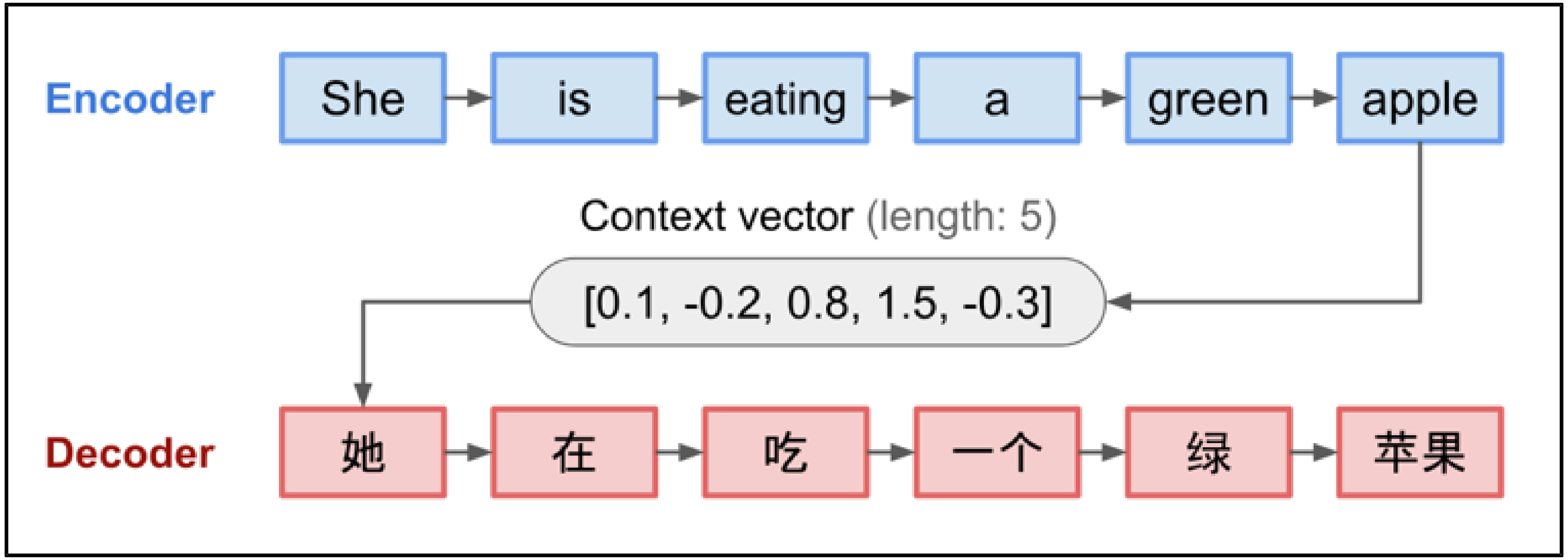
Encoder-decoder models, Attention mechanism

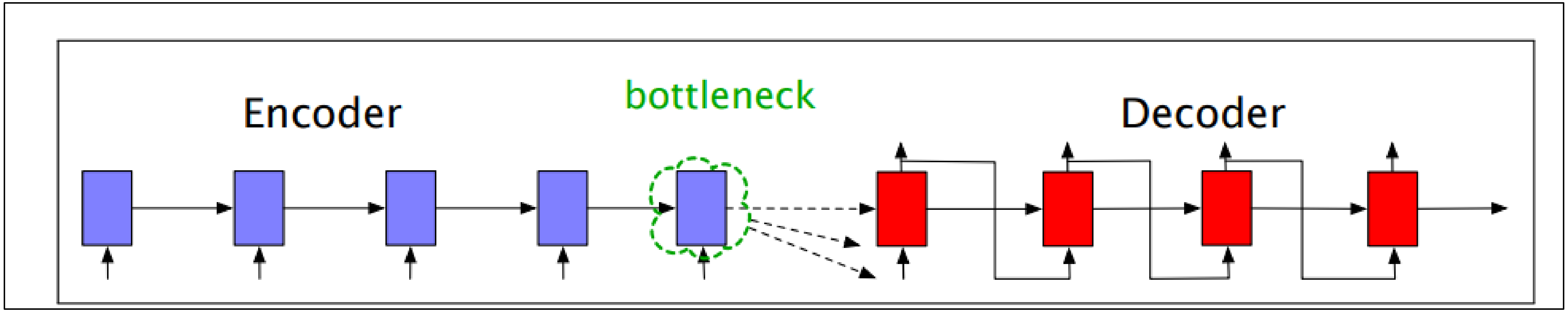
Recap

- Language modeling
- Recurrent Neural Network and Implementation
- Applications of Recurrent Neural Network
- Language modeling using Long Short-term Memory

Encoder-Decoder model



Problem- Bottleneck in Encoder-decoder



*Requiring the **context c** to be only the encoder's final hidden state forces all the information from the entire source sentence to pass through this representational bottleneck*

Problems with Sequence to Sequence models

- fixed-length context vector design
  - incapability of remembering long sentences

Imagine the whole universe in all its beauty - try to visualize everything you can find there and how you can describe it in words. Then imagine all of it is compressed into a single vector of size e.g. 512. Do you feel that the universe is still ok?

Not only it is hard for the encoder to put all information into a single vector - this is also hard for the decoder.

The decoder sees only one representation of source. However, at each generation step, different parts of source can be more useful than others.

https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

Solution to bottleneck problem: Attention

Allow the decoder to get information from all the hidden states of the encoder, not just the last hidden state.

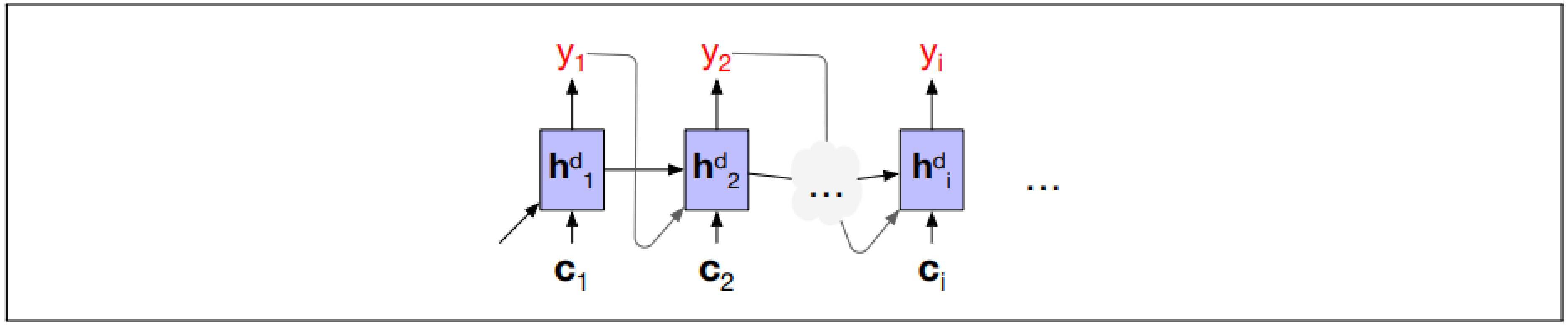


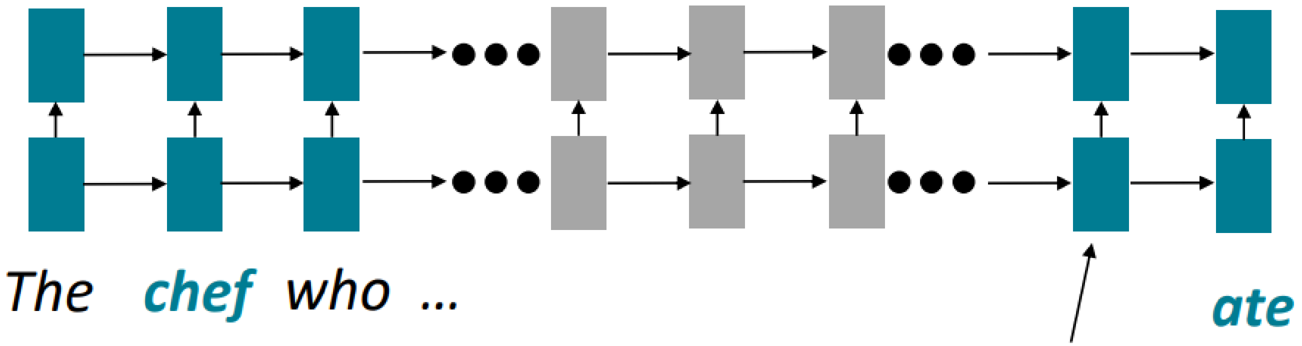
Figure 9.21 The attention mechanism allows each hidden state of the decoder to see a different, dynamic, context, which is a function of all the encoder hidden states.

Image Reference: Speech and Language Processing by Daniel Jurafsky and James H. Martin

Issues with Recurrent models: Linear interaction distance

O(sequence length) steps for distant word pairs to interact means:

- Hard to learn long-distance dependencies (because gradient problems!)
- Linear order of words is "baked in"; we already know sequential structure doesn't tell the whole story...

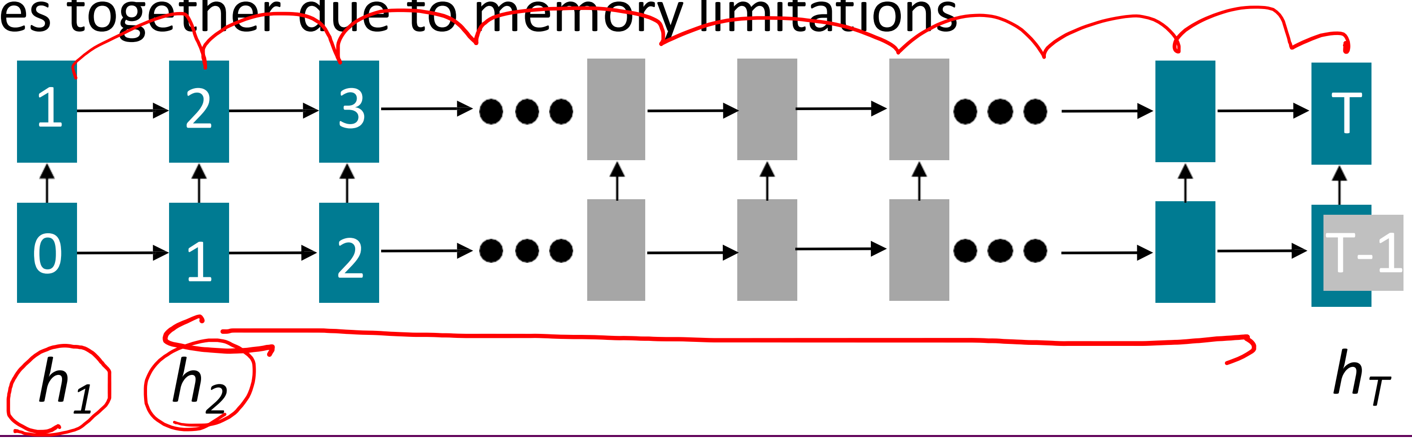


Info of ***chef*** has gone through O(sequence length) many layers!

[Adapted from cs224n-2024-lecture08-transformers.pdf (stanford.edu)](#)

Lack of Parallelizability in RNN

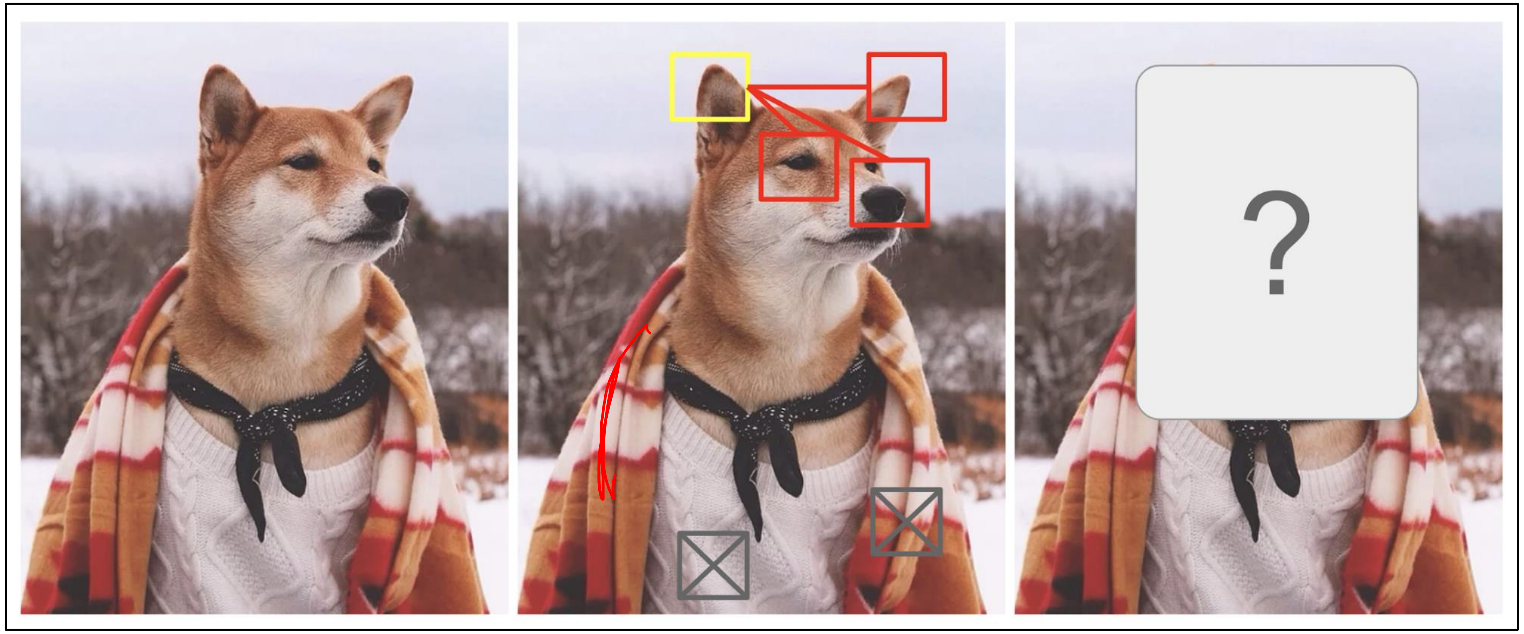
- Forward and backward passes have **O(seq length)** unparallelizable operations
  - GPUs (and TPUs) can perform many independent computations at once!
  - But future RNN hidden states can't be computed in full before past RNN hidden states have been computed
  - Inhibits training on very large datasets!
  - Particularly problematic as sequence length increases, as we can no longer batch many examples together due to memory limitations



Numbers indicate min # of steps before a state can be computed

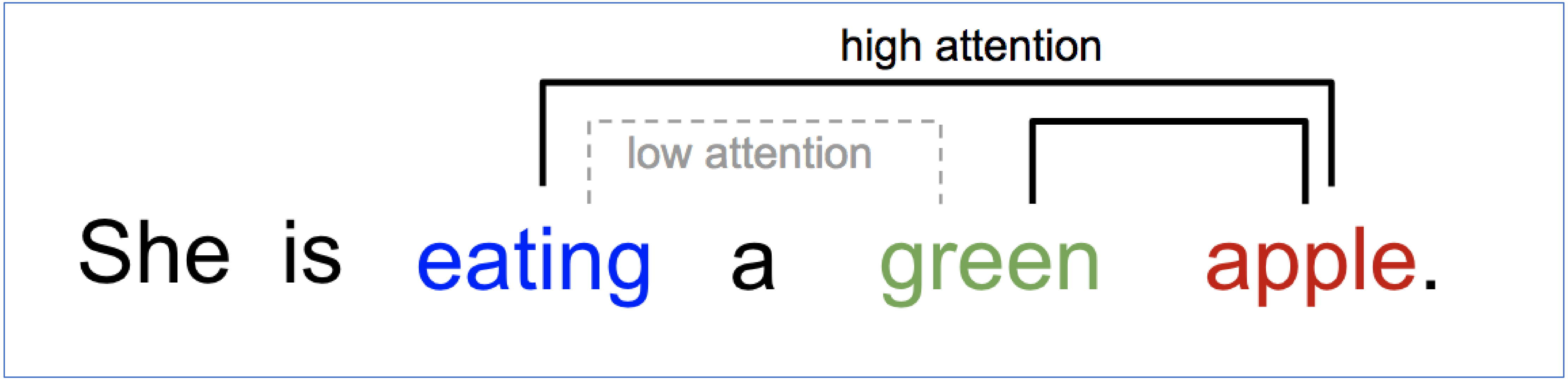
[Adapted from cs224n-2024-lecture08-transformers.pdf (stanford.edu)](#)

Attention



A Shiba Inu in a men's outfit. The credit of the original photo goes to Instagram @mensweardog.
Source: https://lilianweng.github.io/posts/2018-06-24-attention/

