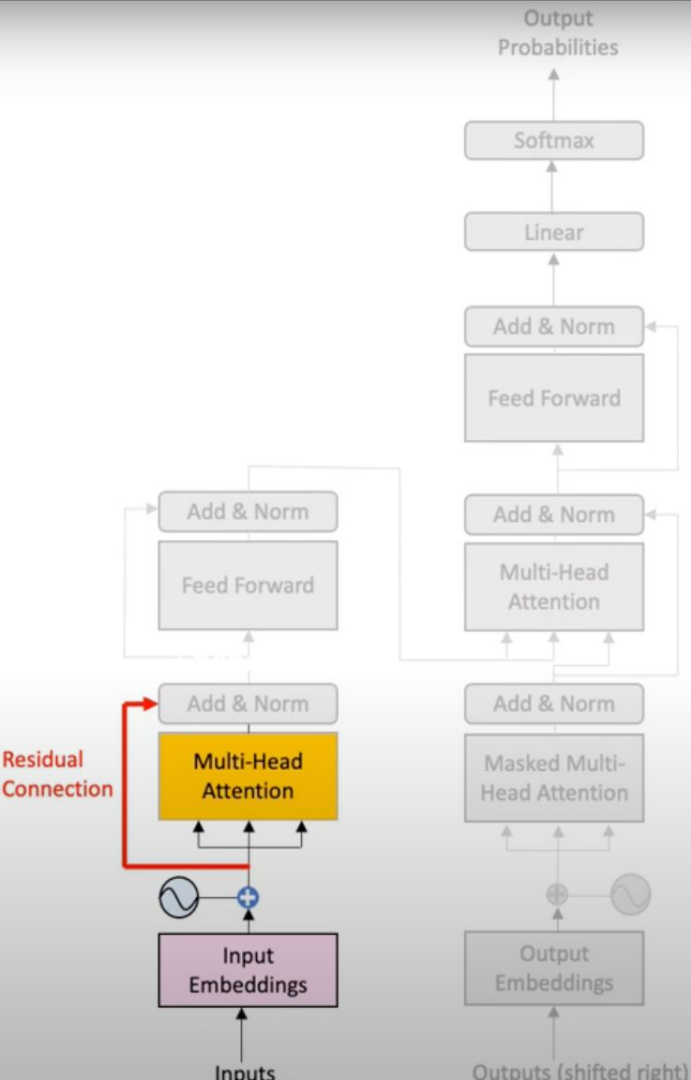
Let's discuss!



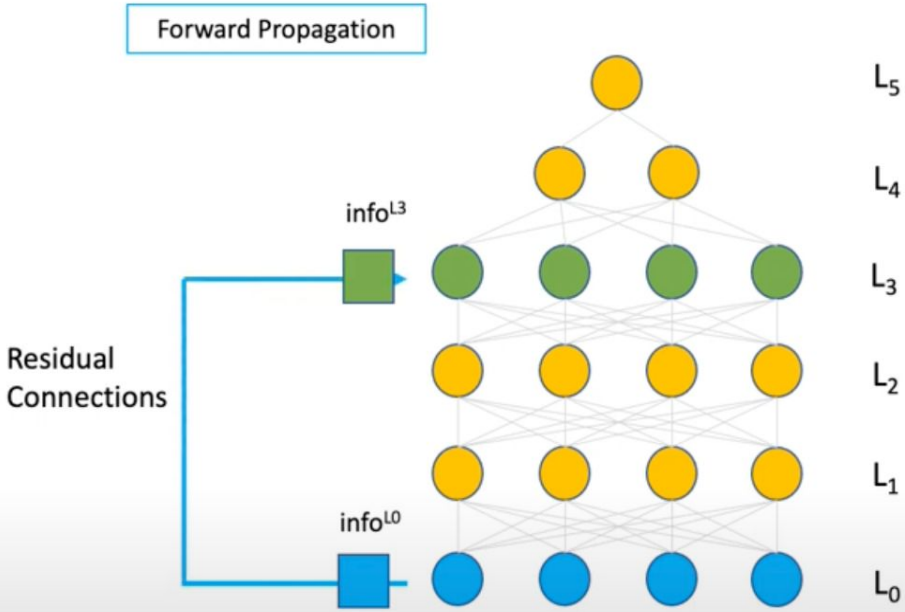
Residual Connection

i) Knowledge Preservation

ii) Vanishing Gradient Problem

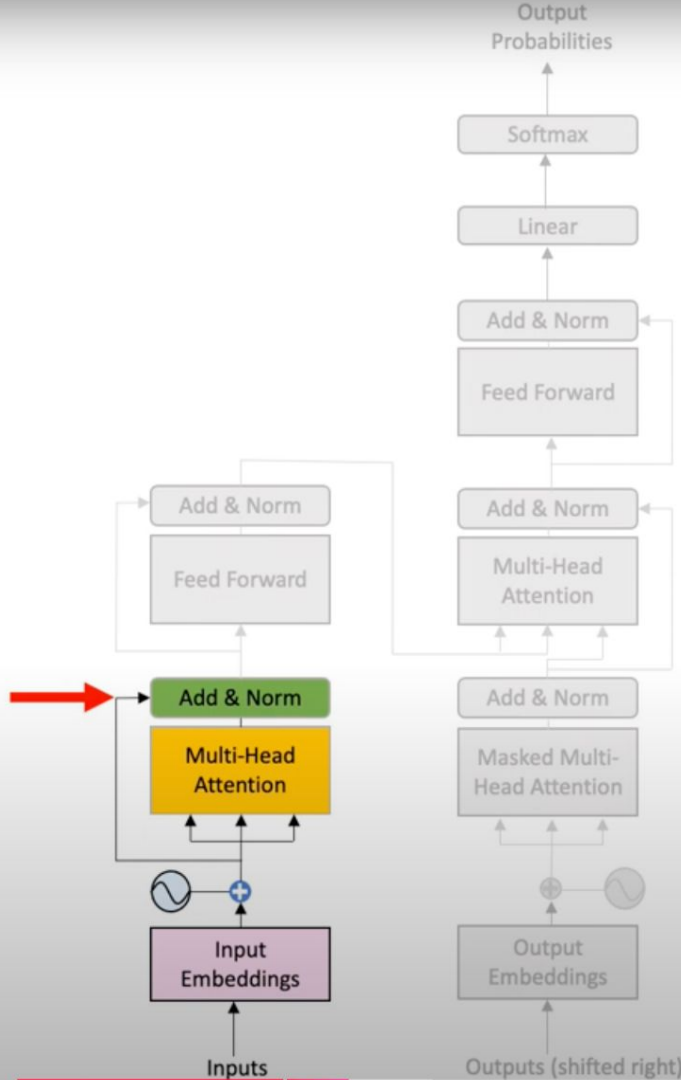


Residual Connection

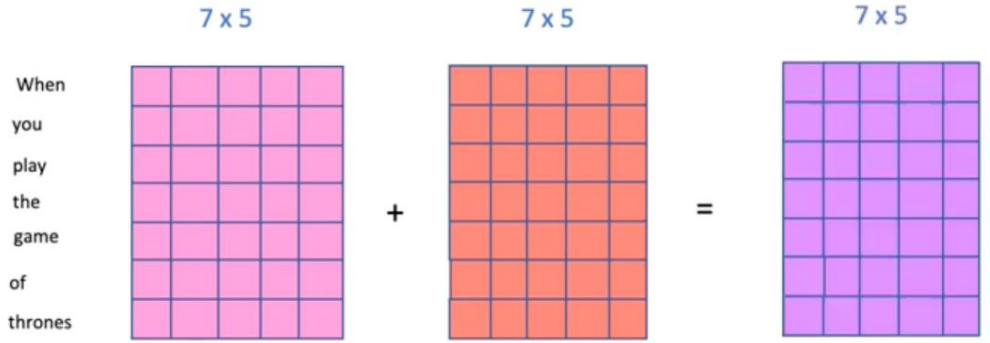


Preservation of earlier information


DANGER SIGN
32.37 M
261

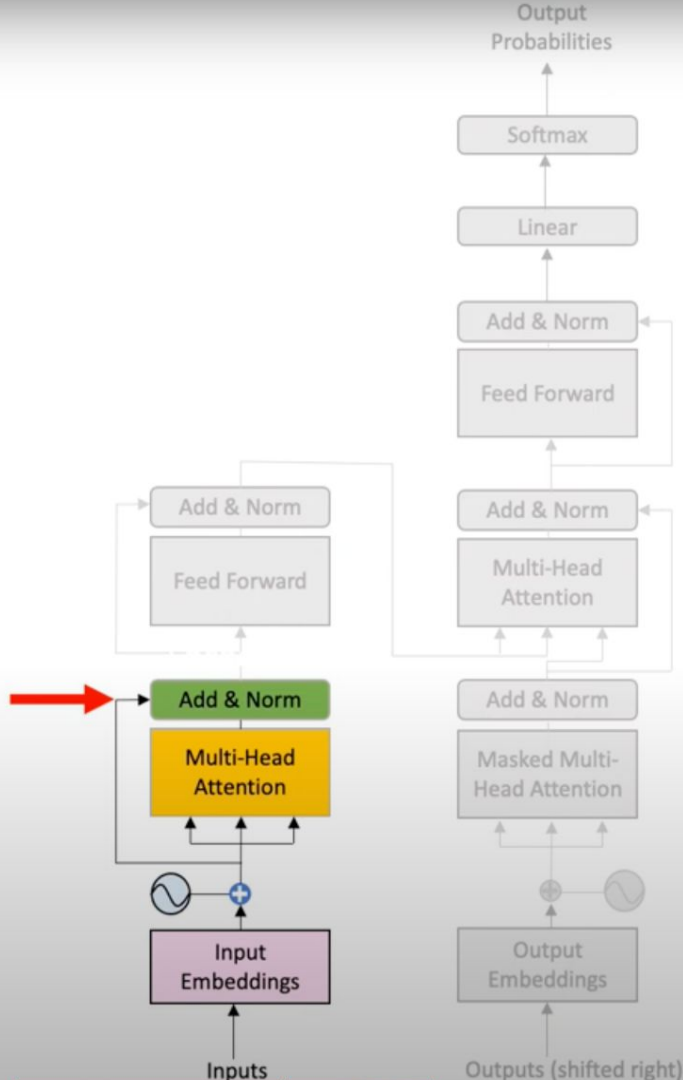


Add and Norm



emb + pos

output of multihead attention

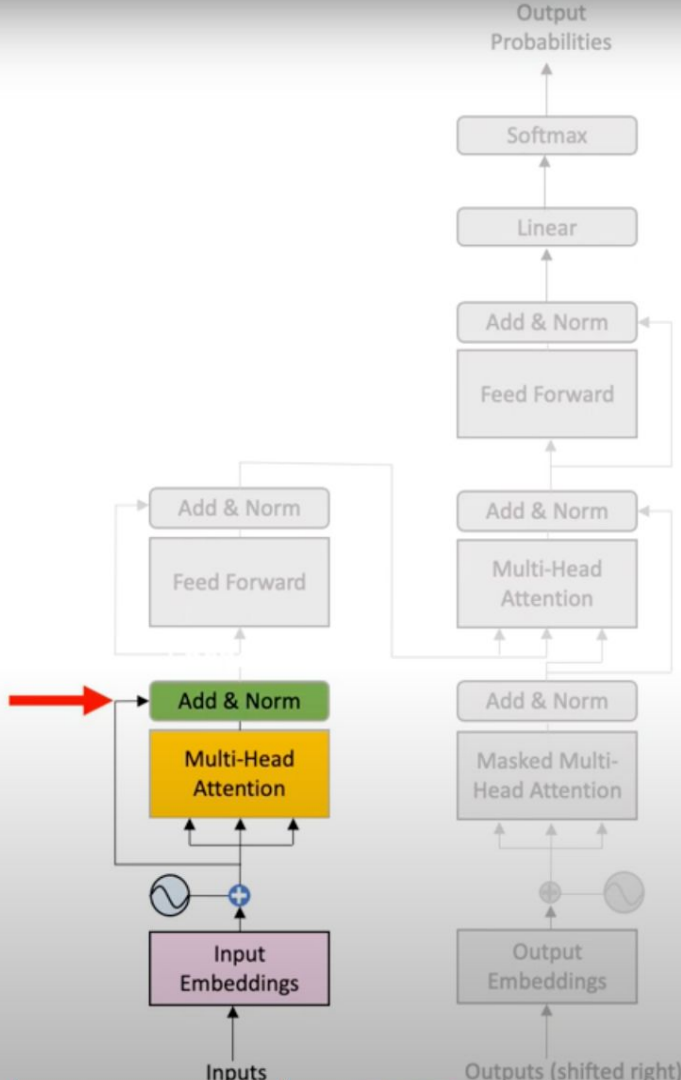


Layer Normalization

7 x 3

						mean (μ)	std (σ)
x_0 = When	0.98	1.28	0.41	0.27	0.41	0.67	0.44
x_1 = you	0.52	0.01	2.06	0.27	0.33	0.64	0.82
x_2 = play	2.22	0.27	0.10	0.41	2.06	1.01	1.04
x_3 = the	0.99	1.00	0.11	0.27	0.33	0.54	0.42
x_4 = game	0.52	0.01	0.33	2.06	0.52	0.69	0.79
x_5 = of	0.10	2.06	0.73	0.27	0.41	0.71	0.79
x_6 = thrones	0.33	0.01	0.13	0.27	1.28	0.40	0.51
	f_0	f_1	f_2	f_3	f_4		