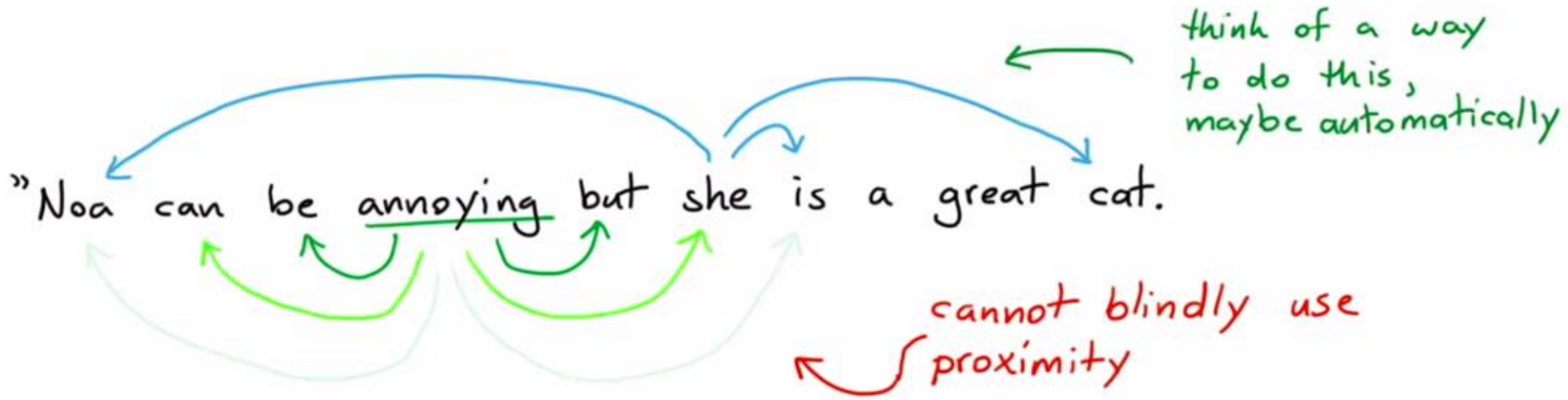
Attention



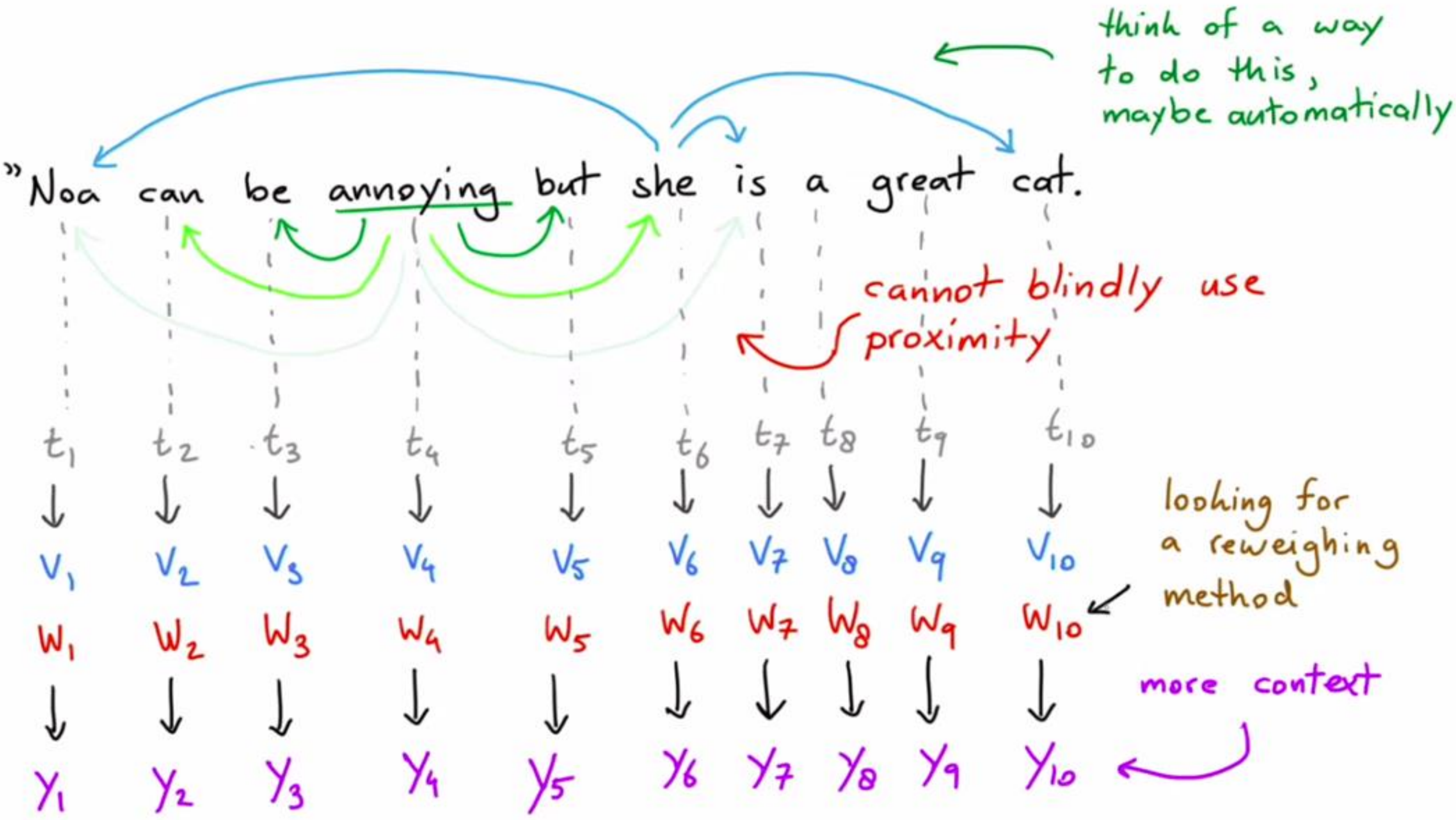
"Noa can be annoying but she is a great cat.

cannot blindly use proximity

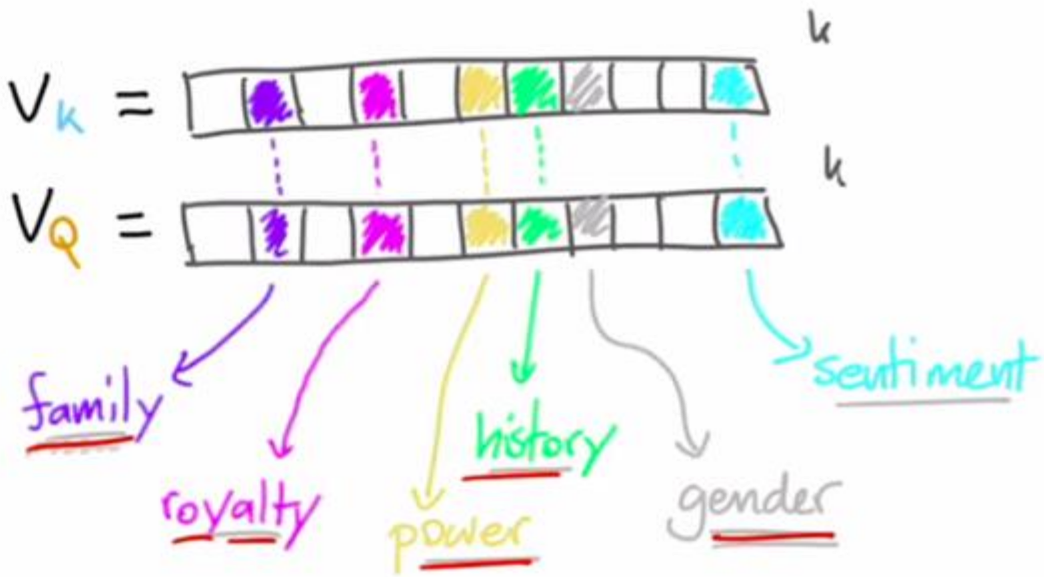
https://learning.rasa.com/transformers/self-attention/



https://learning.rasa.com/transformers/self-attention/



https://learning.rasa.com/transformers/self-attention/



son
daughter
cousin
land
country
army
own

dog
cat
running
swimming
help
nurse
computer

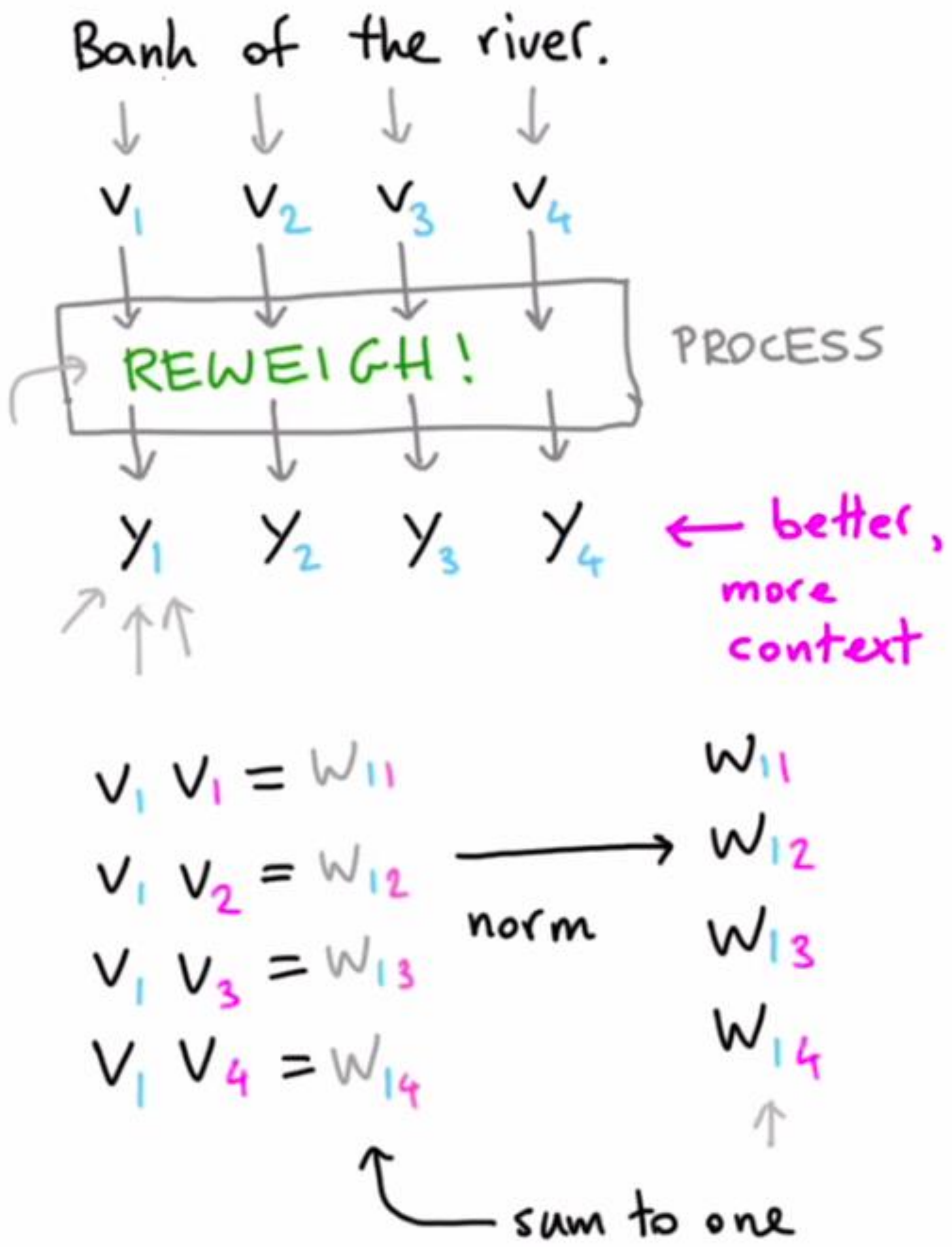
These words share meaning even if they're not in proximity!

Reweighing Plan

$$W_{kQ} = V_k \cdot V_Q \quad ?$$

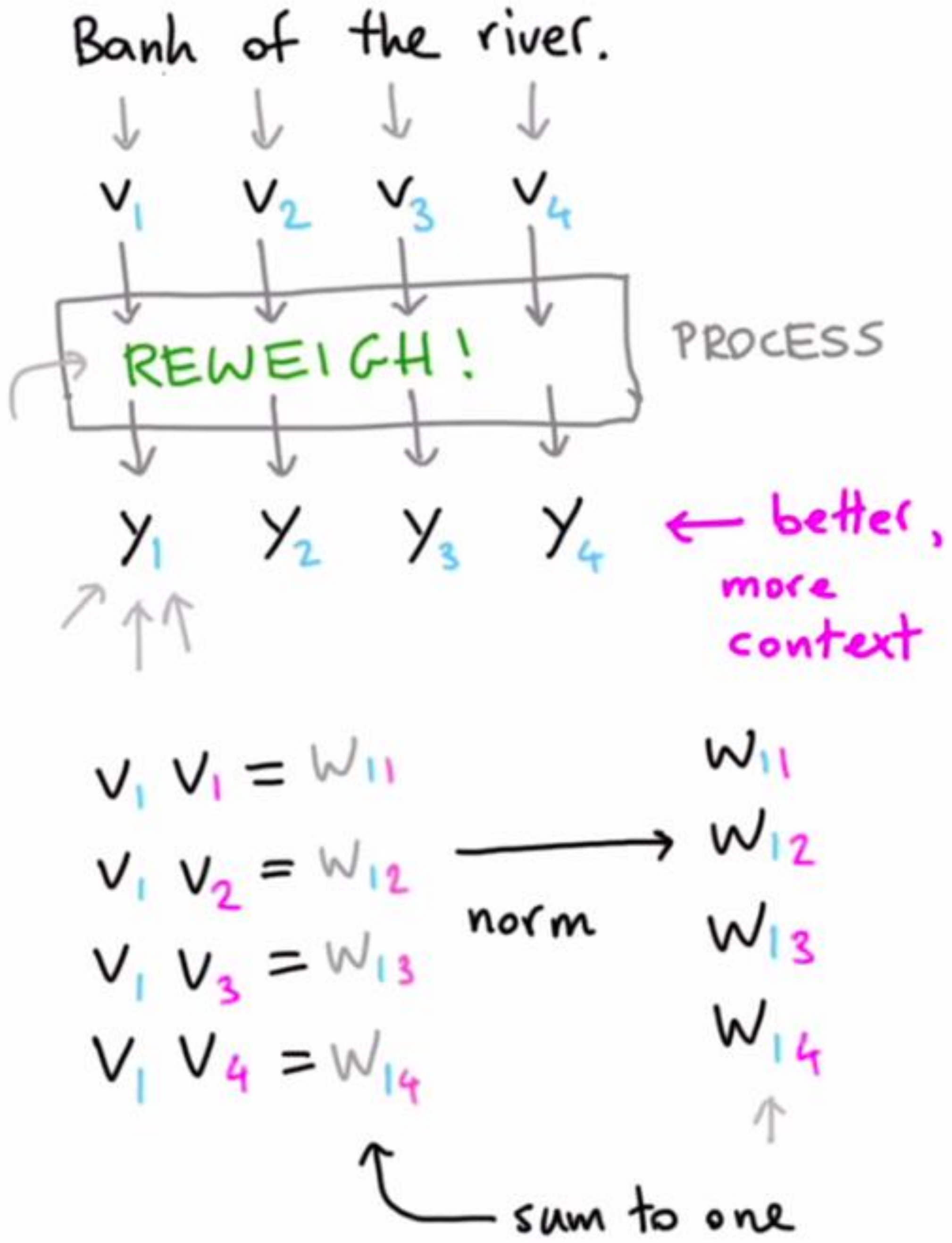
Let's explore this

https://learning.rasa.com/transformers/self-attention/



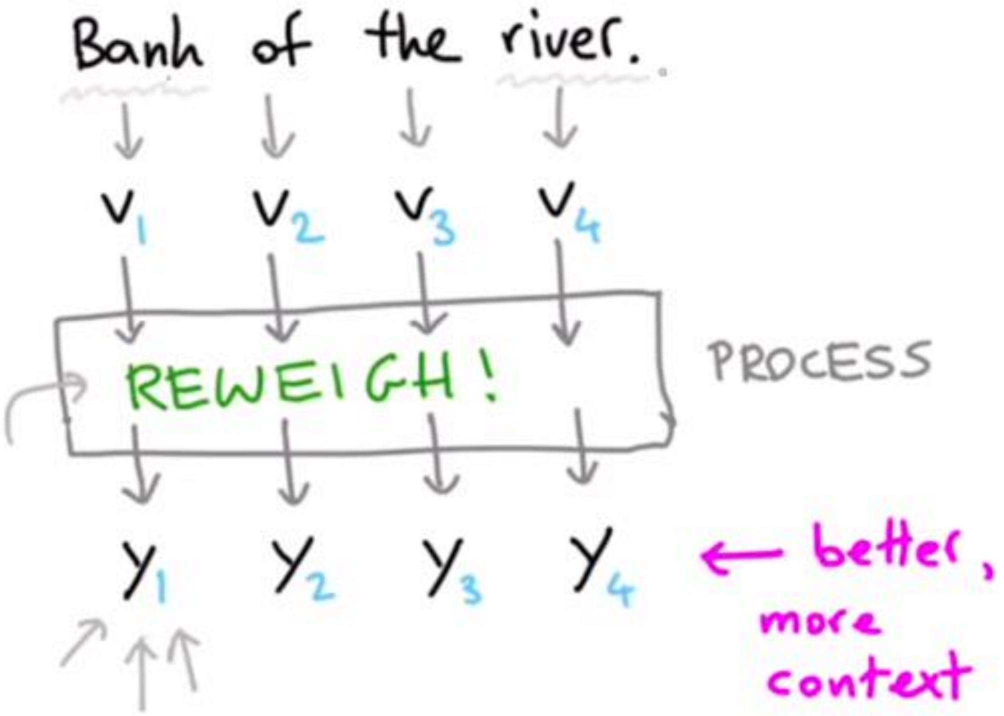
$$w_{11}v_1 + w_{12}v_2 + w_{13}v_3 + w_{14}v_4 = y_1$$

https://learning.rasa.com/transformers/self-attention/



$$w_{11}v_1 + w_{12}v_2 + w_{13}v_3 + w_{14}v_4 = y_1$$

https://learning.rasa.com/transformers/self-attention/



$$w_{11}v_1 + w_{12}v_2 + w_{13}v_3 + w_{14}v_4 = y_1$$

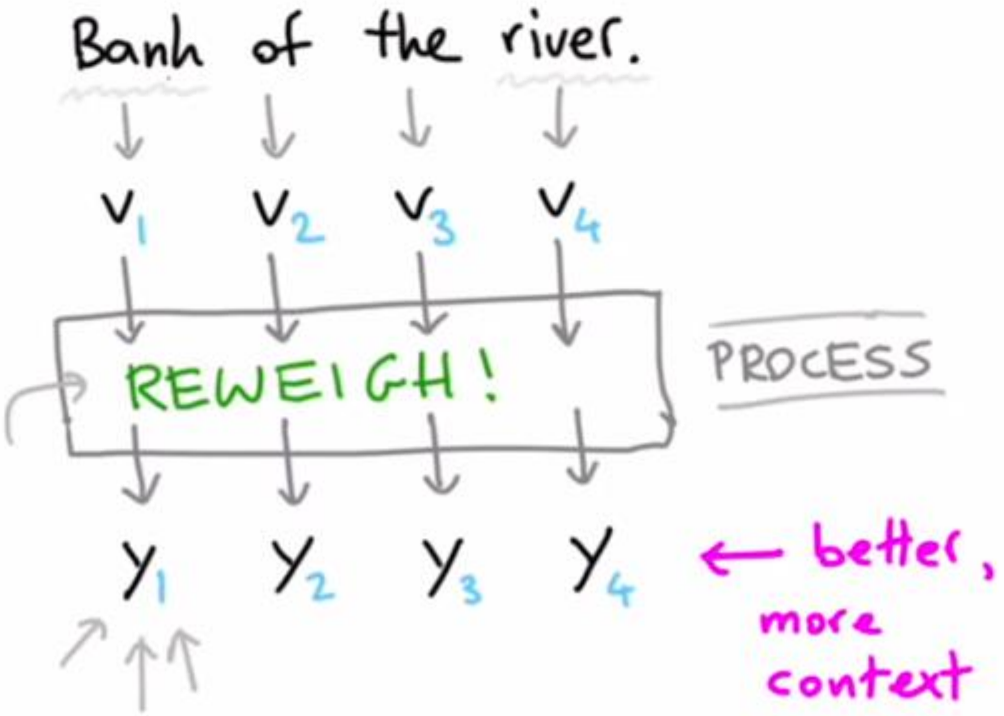
reweigh all vectors towards v_1

$$v_1 v_1 = w_{11}$$
$$v_1 v_2 = w_{12}$$
$$v_1 v_3 = w_{13}$$
$$v_1 v_4 = w_{14}$$

$\xrightarrow{\text{norm}}$ w_{11}, w_{12}, w_{13}, w_{14}

sum to one

https://learning.rasa.com/transformers/self-attention/



$$w_{11}v_1 + w_{12}v_2 + w_{13}v_3 + w_{14}v_4 = y_1$$

$$w_{21}v_1 + w_{22}v + w_{23}v_3 + w_{24}v_4 = y_2$$

$$w_{31}v_1 + w_{32}v_2 + w_{33}v_3 + w_{34}v_4 = y_3$$

$$w_{41}v_1 + w_{42}v_2 + w_{43}v_3 + w_{44}v_4 = y_4$$

$$
\left.\begin{array}{l}
v_1 v_1 = w_{11} \\
v_1 v_2 = w_{12} \\
v_1 v_3 = w_{13} \\
v_1 v_4 = w_{14}
\end{array}\right.
\xrightarrow[\text{norm}]{}
\begin{array}{l}
w_{11} \\
w_{12} \\
w_{13} \\
w_{14}
\end{array}
$$

sum to one

https://learning.rasa.com/transformers/self-attention/

Bank of the river.