$$x_0 = \frac{0.98 - 0.67}{\sqrt{0.44^2 + 0.0001}}$$

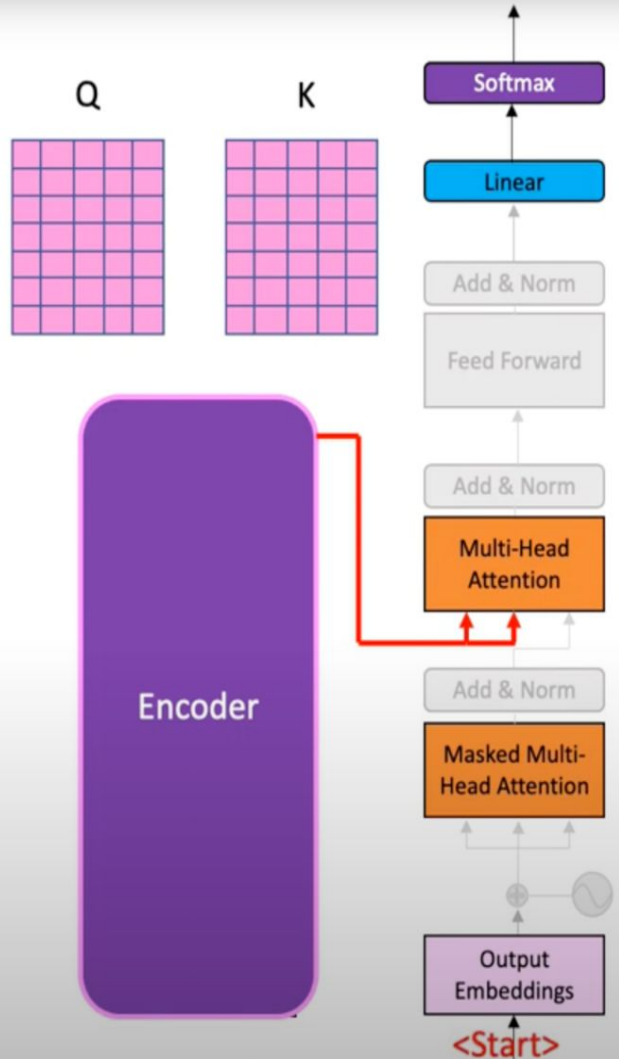


Layer Normalization

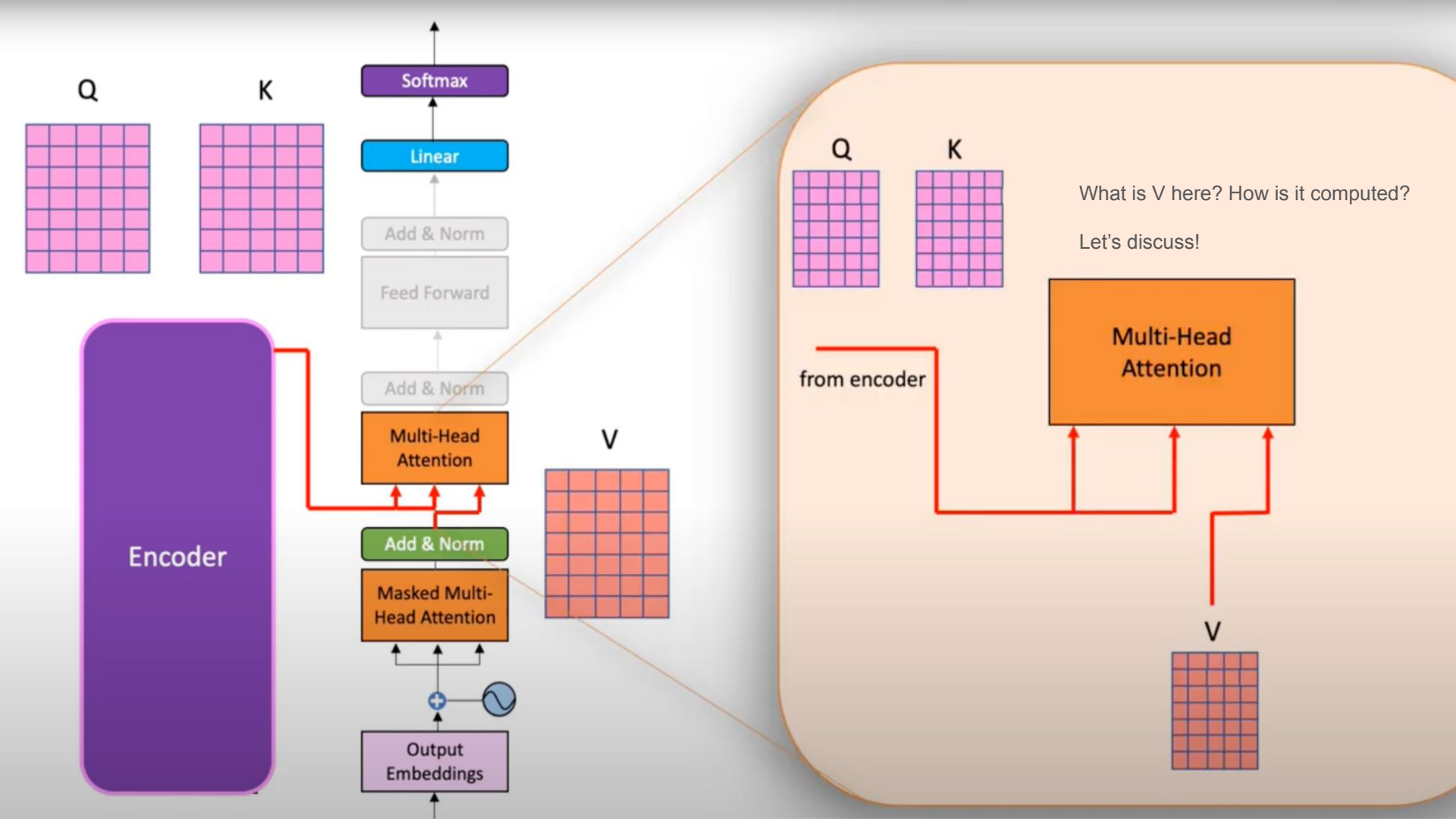
7 x 3

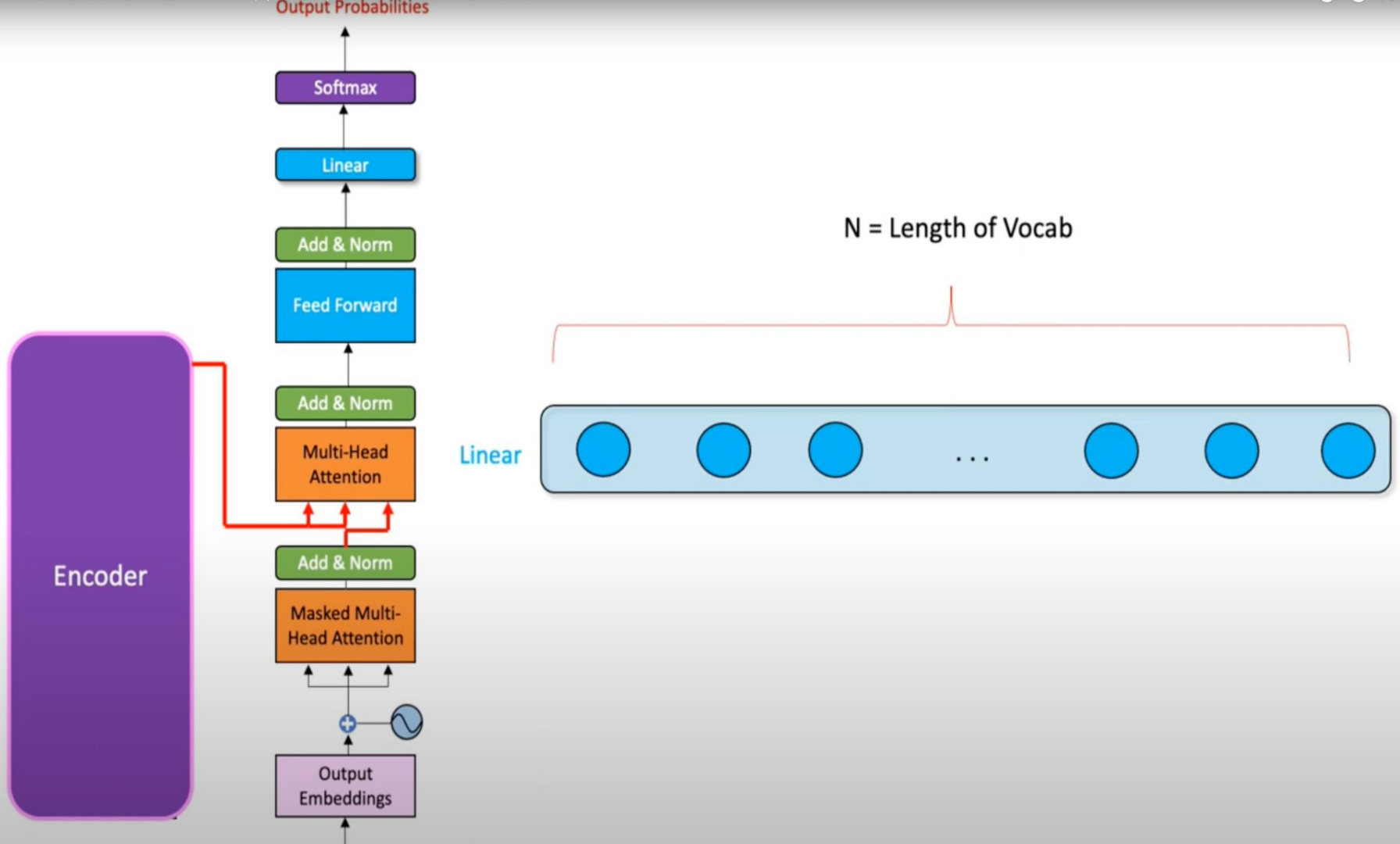
						mean (μ)	std (σ)
x_0 = When	0.71	1.40	-0.59	-0.92	-0.59	0.67	0.44
x_1 = you	-0.14	-0.77	1.74	-0.45	-0.38	0.64	0.82
x_2 = play	1.16	-0.72	-0.88	-0.58	1.01	1.01	1.04
x_3 = the	1.06	1.09	-1.02	-0.64	-0.50	0.54	0.42
x_4 = game	-0.21	-0.85	-0.45	1.73	-0.21	0.69	0.79
x_5 = of	-0.78	1.71	0.02	-0.56	-0.39	0.71	0.79
x_6 = thrones	-0.15	-0.78	-0.54	-0.27	1.73	0.40	0.51
	f_0	f_1	f_2	f_3	f_4		

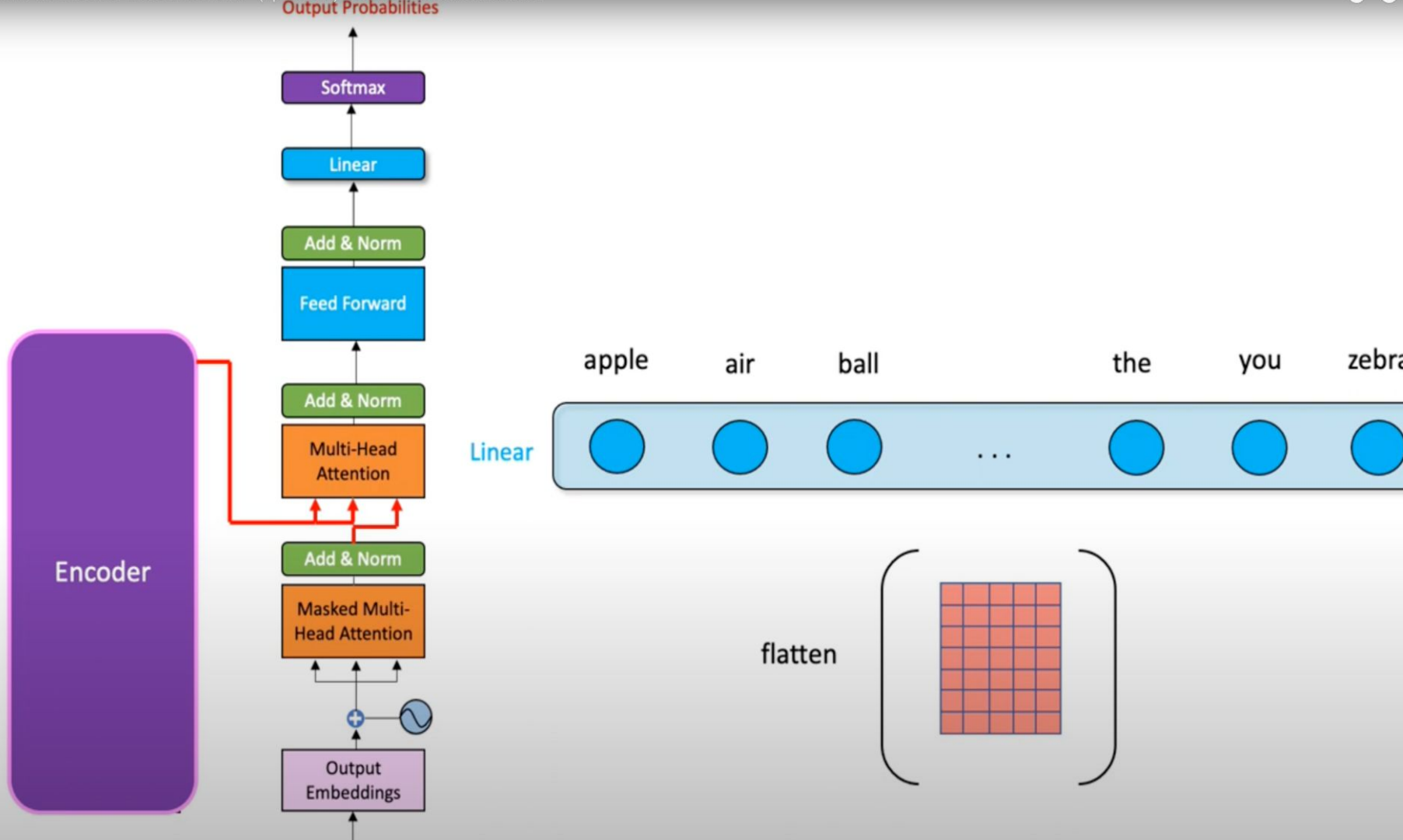
Stable values; mitigates risk of vanishing and exploding values