$v_1 \quad v_2 \quad v_3 \quad v_4$

REWEIGH!

PROCESS

$y_1 \quad y_2 \quad y_3 \quad y_4$ ← better, more context

$$w_{11}v_1 + w_{12}v_2 + w_{13}v_3 + w_{14}v_4 = y_1$$
$$w_{21}v_1 + w_{22}v_2 + w_{23}v_3 + w_{24}v_4 = y_2$$
$$w_{31}v_1 + w_{32}v_2 + w_{33}v_3 + w_{34}v_4 = y_3$$
$$w_{41}v_1 + w_{42}v_2 + w_{43}v_3 + w_{44}v_4 = y_4$$

$$v_1 v_1 = w_{11}$$
$$v_1 v_2 = w_{12}$$
$$v_1 v_3 = w_{13}$$
$$v_1 v_4 = w_{14}$$

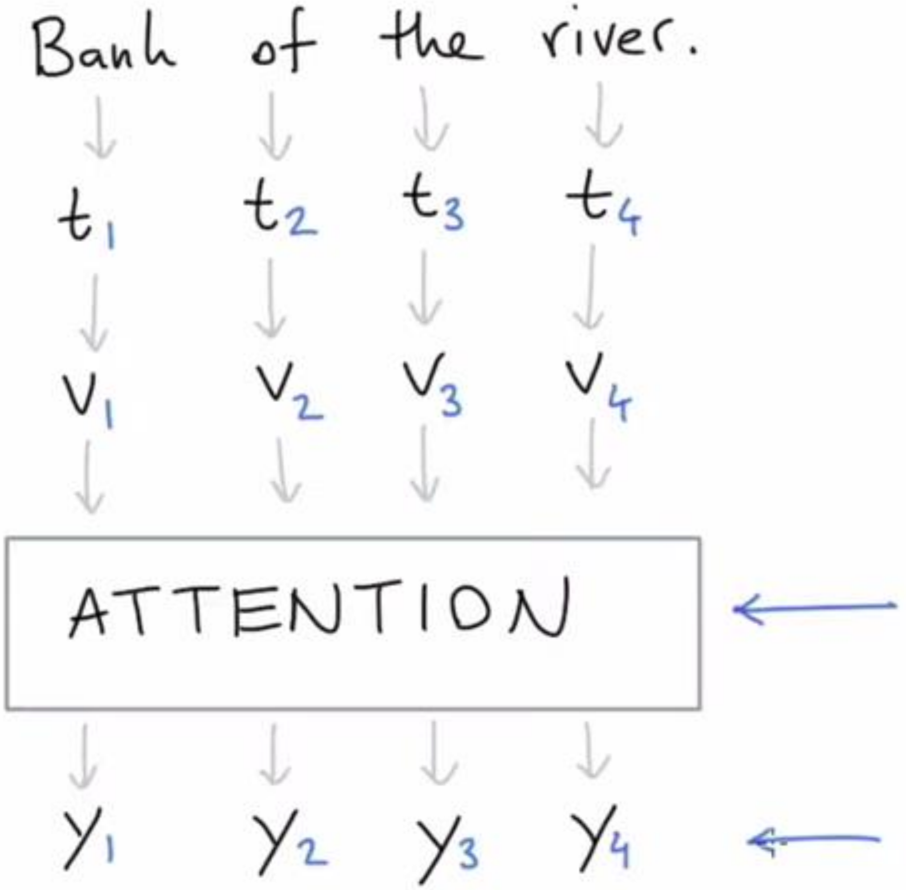
$\xrightarrow{\text{norm}}$ w_{11}, w_{12}, w_{13}, w_{14}

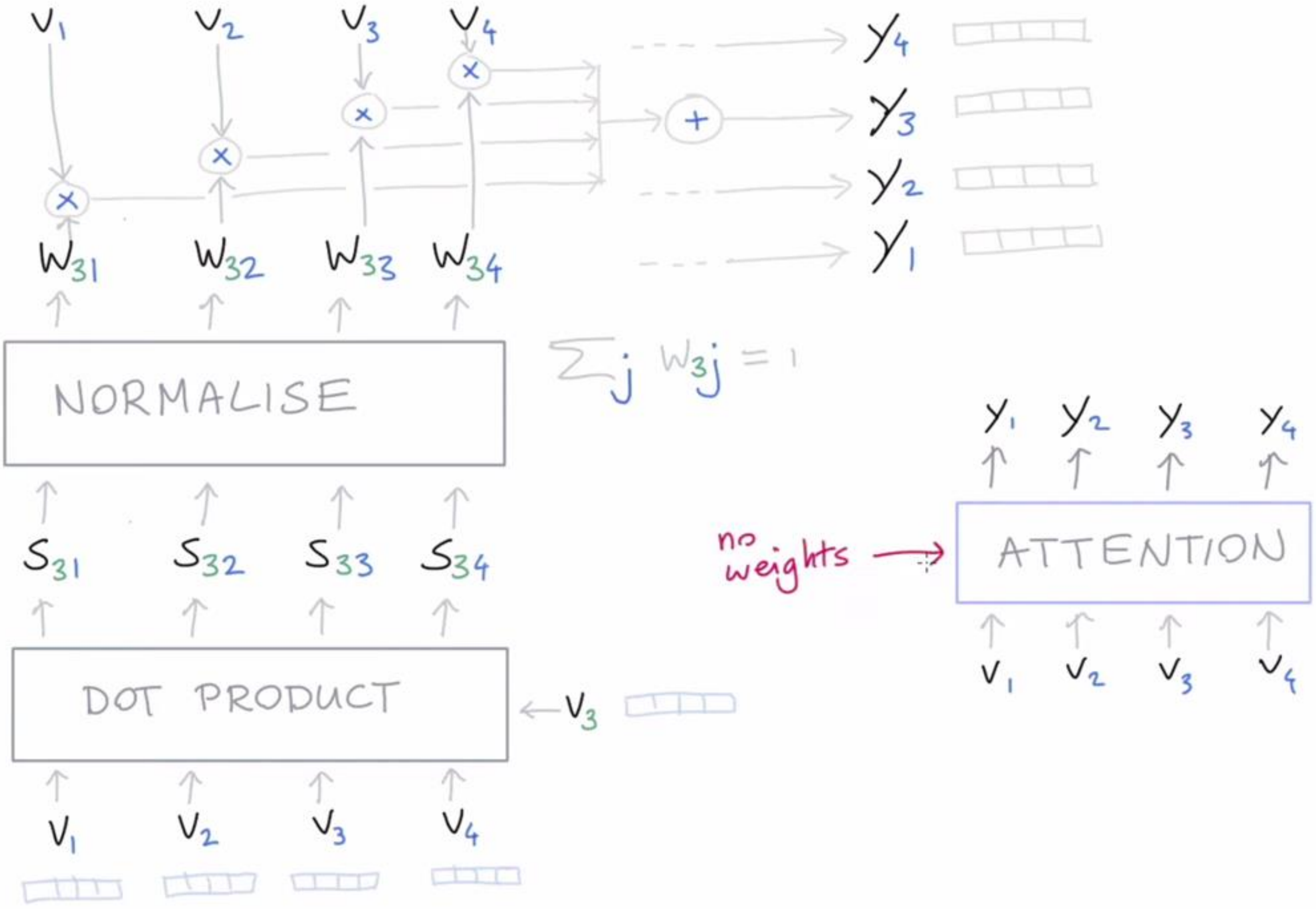
sum to one

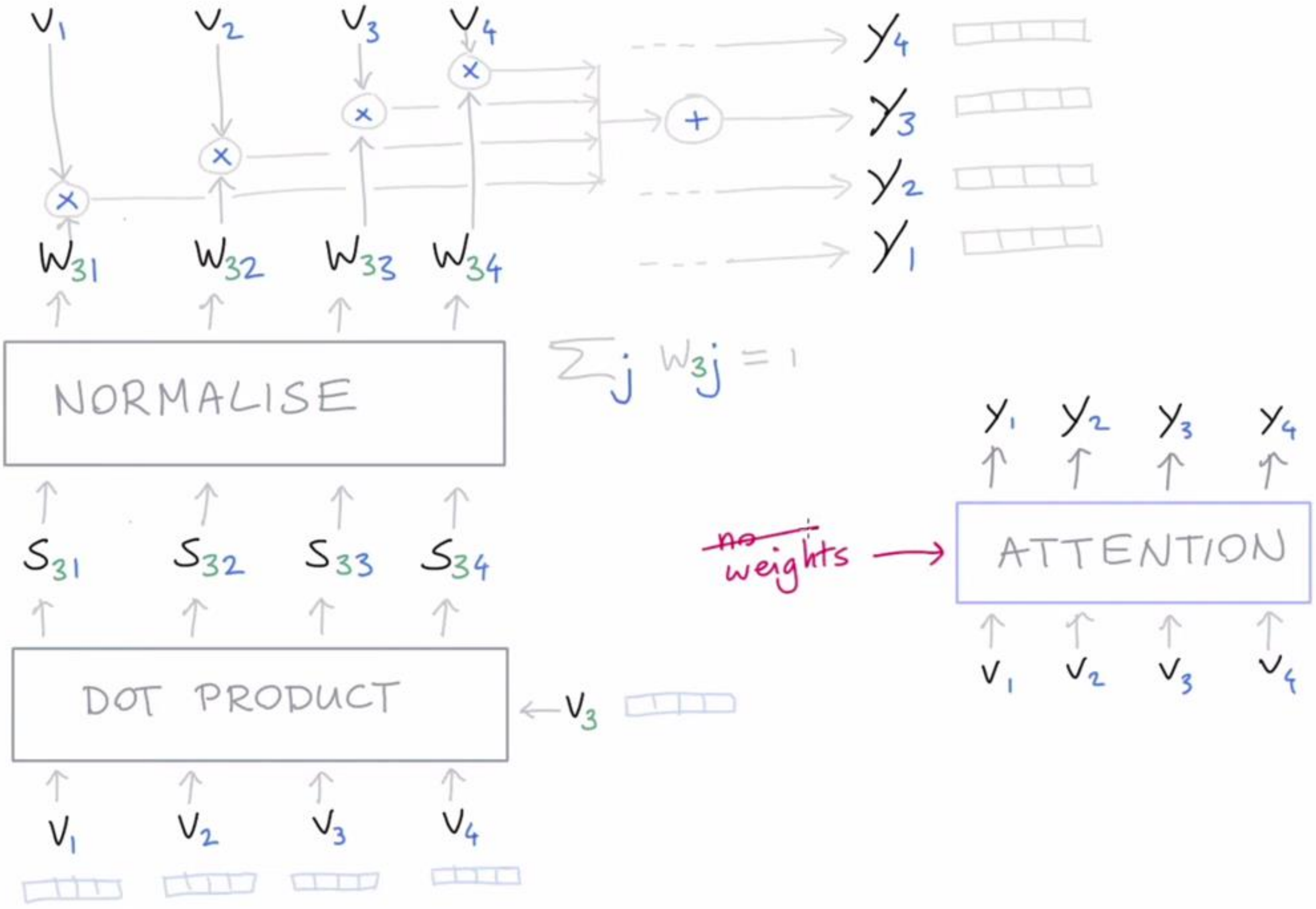
- I've not trained any weights
- Order has no influence
- Proximity has no influence
- Shape independant

$\boxed{\text{SELF ATTENTION}}$

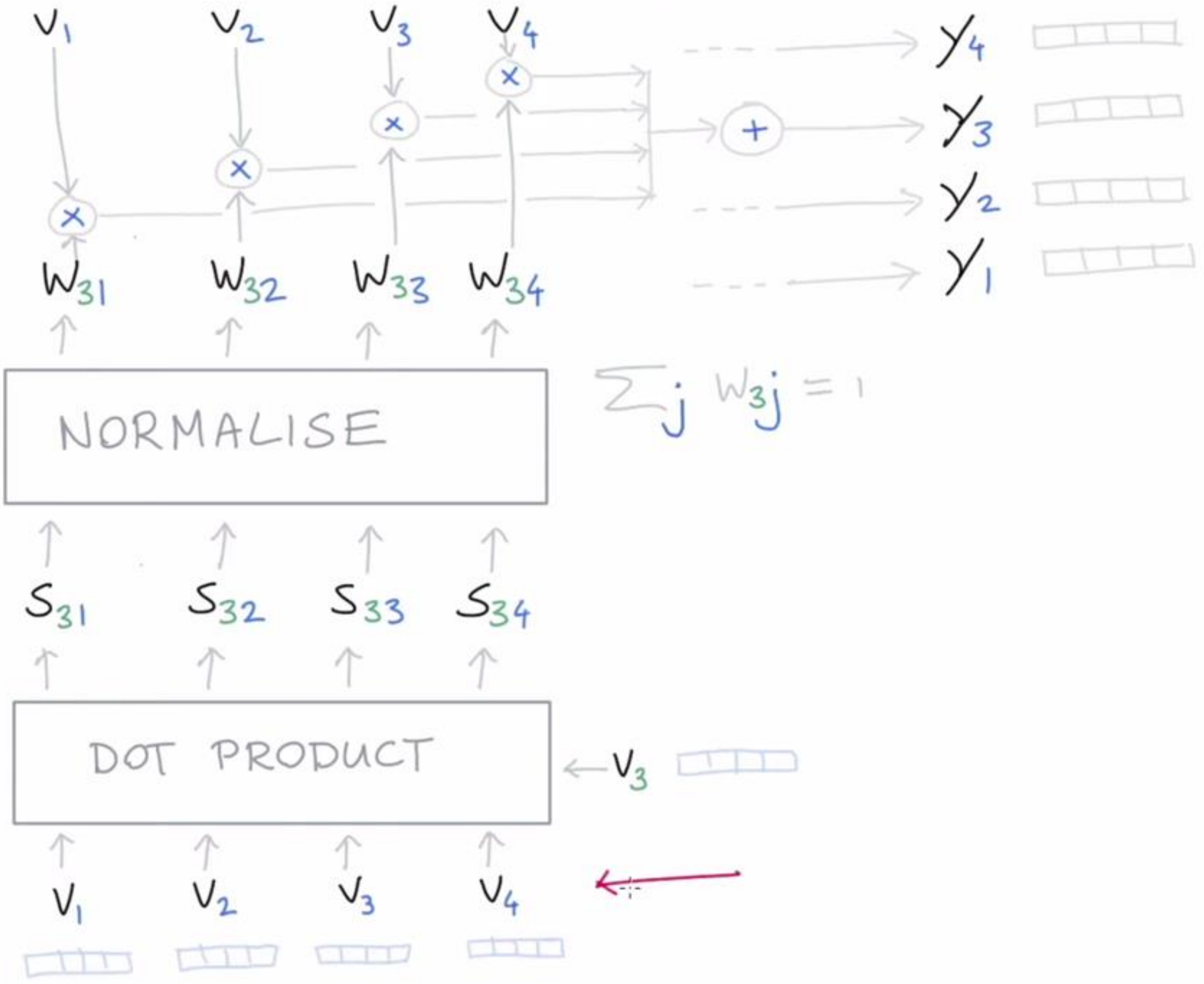
https://learning.rasa.com/transformers/self-attention/


Bank of the river.
t_1 t_2 t_3 t_4
v_1 v_2 v_3 v_4
ATTENTION
y_1 y_2 y_3 y_4

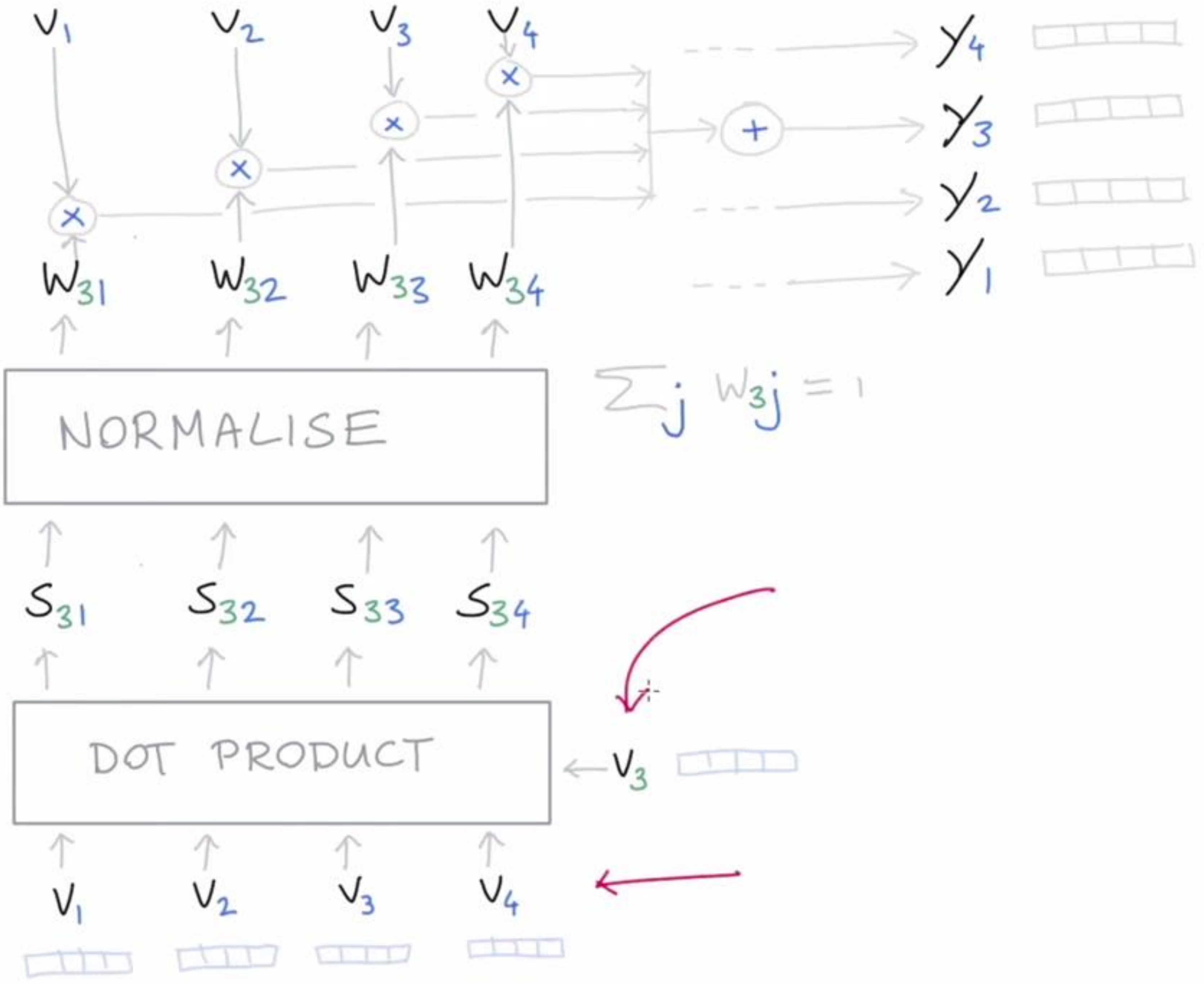

v1 v2 v3 v4
y4
y3
y2
y1
w31 w32 w33 w34
NORMALISE
Σj w3j = 1
s31 s32 s33 s34
DOT PRODUCT
v3
v1 v2 v3 v4
y1 y2 y3 y4
no weights
ATTENTION
v1 v2 v3 v4



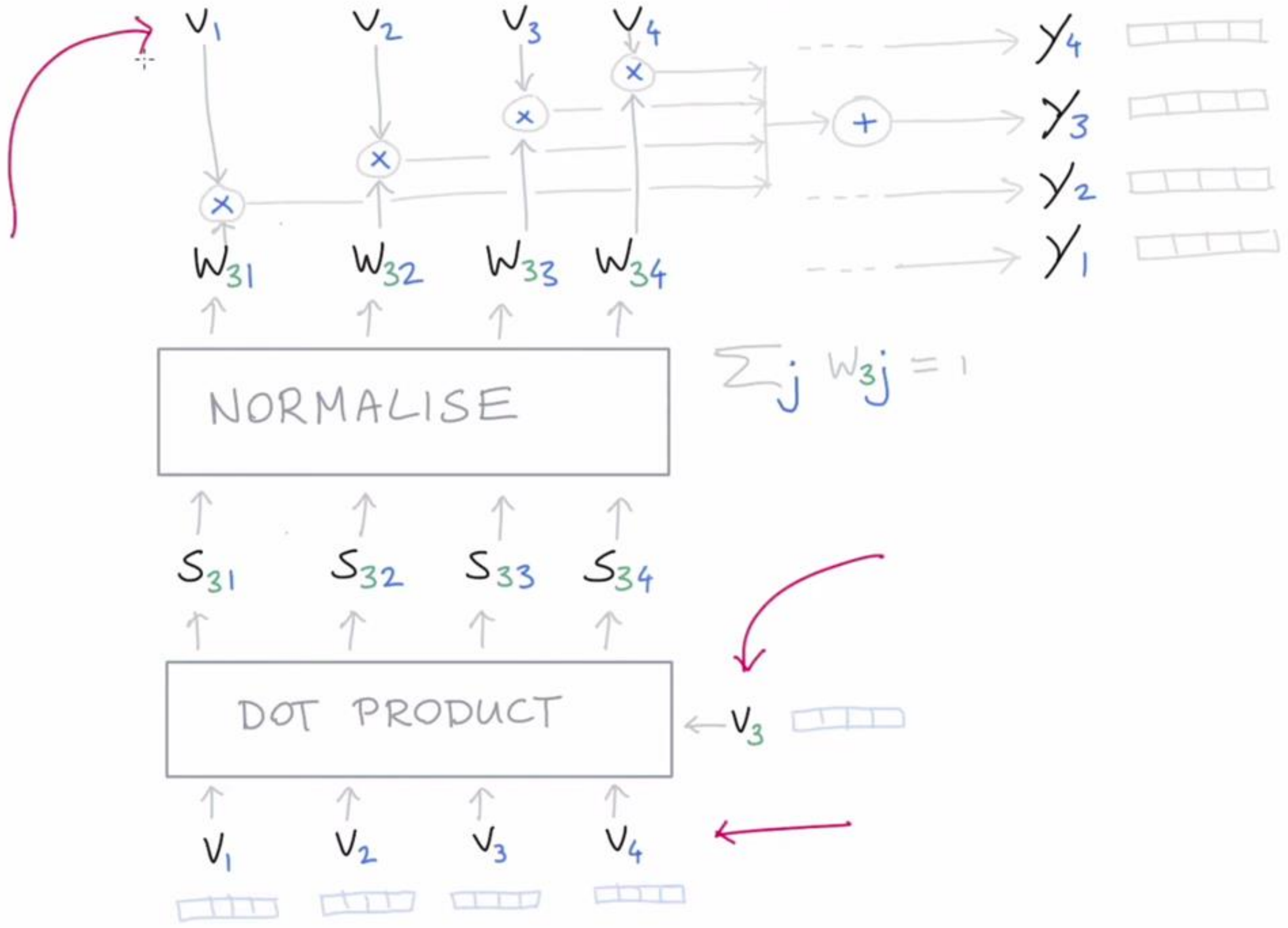
NORMALISE
DOT PRODUCT
ATTENTION
~~no~~ weights
$\sum_j w_{3j} = 1$
v_1 v_2 v_3 v_4
w_{31} w_{32} w_{33} w_{34}
s_{31} s_{32} s_{33} s_{34}
y_1 y_2 y_3 y_4



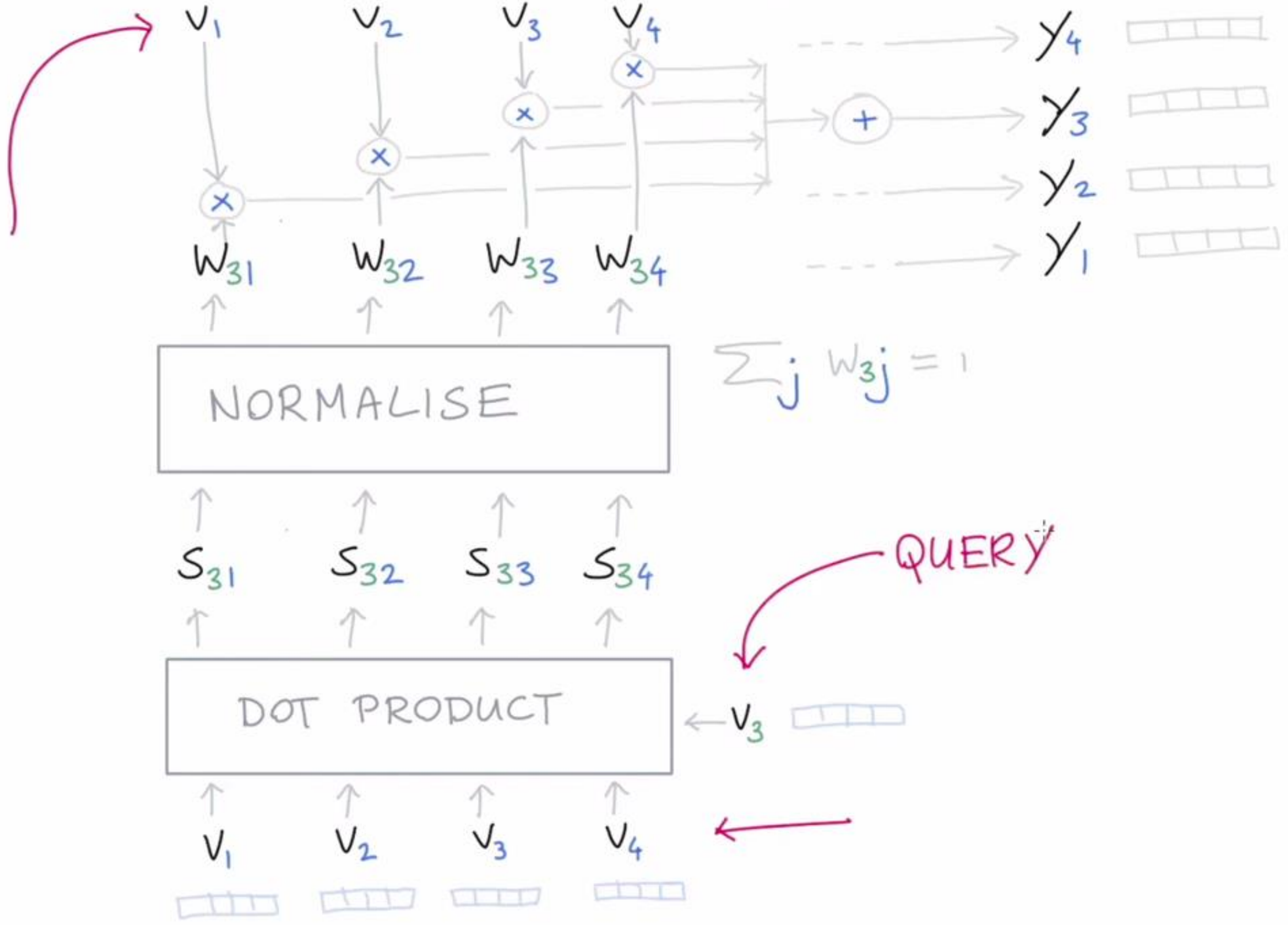
v_1
v_2
v_3
v_4
y_4
y_3
y_2
y_1
W_{31}
W_{32}
W_{33}
W_{34}
NORMALISE
$\sum_j w_{3j} = 1$
S_{31}
S_{32}
S_{33}
S_{34}
DOT PRODUCT
v_3
v_1
v_2
v_3
v_4



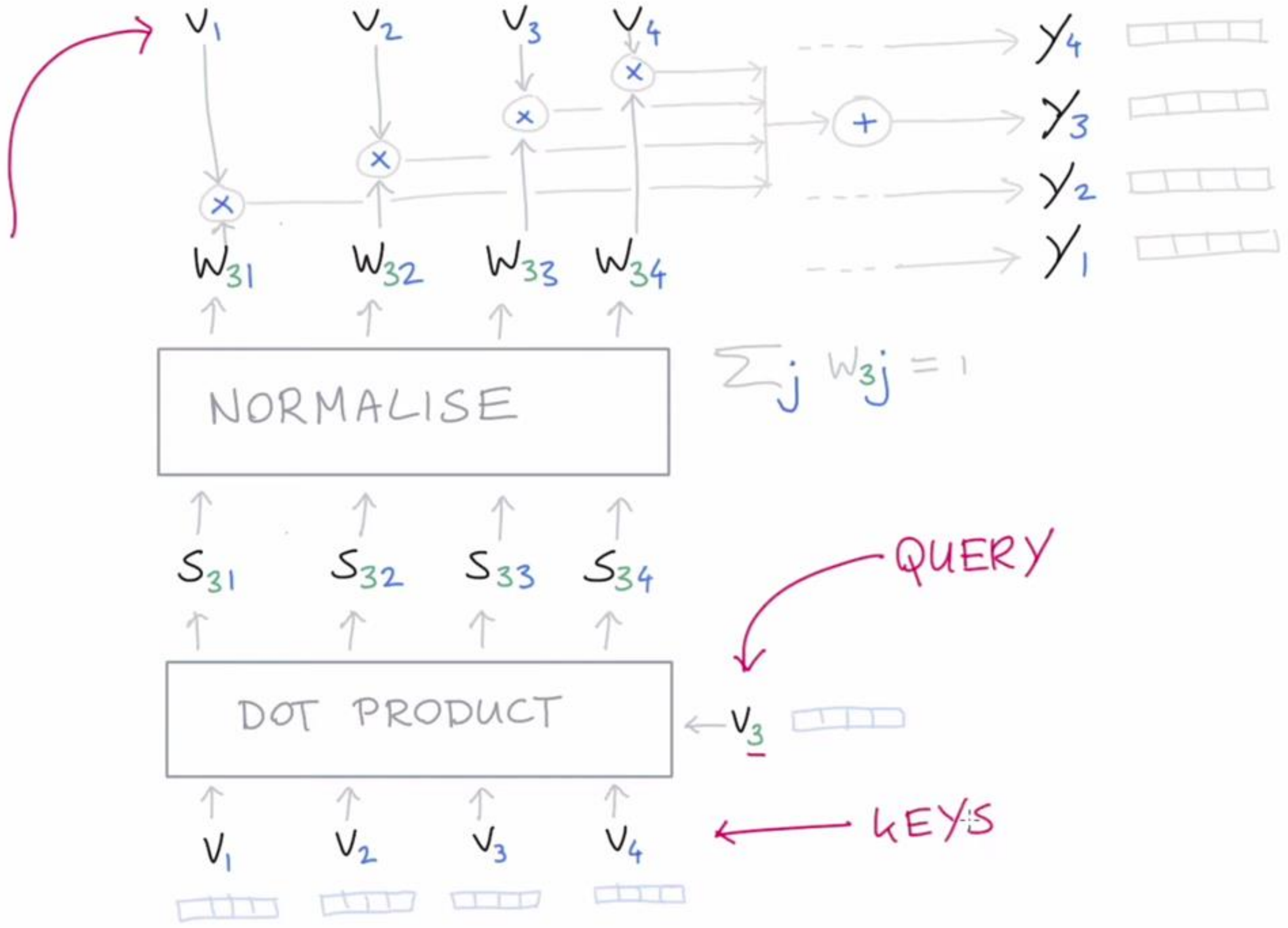
V1 V2 V3 V4
Y4
Y3
Y2
Y1
W31 W32 W33 W34
NORMALISE
Σj W3j = 1
S31 S32 S33 S34
DOT PRODUCT
V3
V1 V2 V3 V4

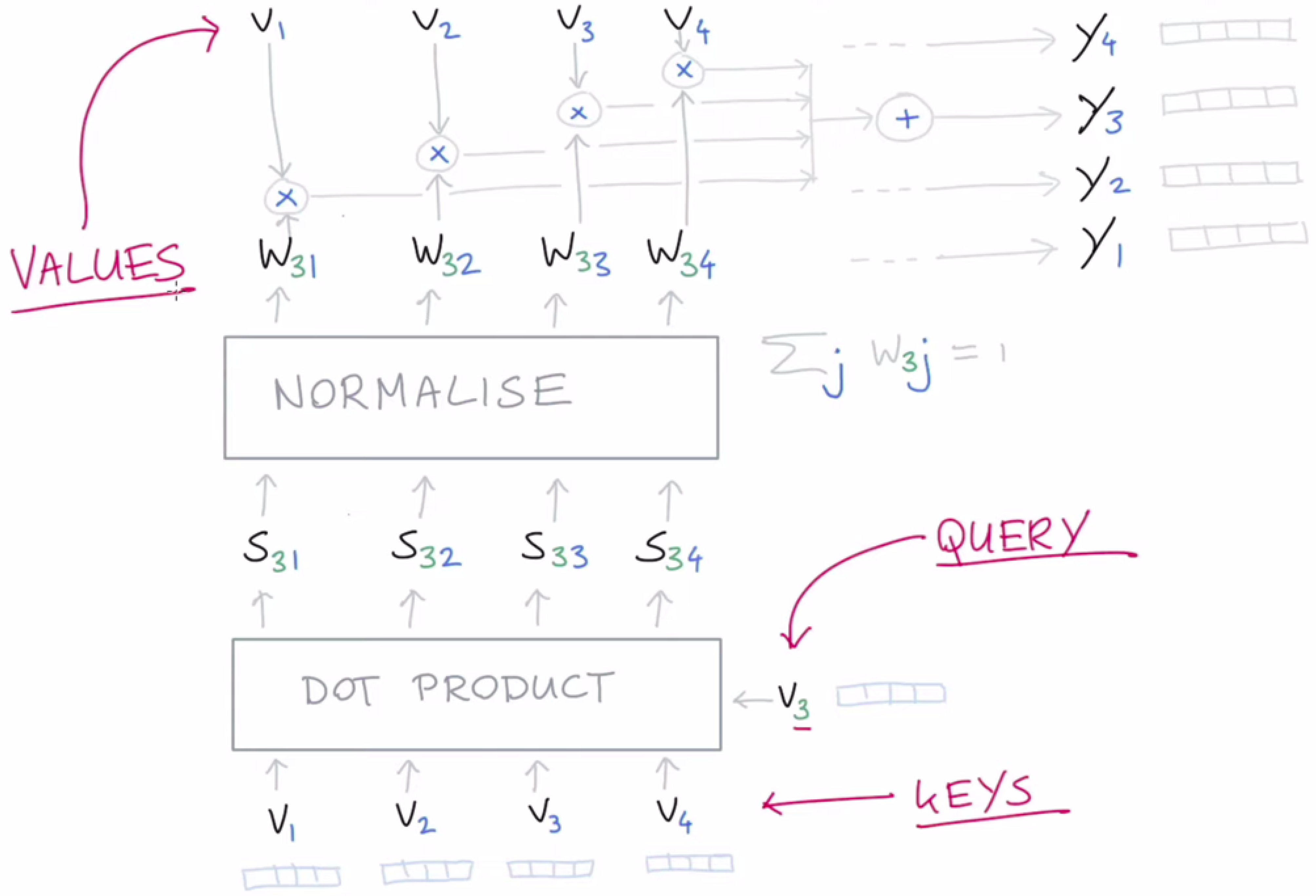


v1 v2 v3 v4
y4
y3
y2
y1
w31 w32 w33 w34
NORMALISE
Σj w3j = 1
s31 s32 s33 s34
DOT PRODUCT
v3
v1 v2 v3 v4



v_1 v_2 v_3 v_4
y_4
y_3
y_2
y_1
w_{31} w_{32} w_{33} w_{34}
NORMALISE
$\sum_j w_{3j} = 1$
QUERY
s_{31} s_{32} s_{33} s_{34}
DOT PRODUCT
v_3
v_1 v_2 v_3 v_4