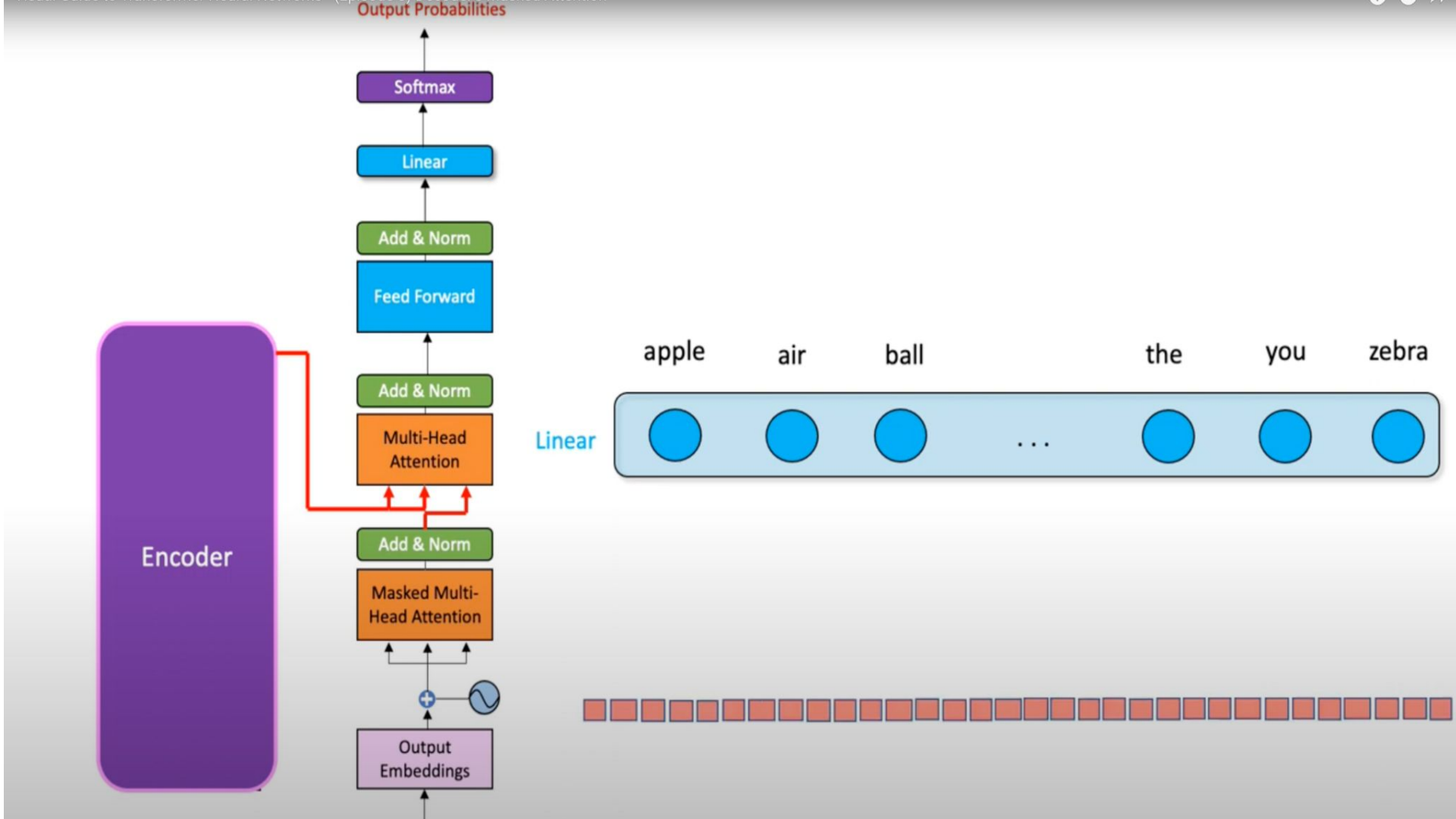


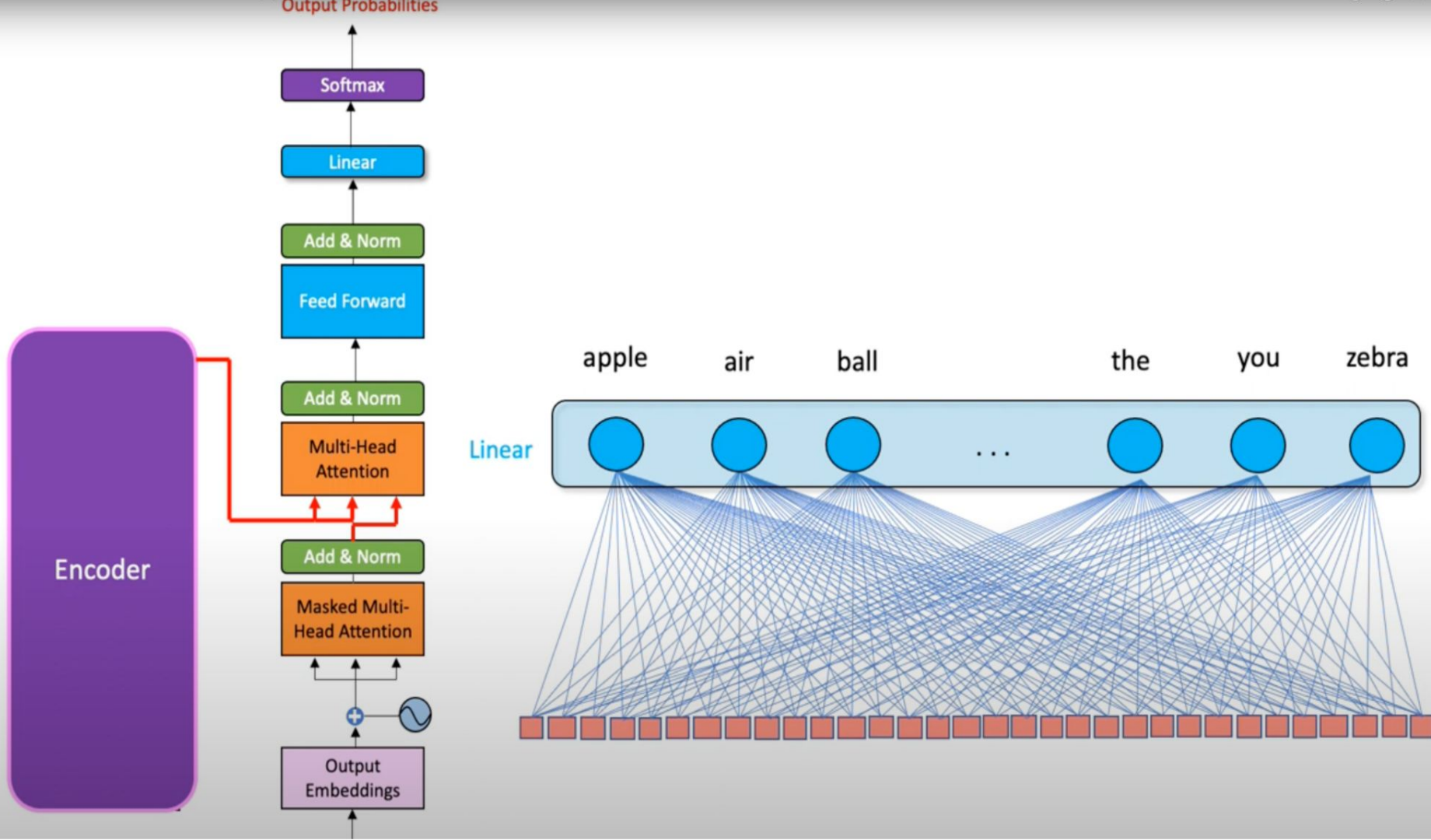
Encoder has learnt latent representation of the input

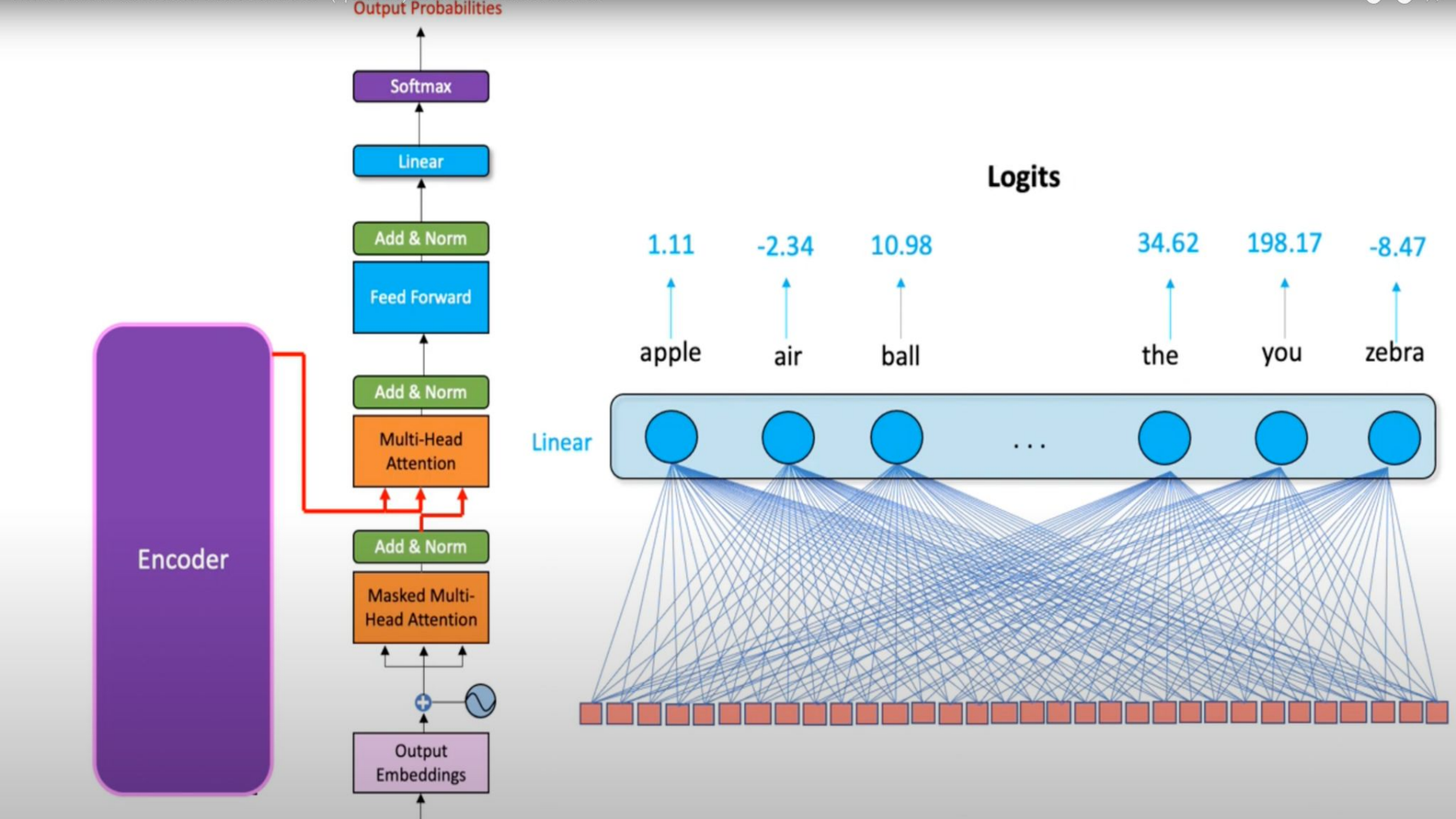

Q
K
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
V
Encoder
Add & Norm
Masked Multi-Head Attention
Output Embeddings
Q
K
What is V here? How is it computed?
Let's discuss!
from encoder
Multi-Head Attention
V

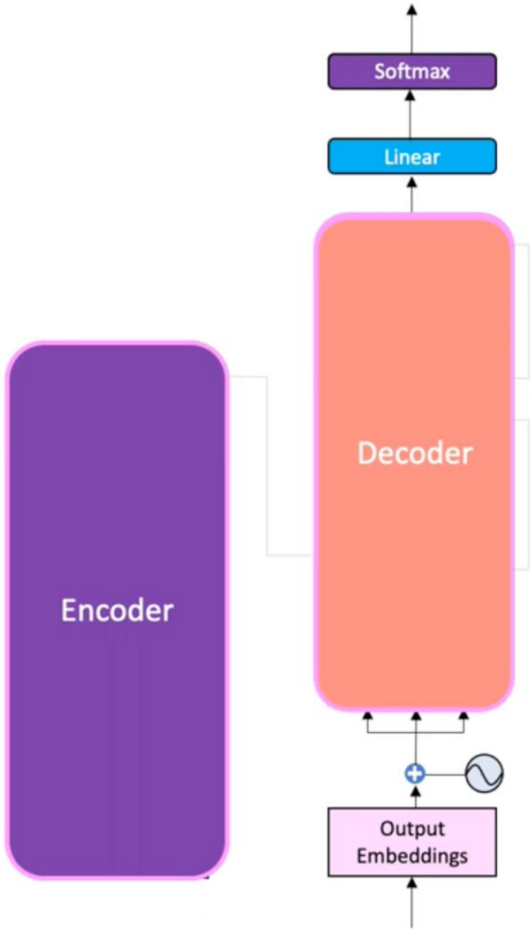

Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Output Embeddings
Encoder
N = Length of Vocab
Linear


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Output Embeddings
Encoder
Linear
apple
air
ball
the
you
zebra
flatten


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Output Embeddings
Encoder
Linear
apple
air
ball
the
you
zebra

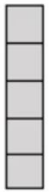

Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Output Embeddings
Encoder
Linear
apple
air
ball
the
you
zebra


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Output Embeddings
Encoder
Logits
1.11
-2.34
10.98
34.62
198.17
-8.47
apple
air
ball
the
you
zebra
Linear



<start> You

$e_p^{(0)}$

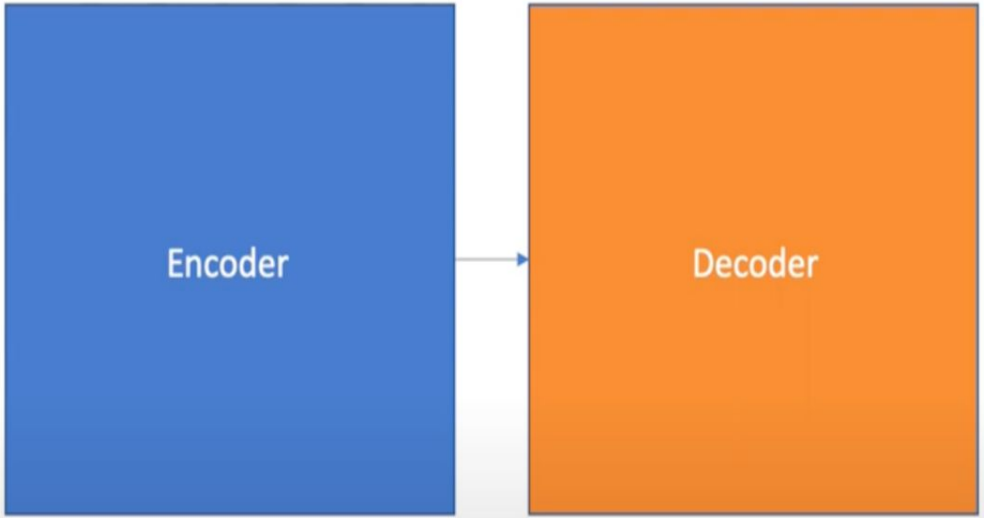



Masking

Transformer - Training

Dialogue PART 1

Fool
!
No
man
can
kill
me



Dialogue PART 2

I
am
no
man
<end>

While predicting the current word, the model should not see future words

Mask future words

Quiz

I don’t spoon-feed you to reach to the answer

Masking

	<start>	I	am	no	man	<end>
<start>	33.6	7.6	15.5	3.8	20.8	22.3
I	7.6	34.0	30.6	8.3	26.5	27.2
am	15.5	30.6	35.9	3.8	34.0	34.8
no	3.8	8.3	3.8	34.8	33.3	33.6
man	20.8	26.5	34.0	33.3	37.0	35.9
<end>	3.8	5.7	11.3	15.1	16.6	37.4

Attention Filter

+

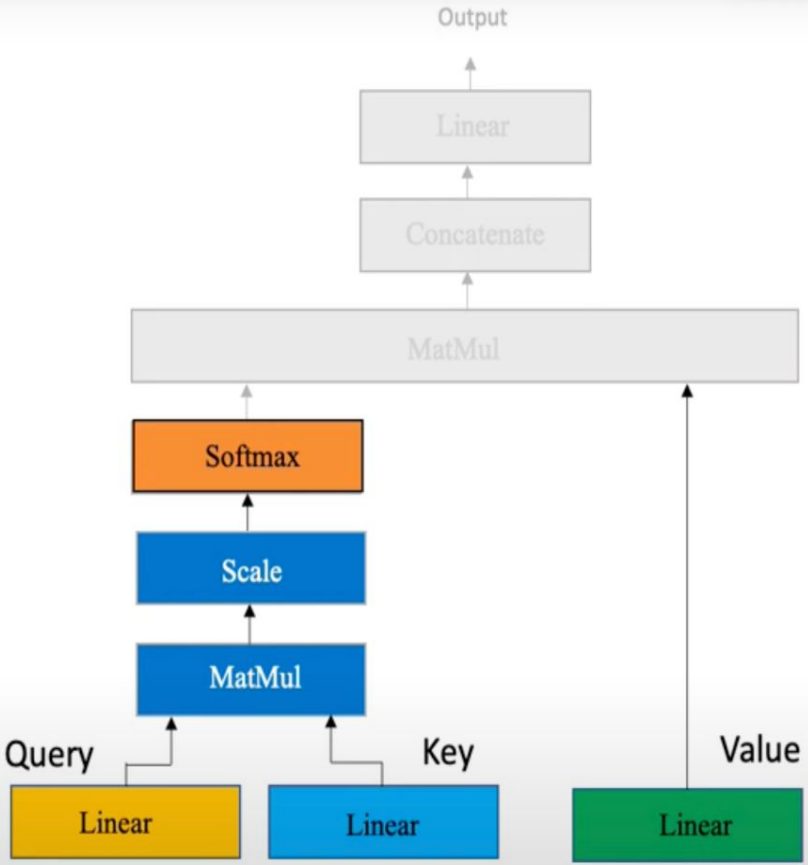
	<start>	I	am	no	man	<end>
<start>	0	-inf	-inf	-inf	-inf	-inf
I	0	0	-inf	-inf	-inf	-inf
am	0	0	0	-inf	-inf	-inf
no	0	0	0	0	-inf	-inf
man	0	0	0	0	0	-inf
<end>	0	0	0	0	0	0