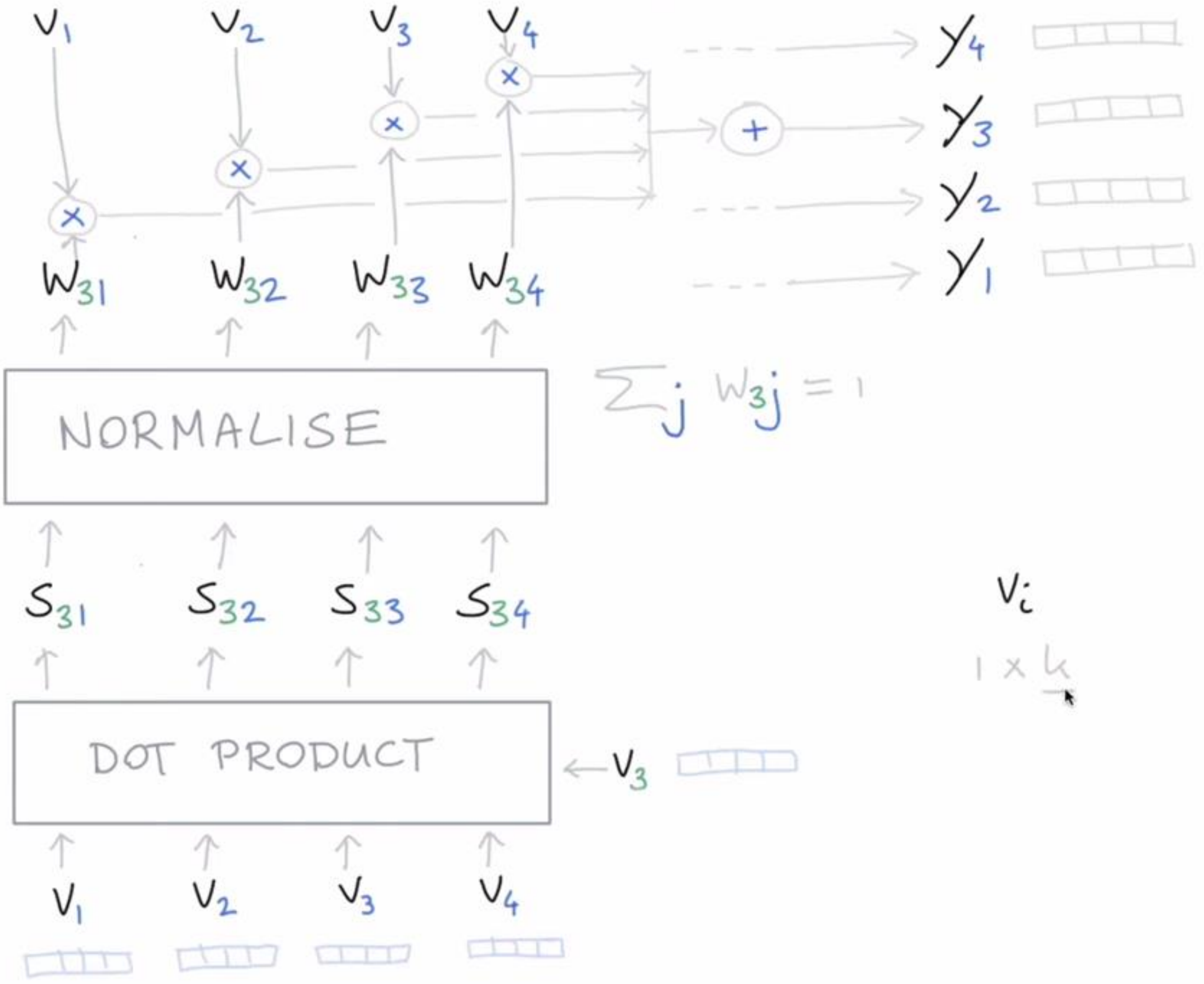

v_1 v_2 v_3 v_4
y_4
y_3
y_2
y_1
w_{31} w_{32} w_{33} w_{34}
NORMALISE
$\sum_j w_{3j} = 1$
s_{31} s_{32} s_{33} s_{34}
QUERY
DOT PRODUCT
v_3
KEYS
v_1 v_2 v_3 v_4



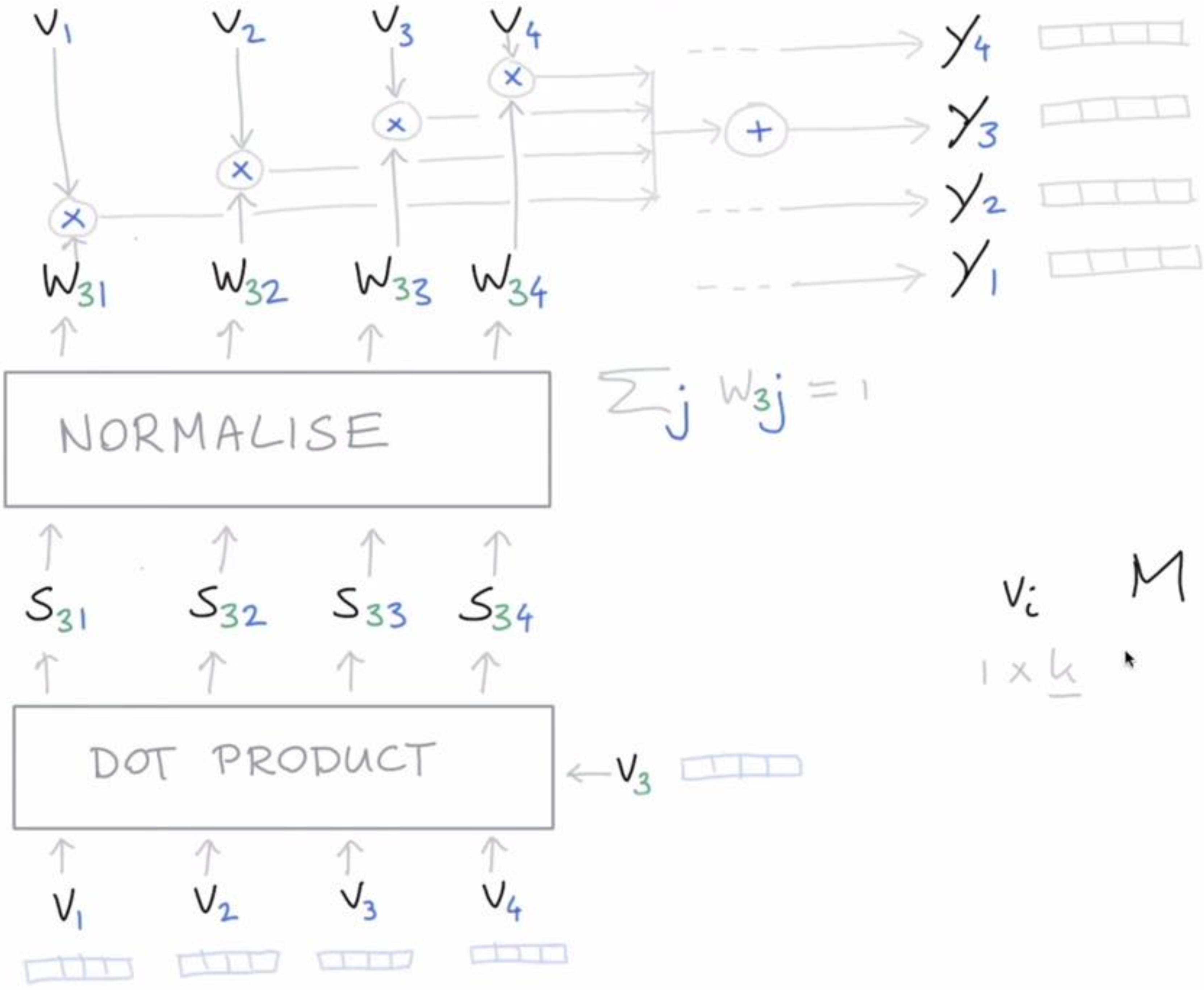
VALUES
v_1 v_2 v_3 v_4
w_{31} w_{32} w_{33} w_{34}
y_4 y_3 y_2 y_1
NORMALISE
$\sum_j w_{3j} = 1$
s_{31} s_{32} s_{33} s_{34}
QUERY
DOT PRODUCT
v_3
KEYS
v_1 v_2 v_3 v_4

Key, Value and Query

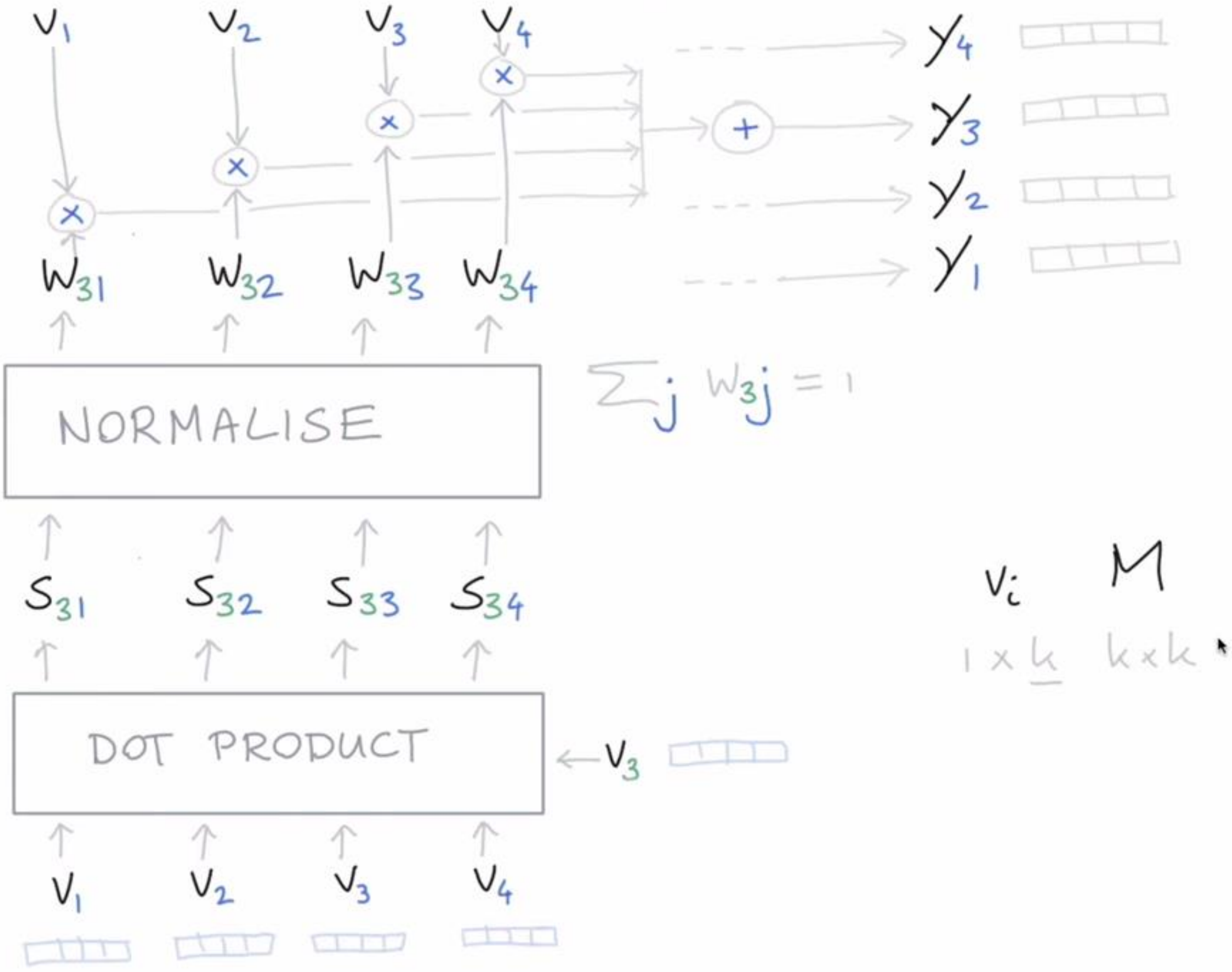
- The major component in the transformer is the unit of multi-head self-attention mechanism.

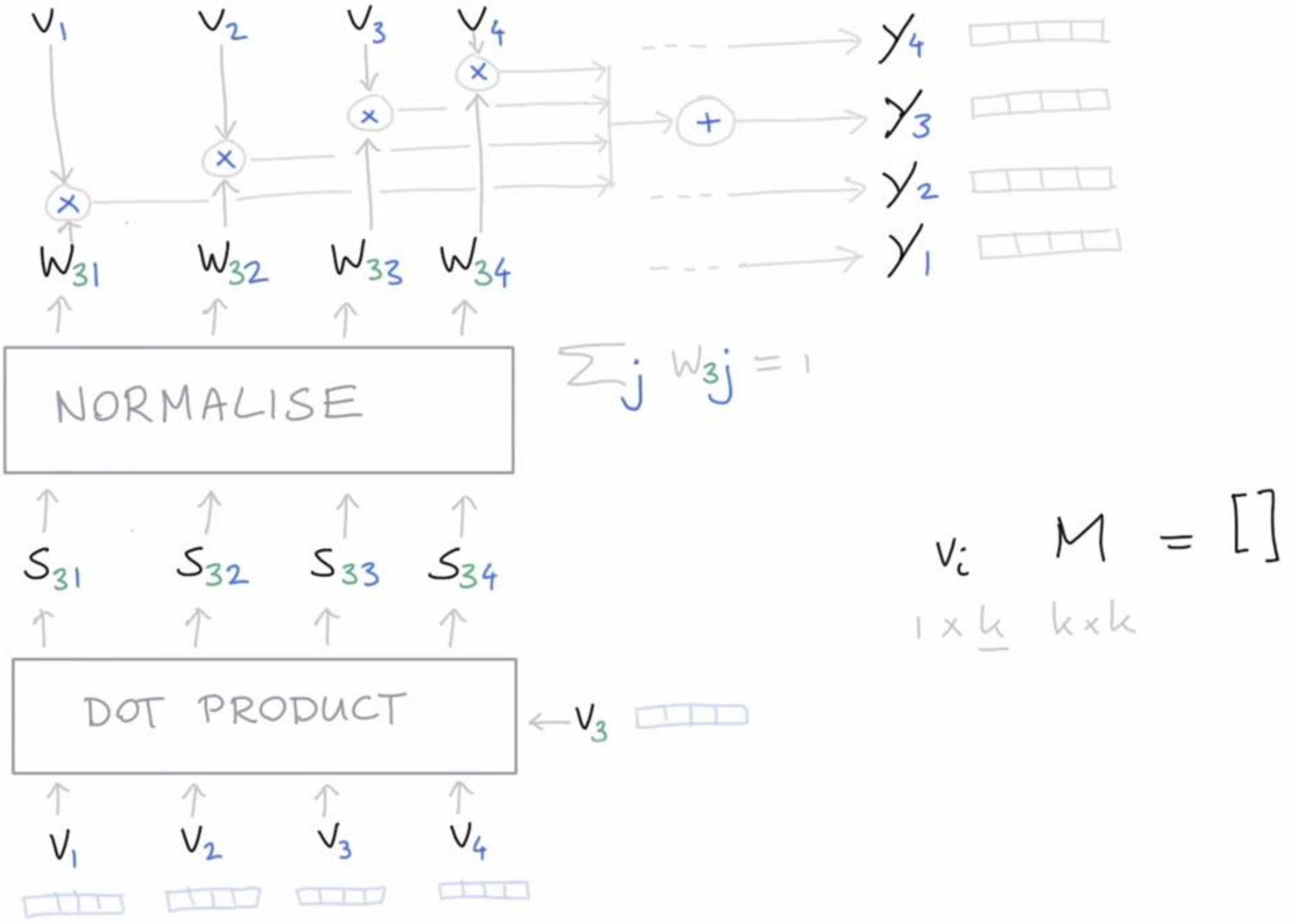


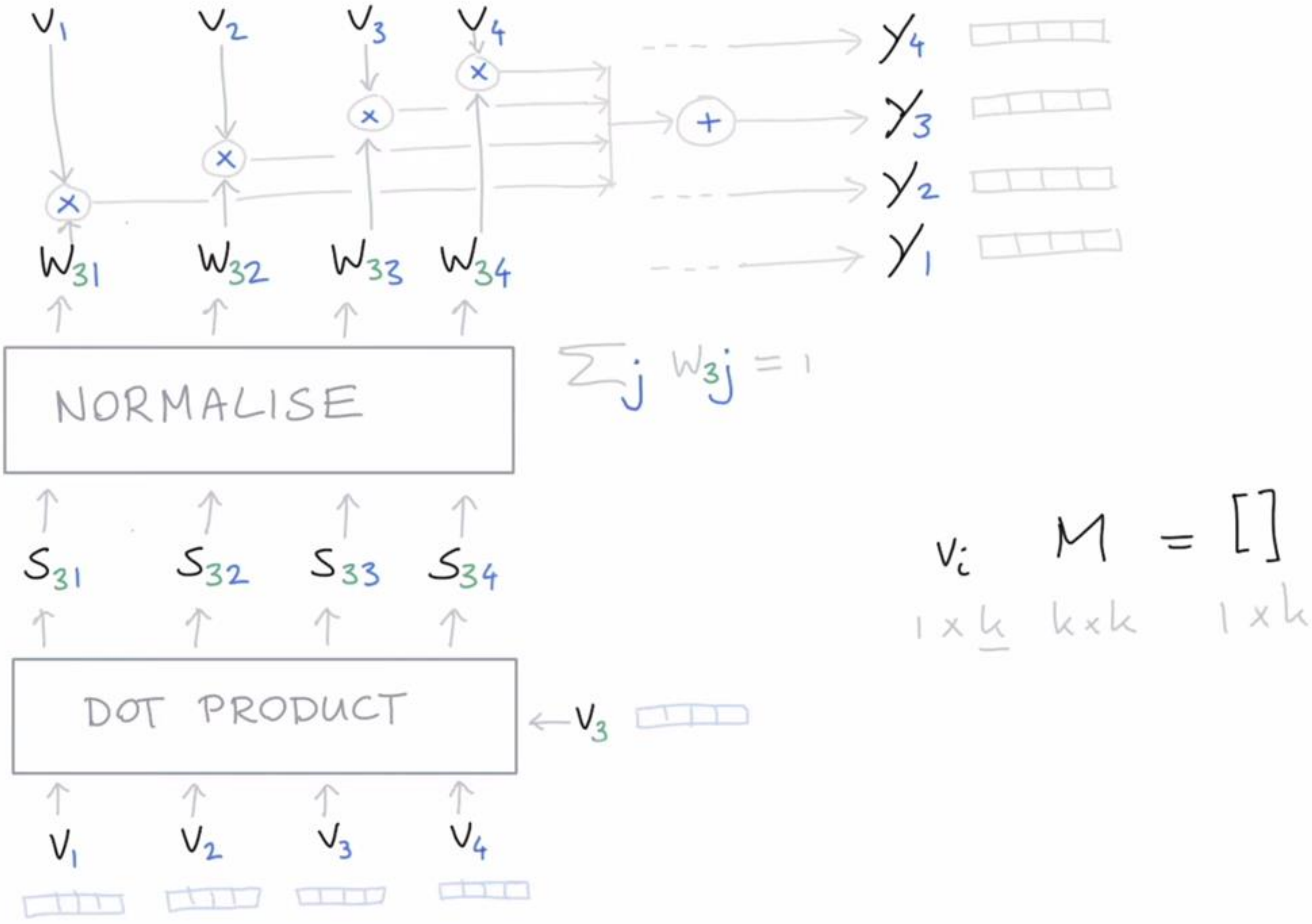
v_1 v_2 v_3 v_4
w_{31} w_{32} w_{33} w_{34}
y_4 y_3 y_2 y_1
NORMALISE
$\sum_j w_{3j} = 1$
s_{31} s_{32} s_{33} s_{34}
v_i
$1 \times k$
DOT PRODUCT
v_3
v_1 v_2 v_3 v_4