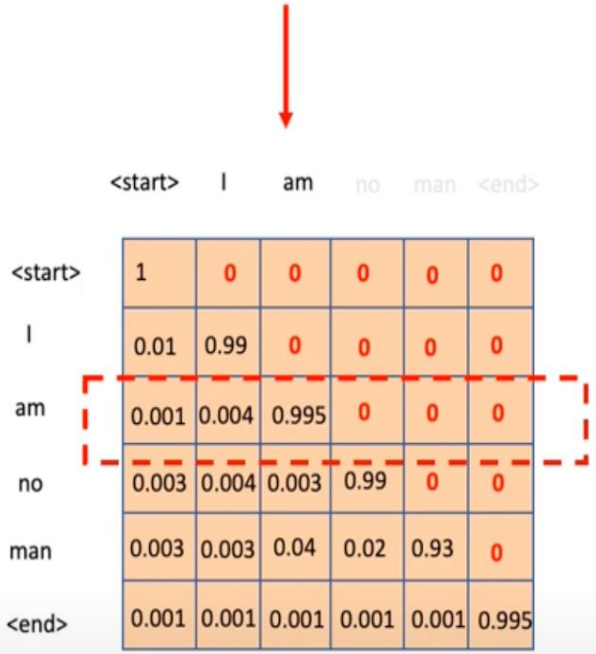
Mask Filter

=

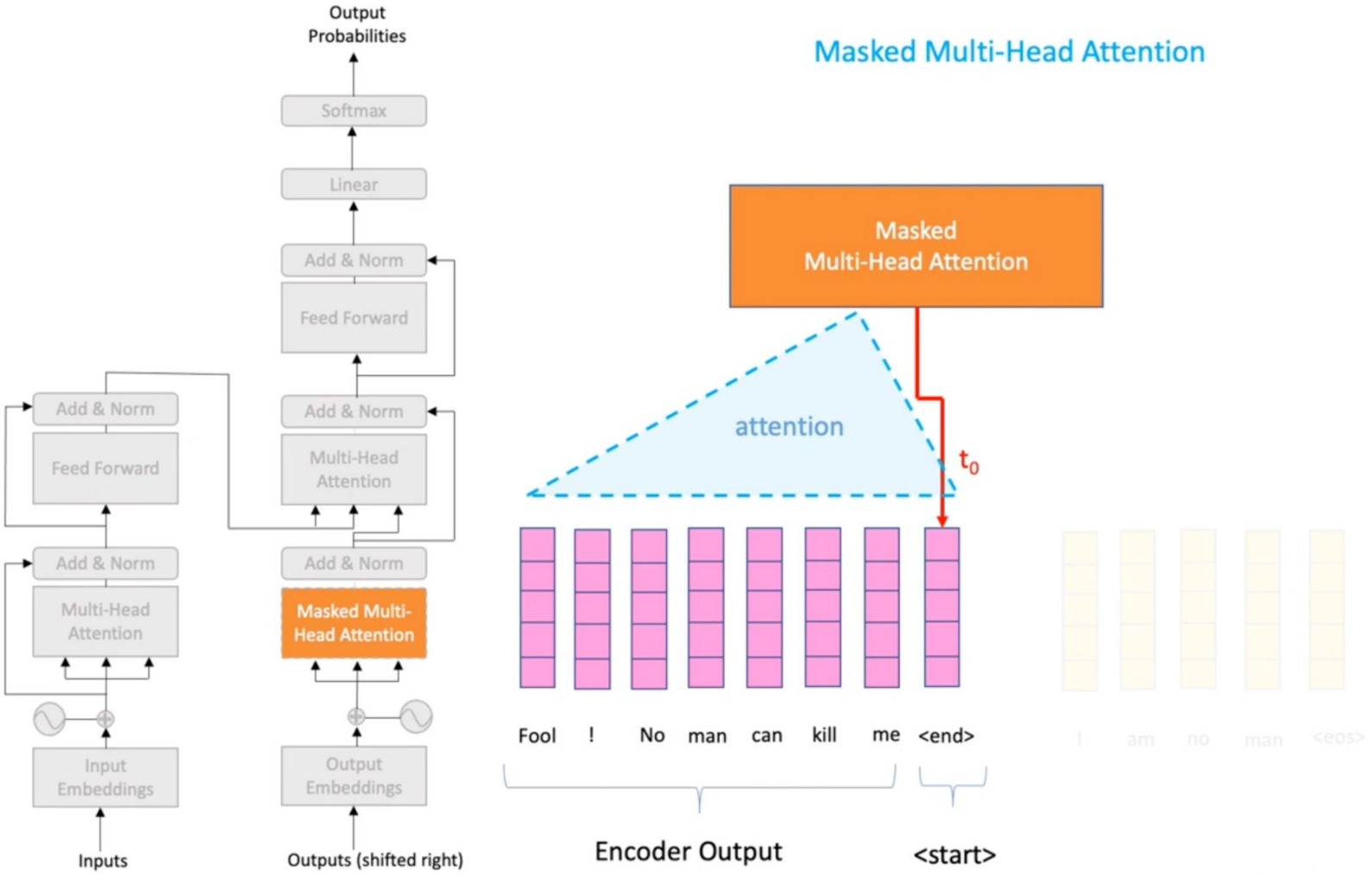
	<start>	I	am	no	man	<end>
<start>	33.6	-inf	-inf	-inf	-inf	-inf
I	7.6	34.0	-inf	-inf	-inf	-inf
am	15.5	30.6	35.9	-inf	-inf	-inf
no	3.8	8.3	3.8	34.8	-inf	-inf
man	20.8	26.5	34.0	33.3	37.0	-inf
<end>	3.8	5.7	11.3	15.1	16.6	37.4

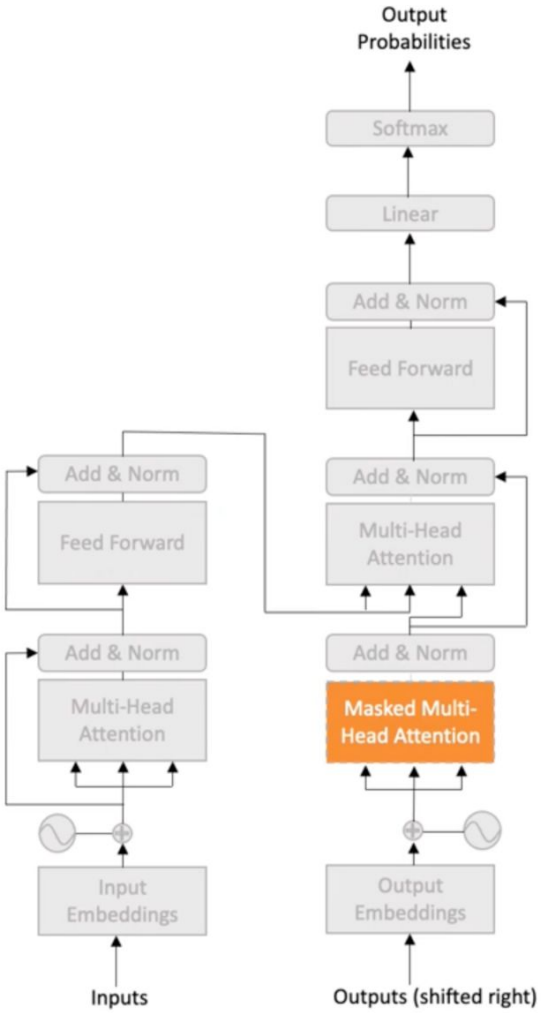
Masked-Attention Filter

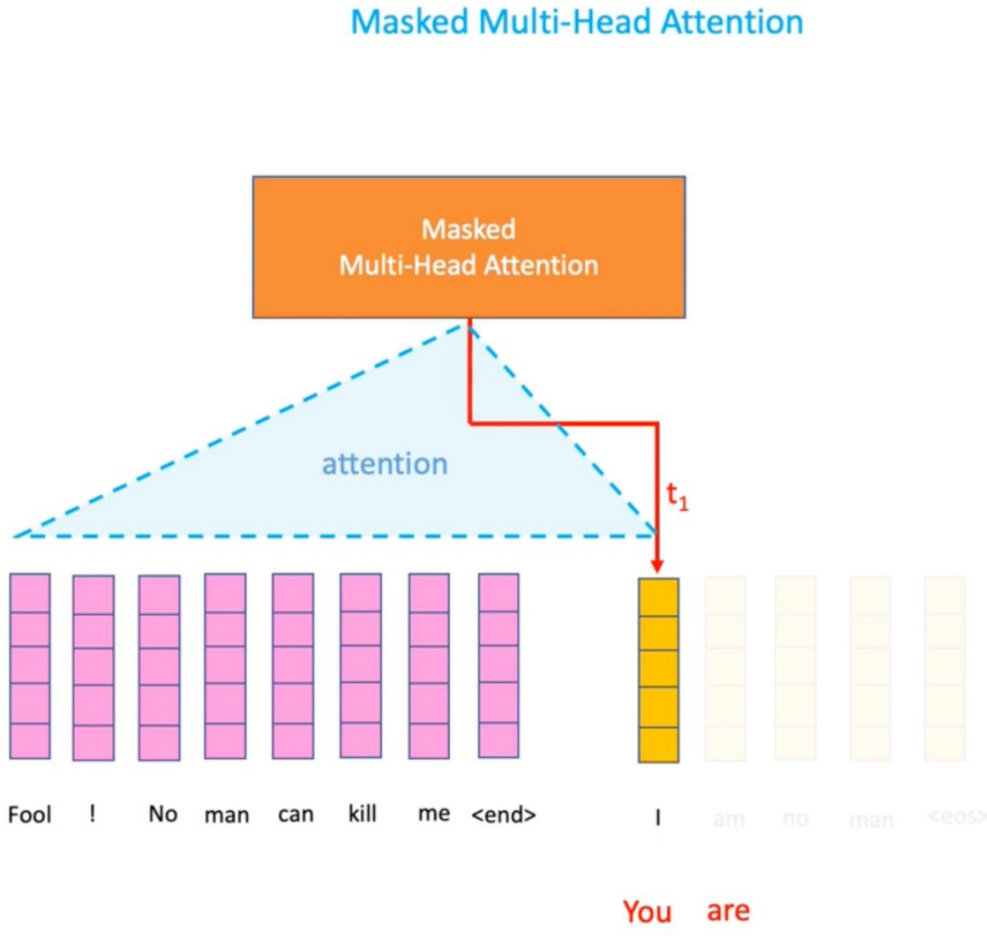


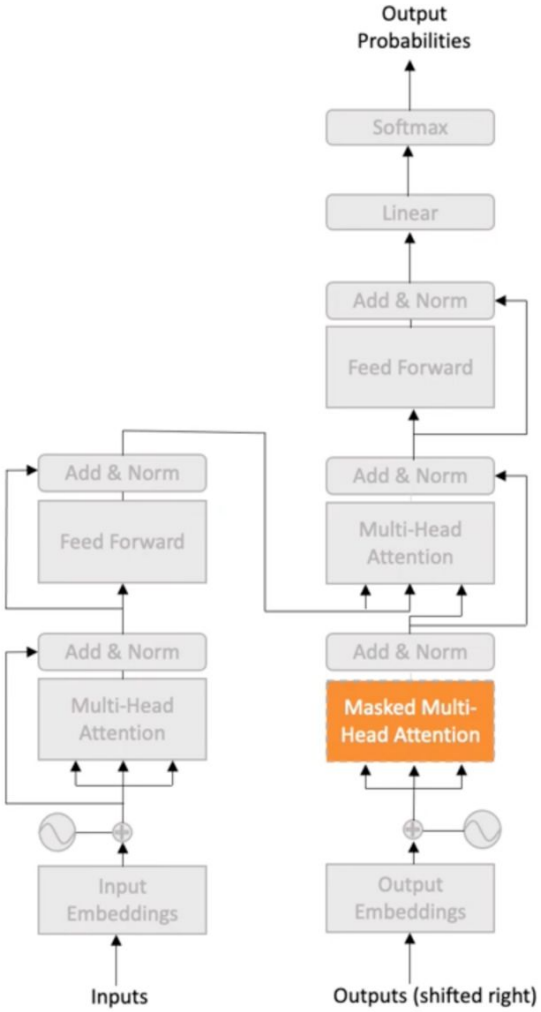


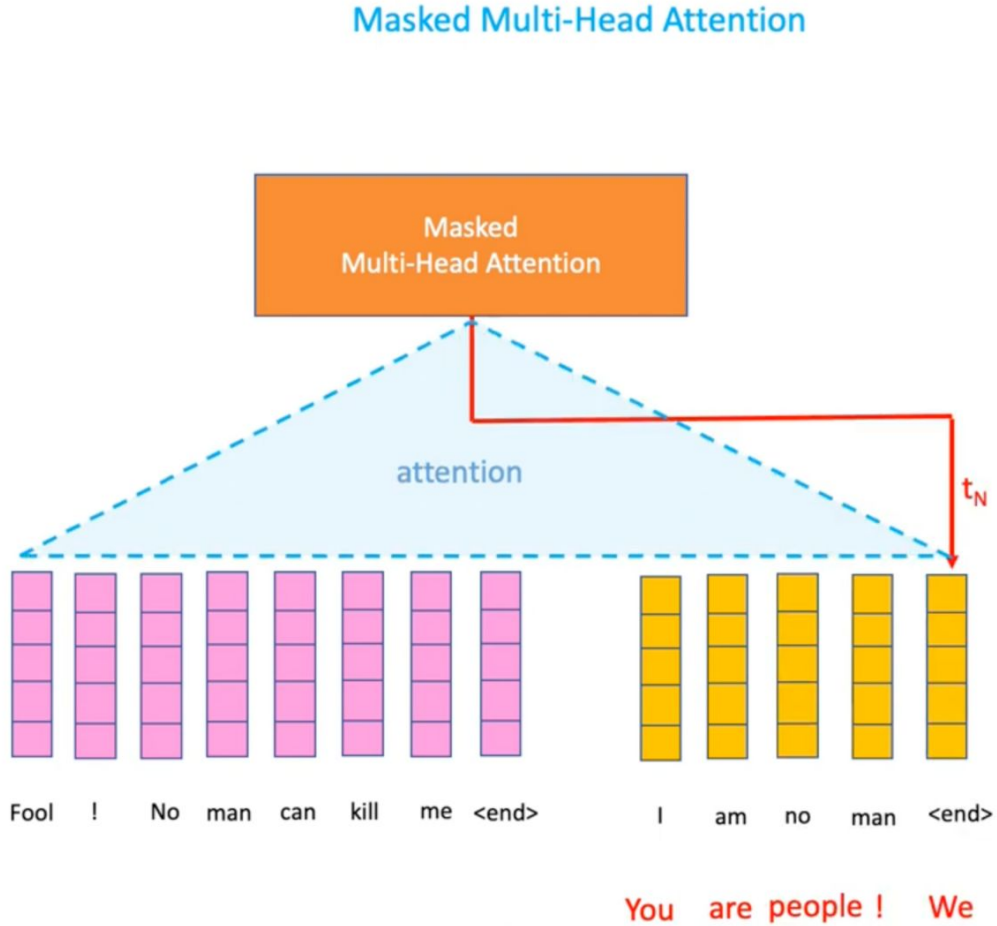
Masked-Attention Filter


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Input Embeddings
Output Embeddings
Inputs
Outputs (shifted right)
Masked Multi-Head Attention
Masked Multi-Head Attention
attention
t_0
Fool
!
No
man
can
kill
me
<end>
Encoder Output
<start>


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


Masked Multi-Head Attention
Masked Multi-Head Attention
attention
t_1
Fool ! No man can kill me <end> I am no man <eos>
You are


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


Masked Multi-Head Attention
Masked Multi-Head Attention
attention
t_N
Fool ! No man can kill me <end>
I am no man <end>
You are people ! We

Finally you did it!!!
Be ready to fly high in the course

Acknowledgements

Hedu AI by Batool Haider