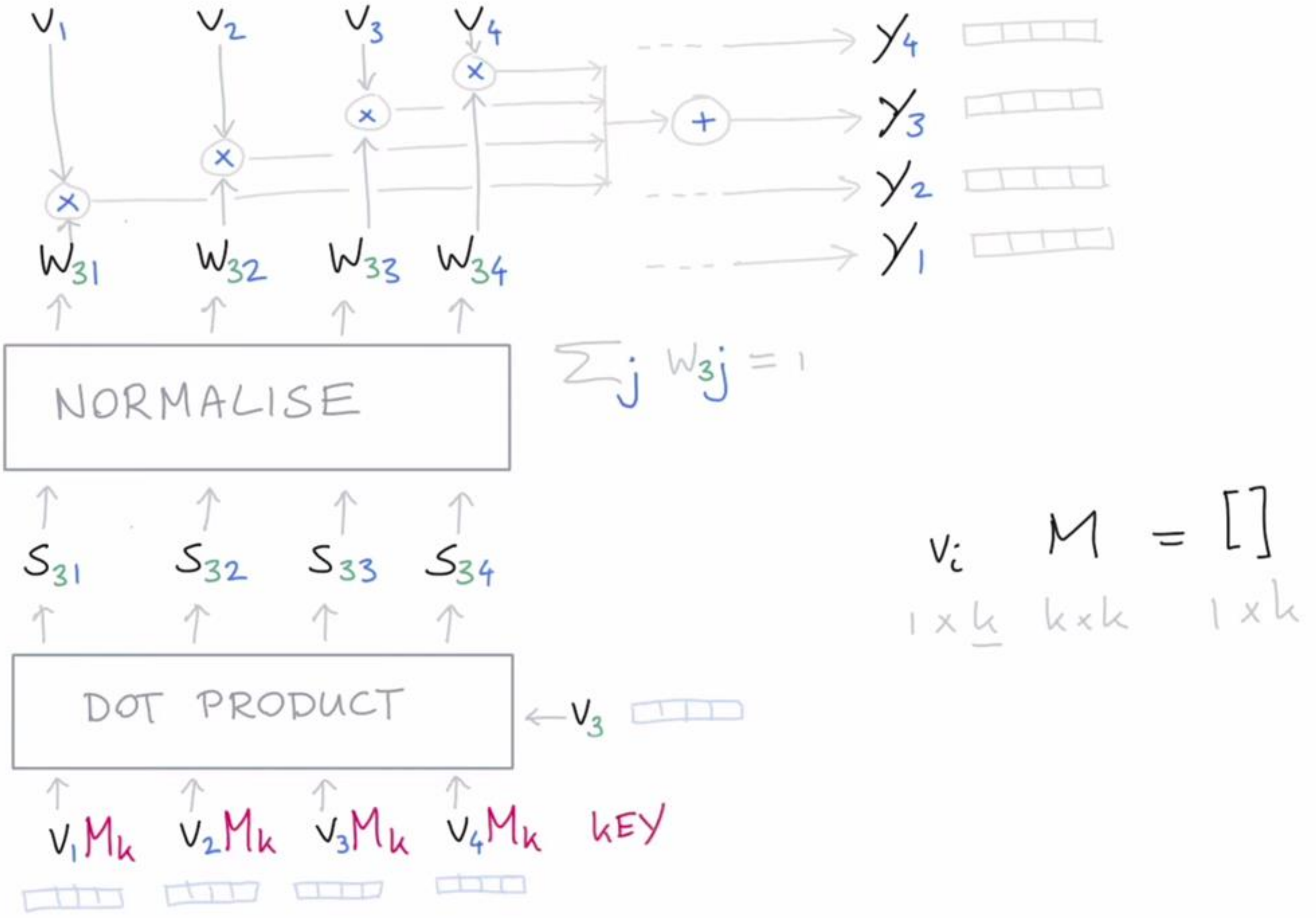
v_1 v_2 v_3 v_4
y_4
y_3
y_2
y_1
w_{31} w_{32} w_{33} w_{34}
NORMALISE
$\sum_j w_{3j} = 1$
s_{31} s_{32} s_{33} s_{34}
v_i M
$1 \times k$
DOT PRODUCT
v_3
v_1 v_2 v_3 v_4


v1
v2
v3
v4
y4
y3
y2
y1
w31
w32
w33
w34
NORMALISE
Σj w3j = 1
s31
s32
s33
s34
vi
M
1 × k
k × k
DOT PRODUCT
v3
v1
v2
v3
v4

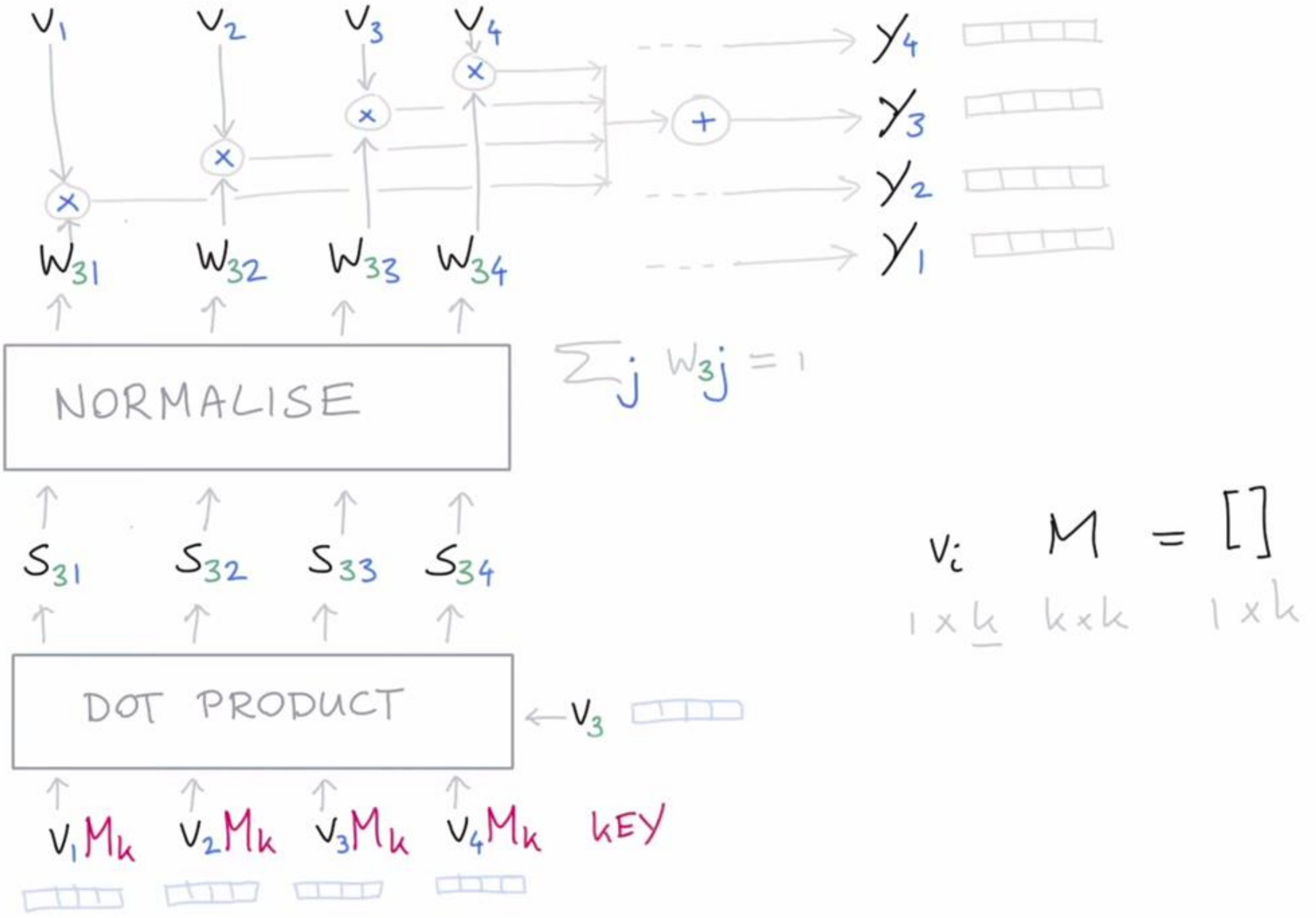

v_1 v_2 v_3 v_4
y_4 y_3 y_2 y_1
+
w_{31} w_{32} w_{33} w_{34}
NORMALISE
$\sum_j w_{3j} = 1$
s_{31} s_{32} s_{33} s_{34}
DOT PRODUCT
v_3
v_1 v_2 v_3 v_4
v_i $M = [\,]$
$1 \times k$ $k \times k$



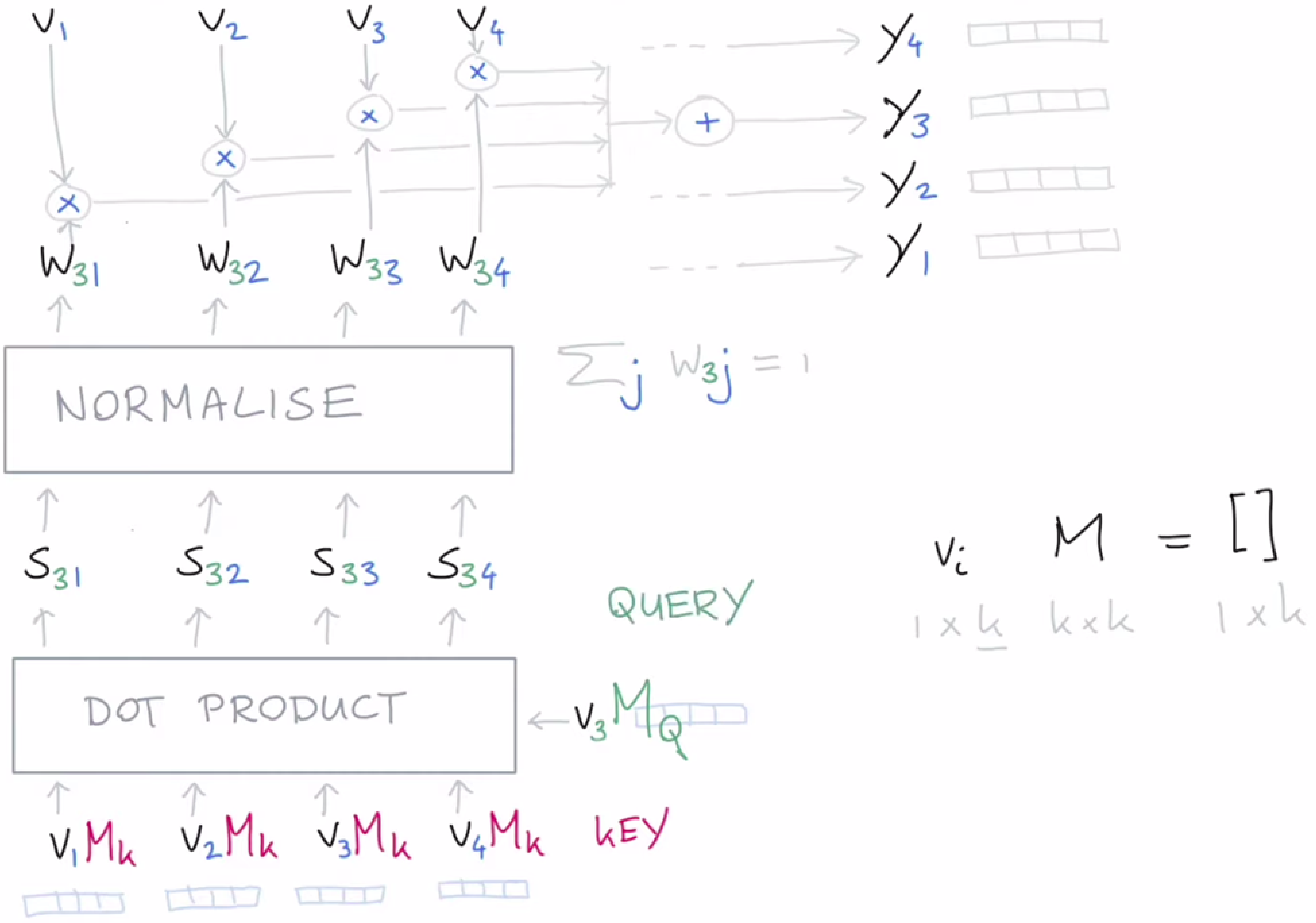
v1 v2 v3 v4
y4
y3
y2
y1
W31 W32 W33 W34
NORMALISE
Σj W3j = 1
vi M = []
1 x k k x k 1 x k
S31 S32 S33 S34
DOT PRODUCT
v3
v1 v2 v3 v4



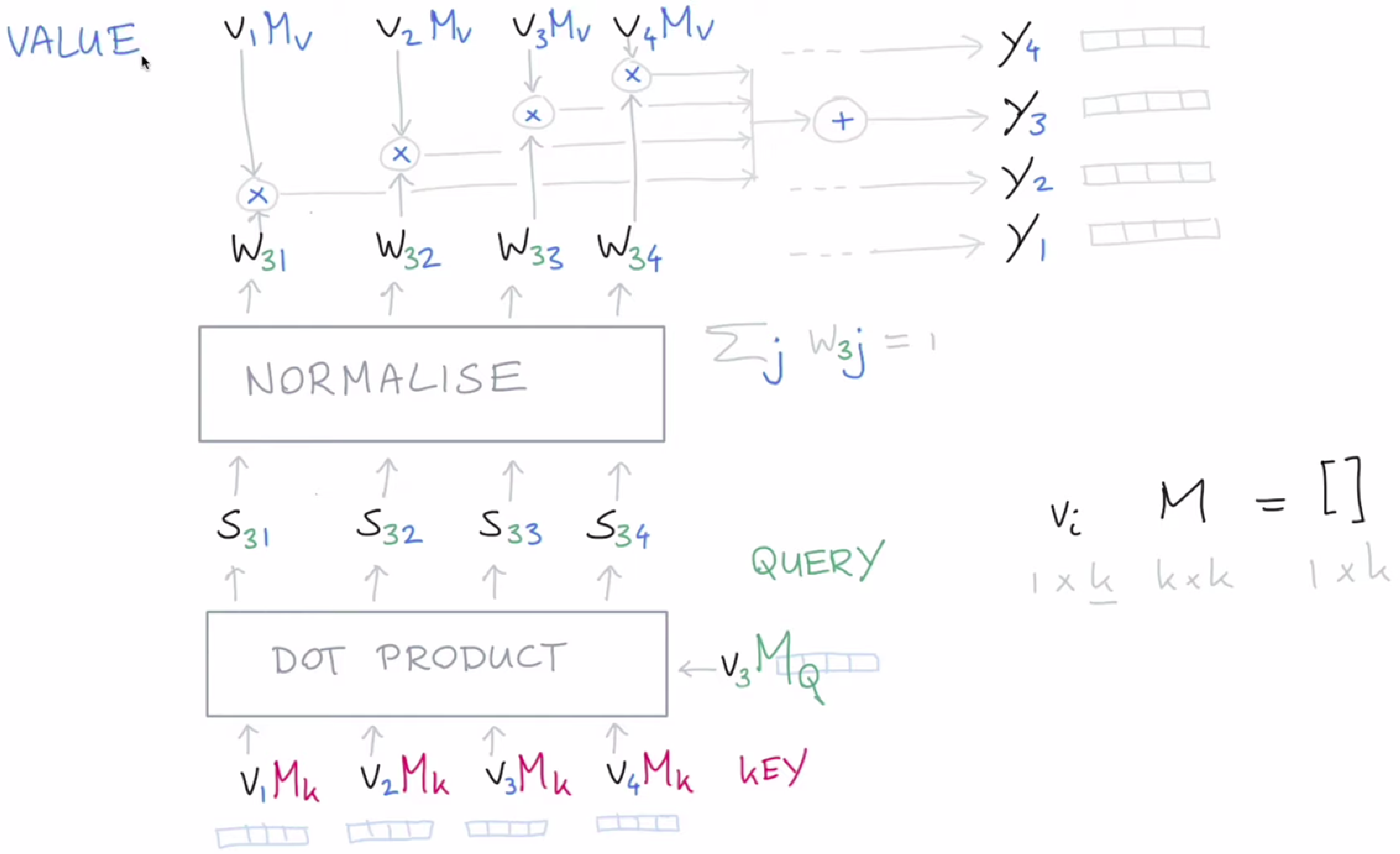
v1 v2 v3 v4
y4
y3
y2
y1
W31 W32 W33 W34
NORMALISE
Σj W3j = 1
vi M = [ ]
1×k k×k 1×k
S31 S32 S33 S34
DOT PRODUCT
v3
v1Mk v2Mk v3Mk v4Mk KEY

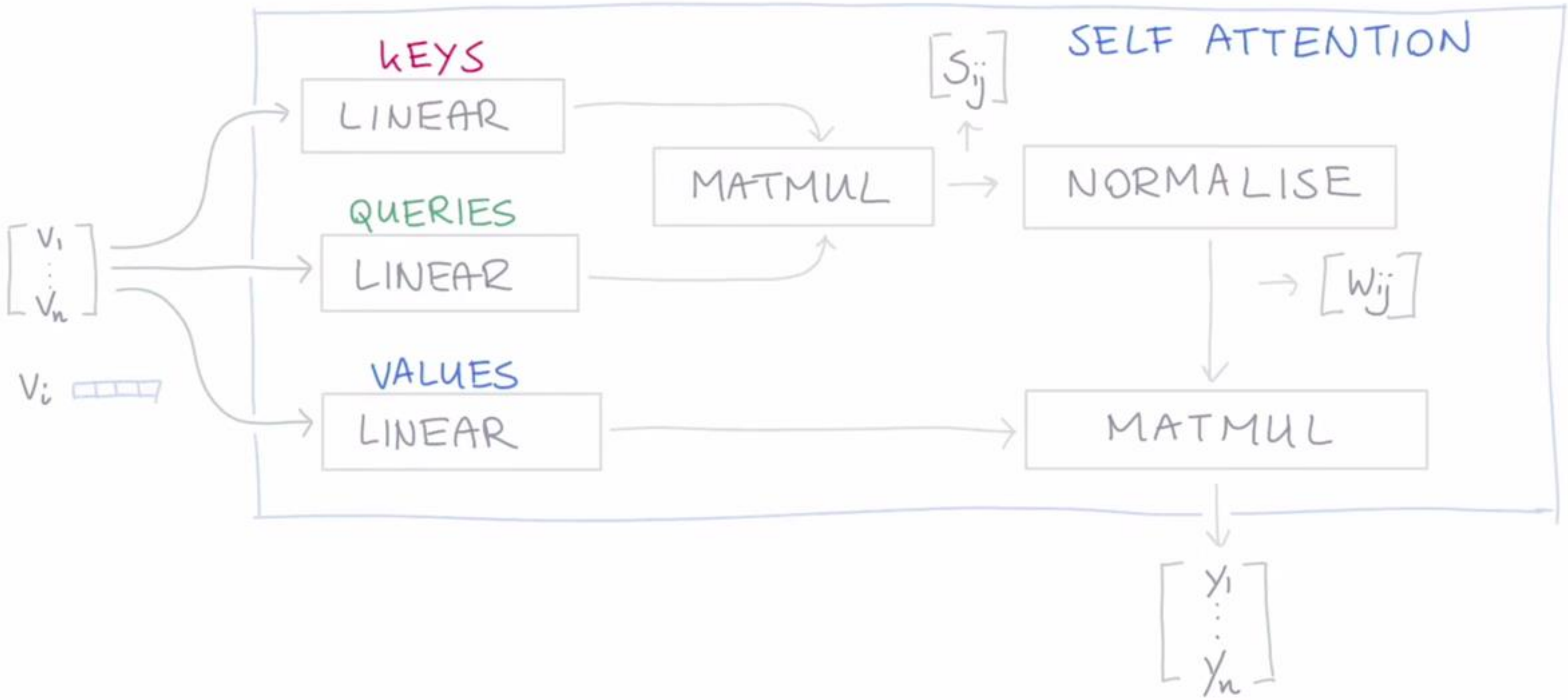


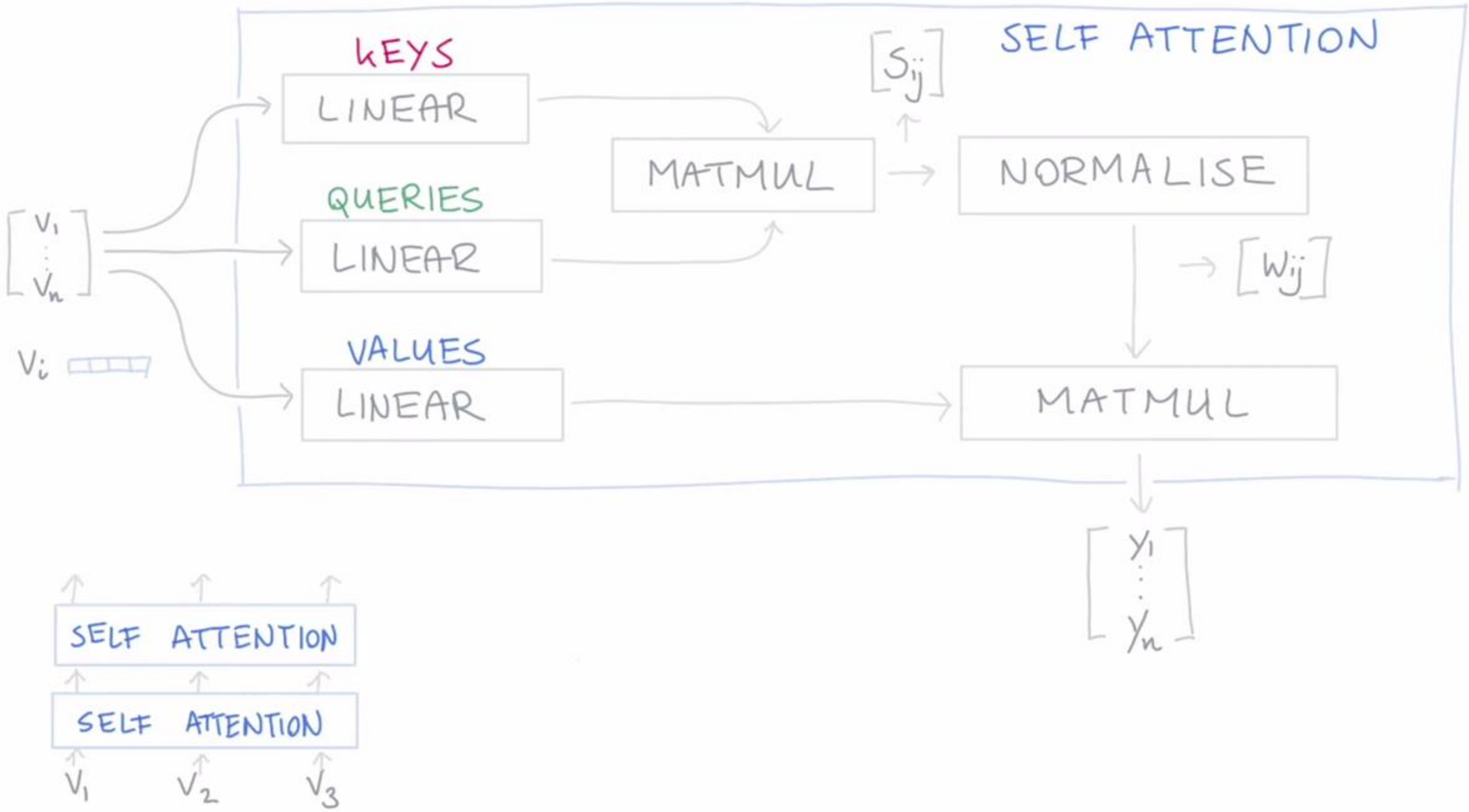
v_1 v_2 v_3 v_4
y_4
y_3
y_2
y_1
w_{31} w_{32} w_{33} w_{34}
NORMALISE
$\Sigma_j w_{3j} = 1$
v_i M $=$ $[\,]$
$1 \times k$ $k \times k$ $1 \times k$
s_{31} s_{32} s_{33} s_{34}
DOT PRODUCT
v_3
$v_1 M_k$ $v_2 M_k$ $v_3 M_k$ $v_4 M_k$
KEY

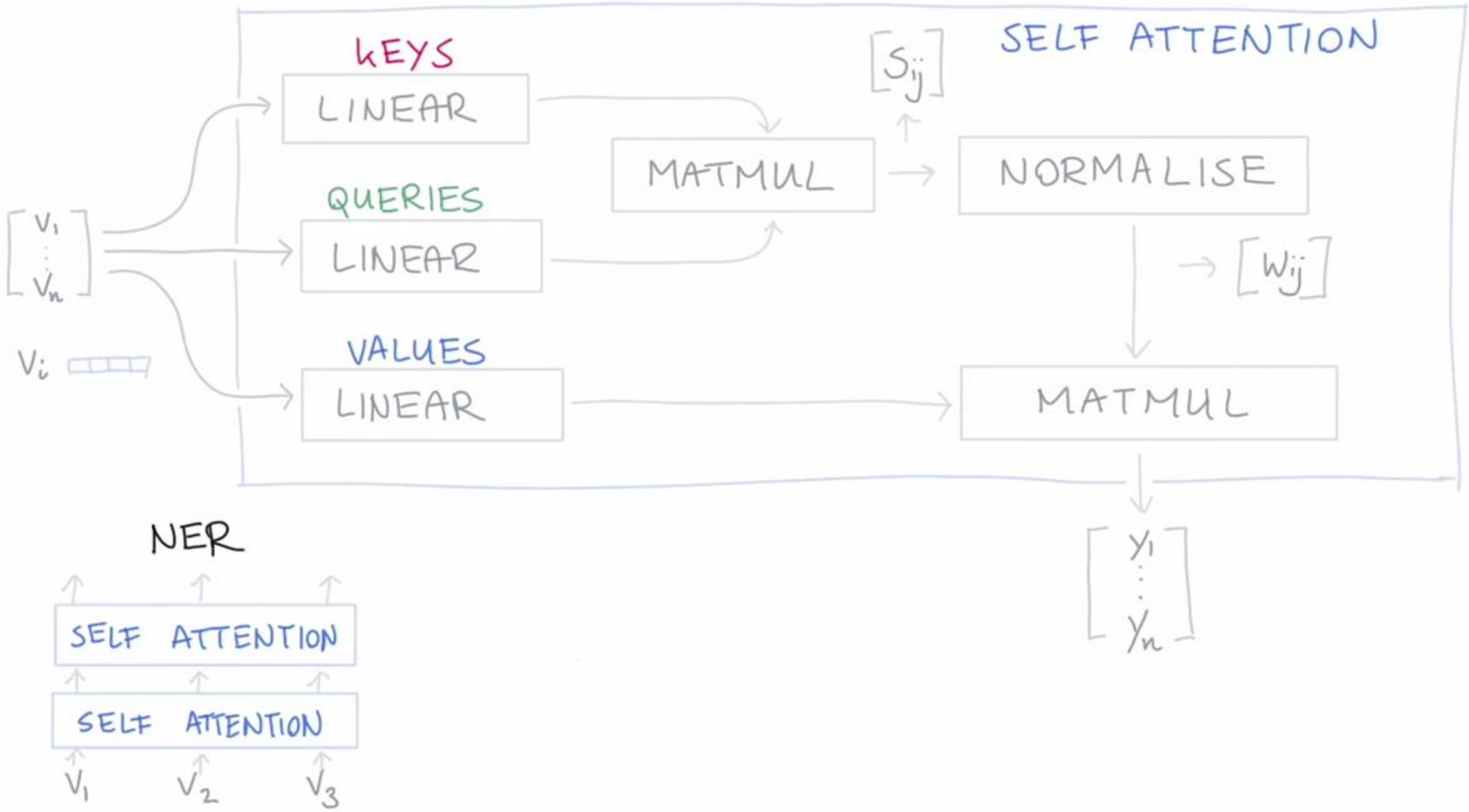


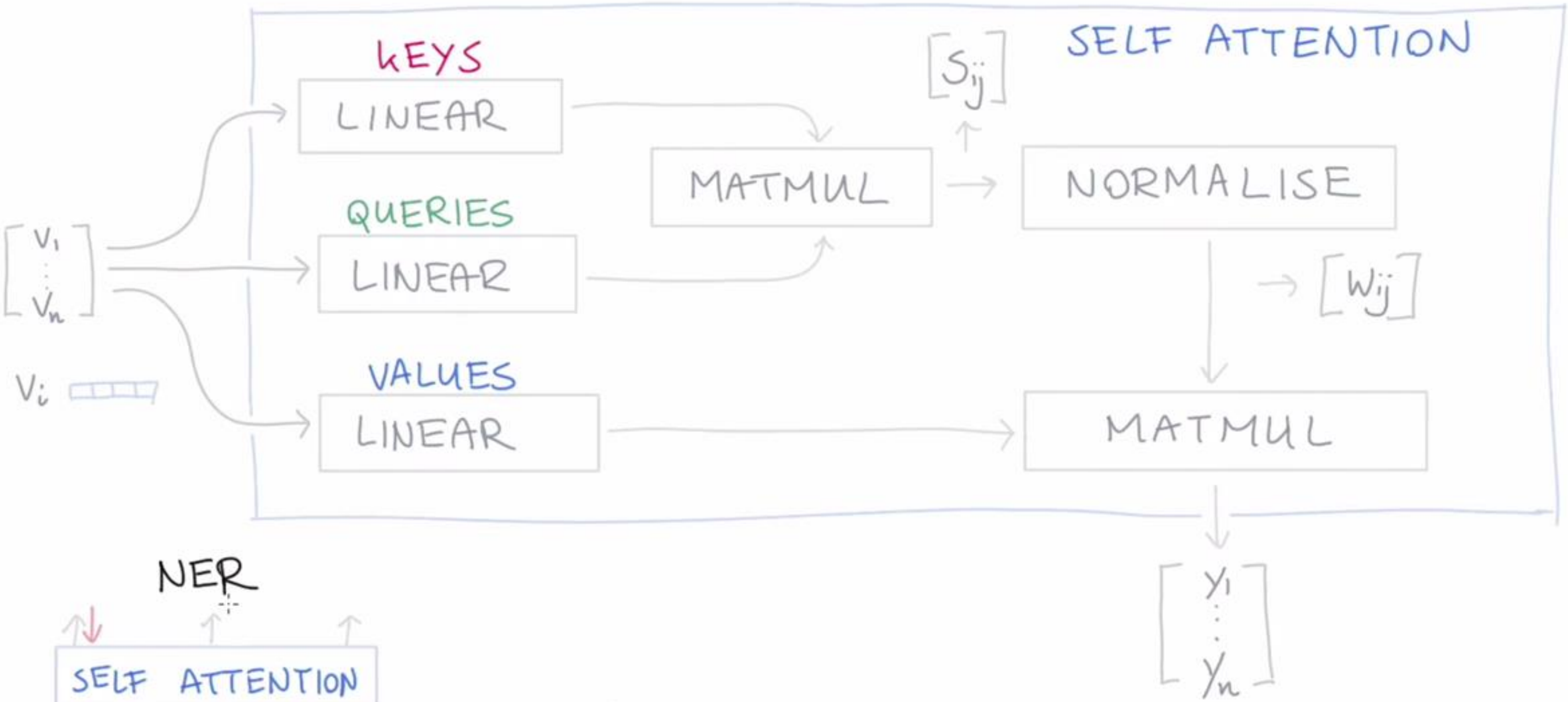
v_1 v_2 v_3 v_4
y_4 y_3 y_2 y_1
w_{31} w_{32} w_{33} w_{34}
NORMALISE
$\sum_j w_{3j} = 1$
$v_i \; M = [\;]$
$1 \times k \quad k \times k \quad 1 \times k$
s_{31} s_{32} s_{33} s_{34}
QUERY
DOT PRODUCT
$v_3 M_Q$
$v_1 M_k$ $v_2 M_k$ $v_3 M_k$ $v_4 M_k$
KEY


VALUE
v1Mv
v2Mv
v3Mv
v4Mv
y4
y3
y2
y1
+
×
W31
W32
W33
W34
Σj W3j = 1
NORMALISE
S31
S32
S33
S34
QUERY
vi M = [ ]
1 x k
k x k
1 x k
DOT PRODUCT
v3MQ
v1Mk
v2Mk
v3Mk
v4Mk
KEY


SELF ATTENTION
KEYS
LINEAR
QUERIES
LINEAR
VALUES
LINEAR
MATMUL
$[S_{ij}]$
NORMALISE
$[W_{ij}]$
MATMUL
v_1
v_n
v_i
y_1
y_n


KEYS
LINEAR
QUERIES
LINEAR
VALUES
LINEAR
MATMUL
$[S_{ij}]$
SELF ATTENTION
NORMALISE
$[W_{ij}]$
MATMUL
$[V_1 \dots V_n]$
V_i
$[y_1 \dots y_n]$
SELF ATTENTION
SELF ATTENTION
V_1 V_2 V_3

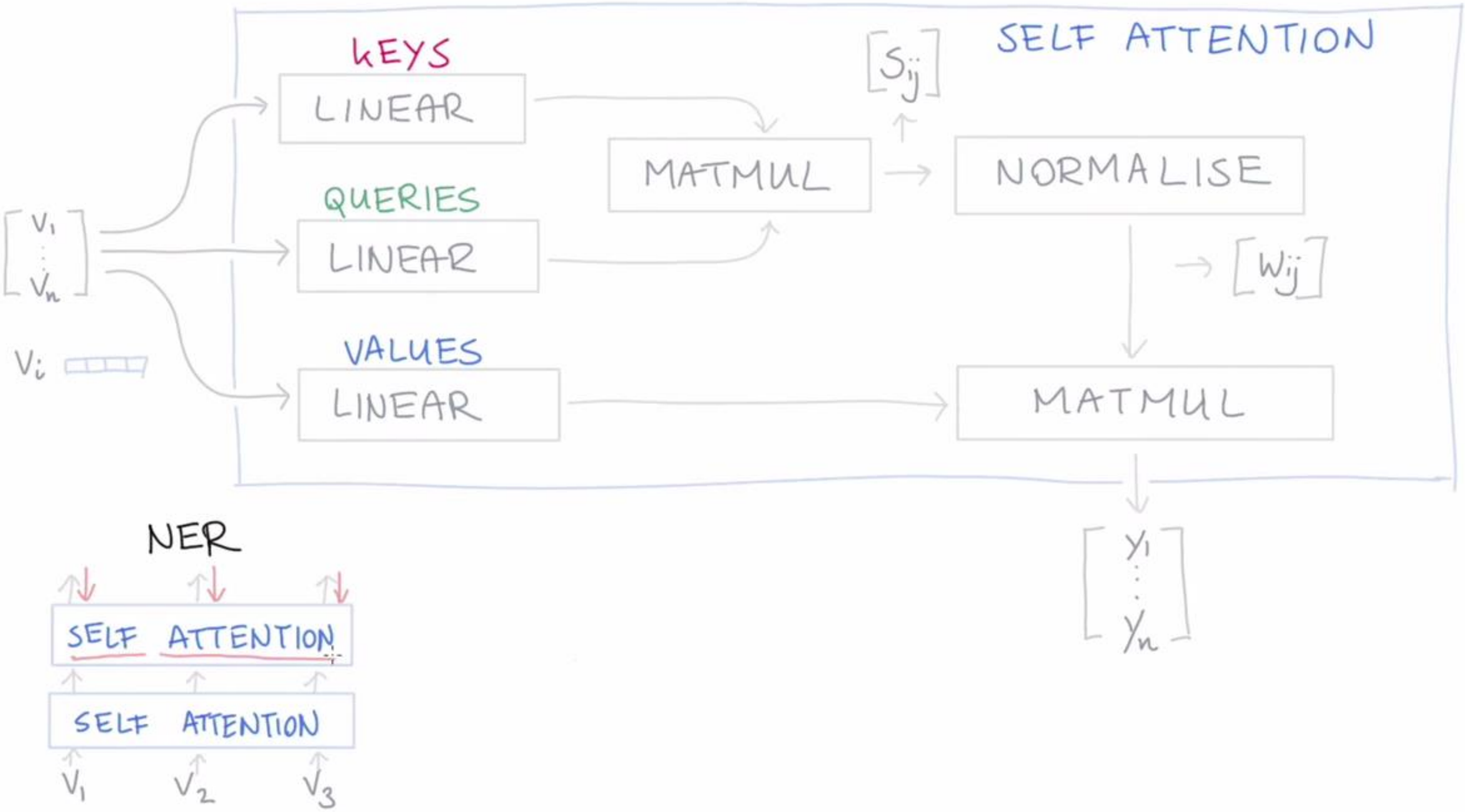

SELF ATTENTION
KEYS
LINEAR
QUERIES
LINEAR
VALUES
LINEAR
MATMUL
$[S_{ij}]$
NORMALISE
$[W_{ij}]$
MATMUL
V_1
V_n
V_i
y_1
y_n
NER
SELF ATTENTION
SELF ATTENTION
V_1
V_2
V_3

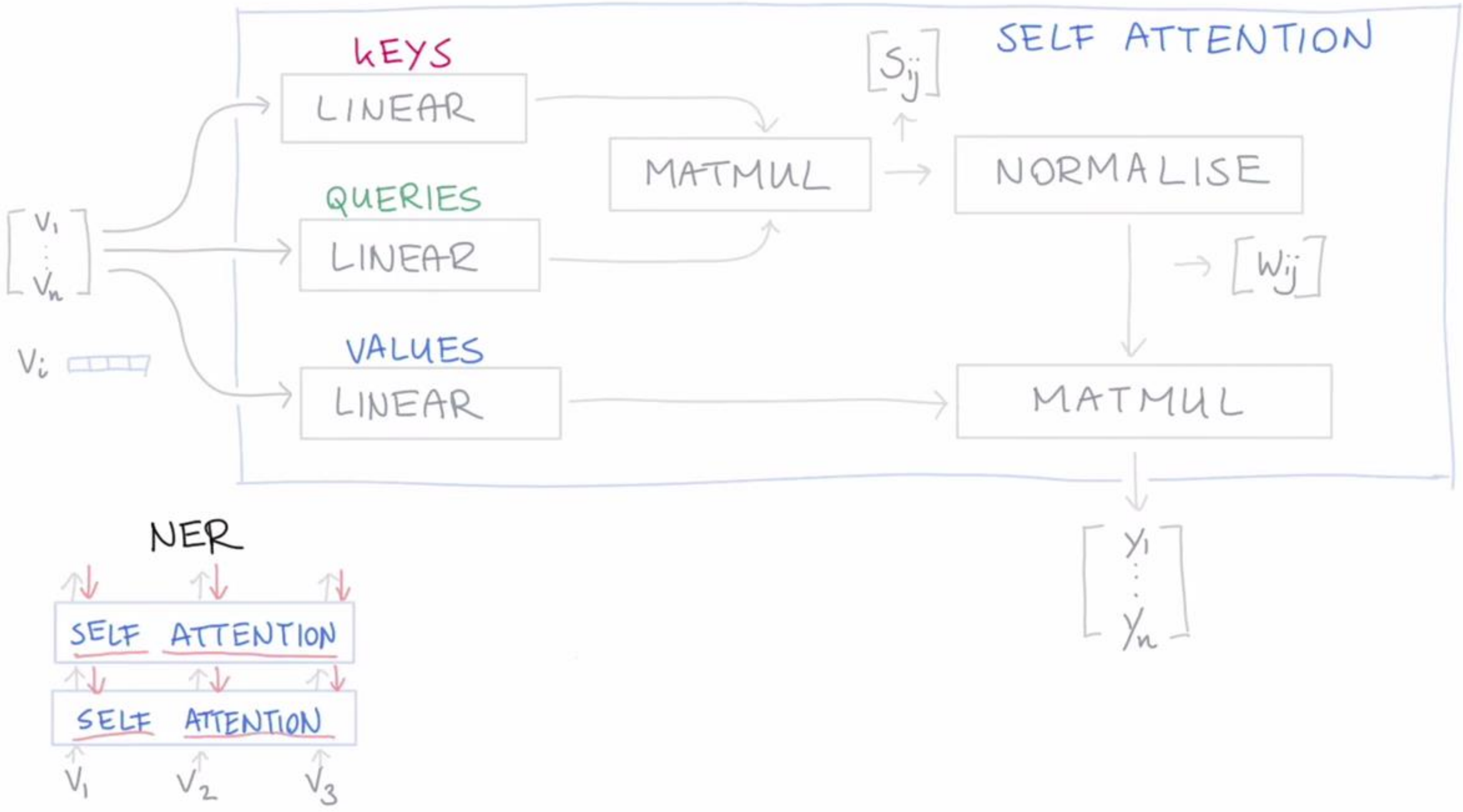


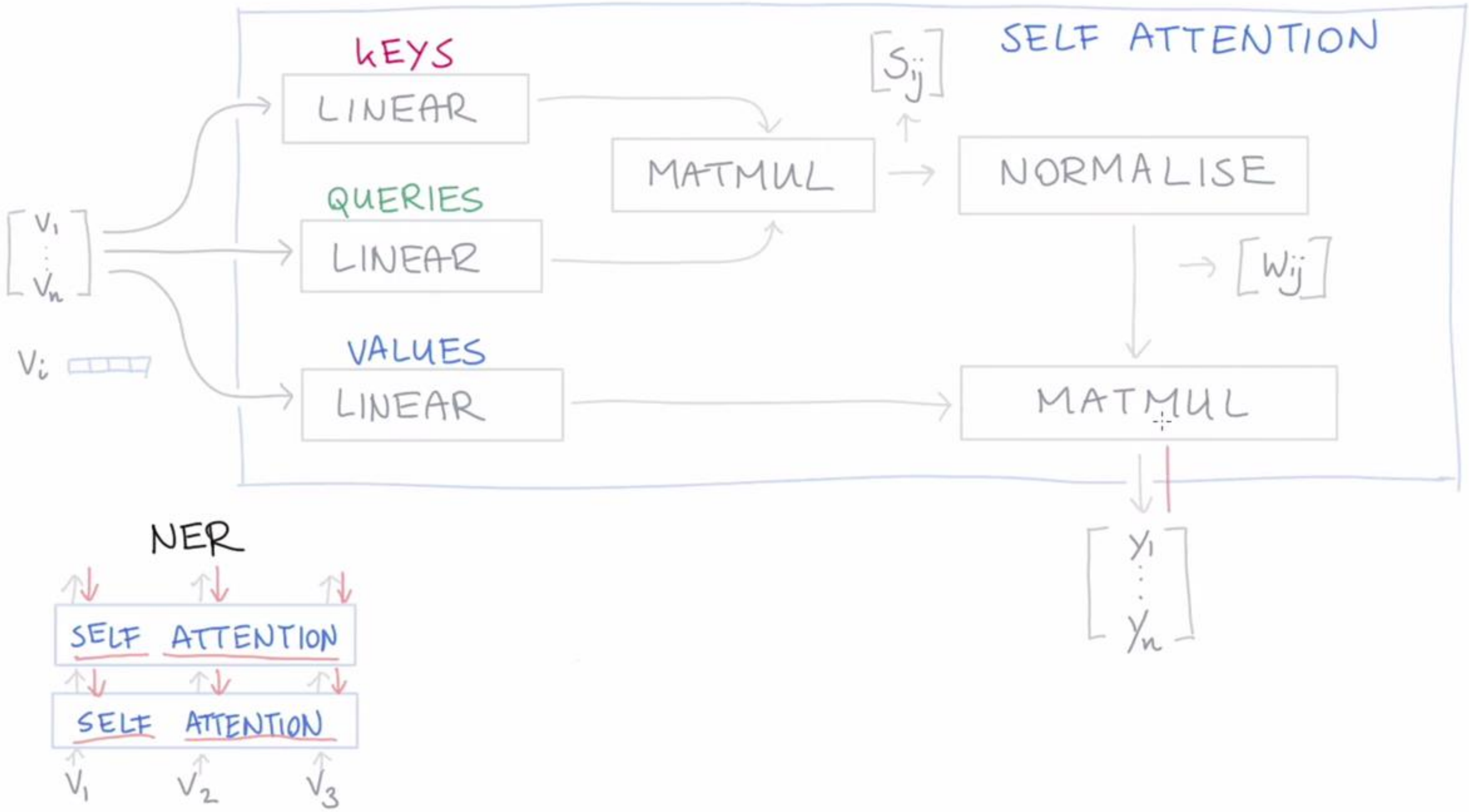
SELF ATTENTION
KEYS
LINEAR
QUERIES
LINEAR
VALUES
LINEAR
MATMUL
$[S_{ij}]$
NORMALISE
$[W_{ij}]$
MATMUL
V_1
V_n
V_i
y_1
y_n

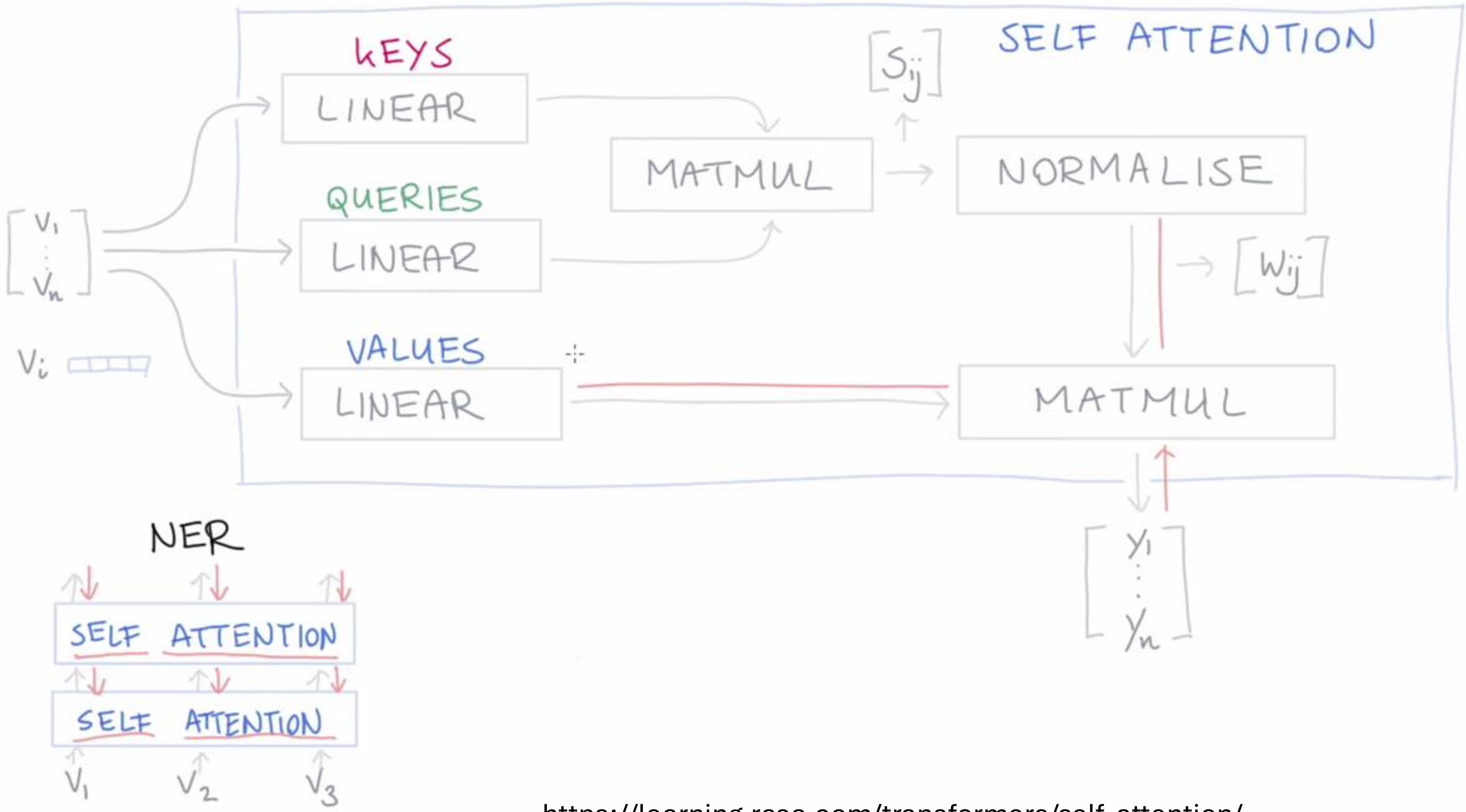
NER

SELF ATTENTION
SELF ATTENTION
V_1
V_2
V_3

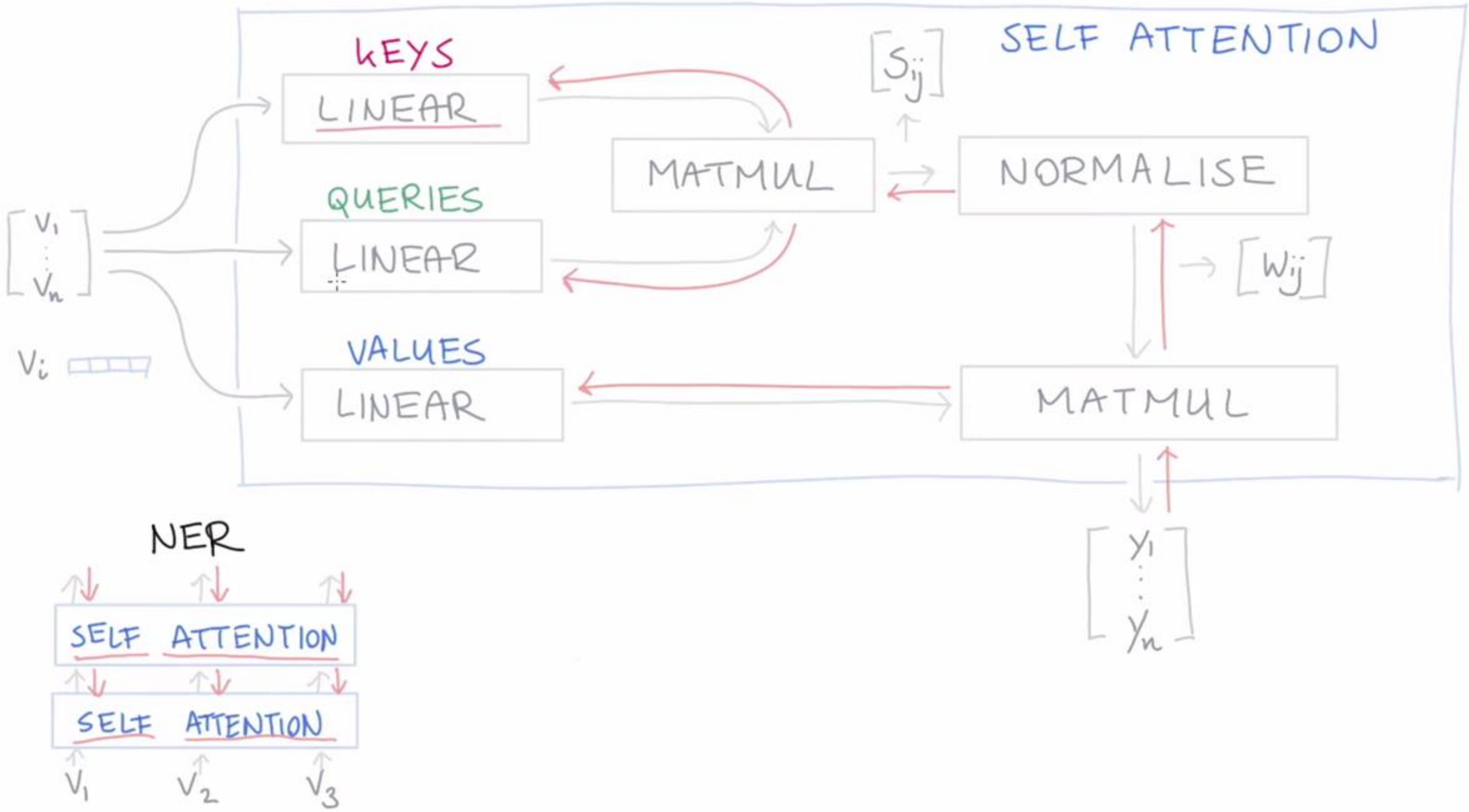

SELF ATTENTION
KEYS
LINEAR
QUERIES
LINEAR
VALUES
LINEAR
MATMUL
NORMALISE
MATMUL
$[S_{ij}]$
$[W_{ij}]$
$V_1 \ldots V_n$
V_i
$y_1 \ldots y_n$
NER
SELF ATTENTION
SELF ATTENTION
V_1 V_2 V_3

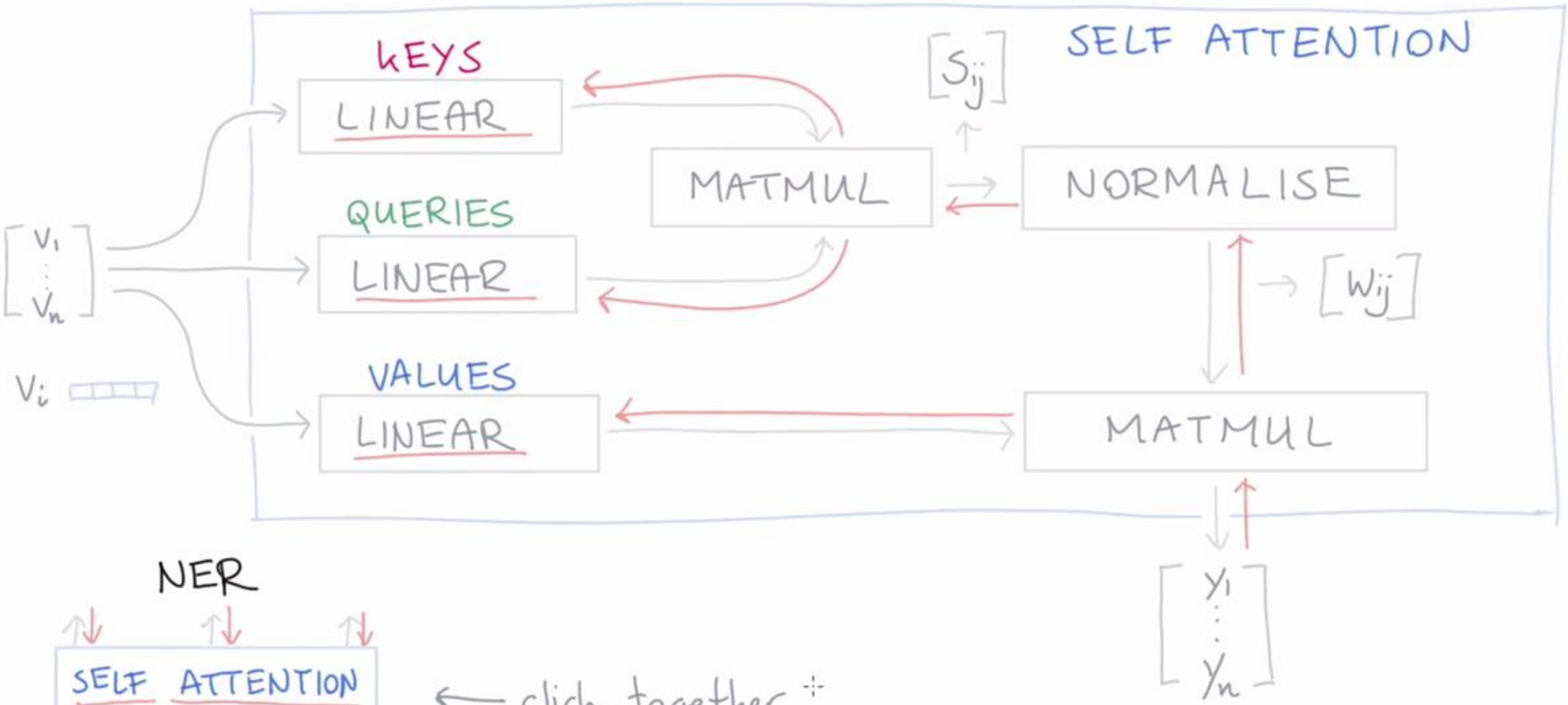

SELF ATTENTION
KEYS
LINEAR
QUERIES
LINEAR
VALUES
LINEAR
MATMUL
NORMALISE
MATMUL
$[S_{ij}]$
$[W_{ij}]$
$V_1 \dots V_n$
V_i
$y_1 \dots y_n$
NER
SELF ATTENTION
SELF ATTENTION
V_1 V_2 V_3


SELF ATTENTION
KEYS
LINEAR
QUERIES
LINEAR
VALUES
LINEAR
MATMUL
NORMALISE
MATMUL
$[S_{ij}]$
$[W_{ij}]$
V_1 … V_n
V_i
y_1 … y_n
NER
SELF ATTENTION
SELF ATTENTION
V_1 V_2 V_3



https://learning.rasa.com/transformers/self-attention/


SELF ATTENTION
KEYS
LINEAR
QUERIES
LINEAR
VALUES
LINEAR
MATMUL
NORMALISE
MATMUL
$[S_{ij}]$
$[W_{ij}]$
V_1 ⋮ V_n
V_i
y_1 ⋮ y_n
NER
SELF ATTENTION
SELF ATTENTION
V_1
V_2
V_3



NER

← click together

Acknowledgments

These slides were adapted from the book

SPEECH and LANGUAGE PROCESSING: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition

and

some modifications from presentations and resources found in the WEB by several scholars mentioned in references.

References

- https://slds-lmu.github.io/seminar_nlp_ss20/attention-and-self-attention-for-nlp.html
- Attention? Attention! | Lil'Log (lilianweng.github.io)

Reference materials

- https://vlanc-lab.github.io/mu-nlp-course/teachings/fall2024-AI-schedule.html

- Lecture notes
- (A) Speech and Language Processing by Daniel Jurafsky and James H. Martin
- (B) Natural Language Processing with Python. (updated edition based on Python 3 and NLTK 3) Steven Bird et al. O'Reilly Media