Natural Language Processing

CS 3216/UG, AI 5203/PG

Week-8

Encoder-decoder models, Attention mechanism

Mahindra University

Global Thinkers. Engaged Leaders.

Recap

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

History of Neural models in NLP

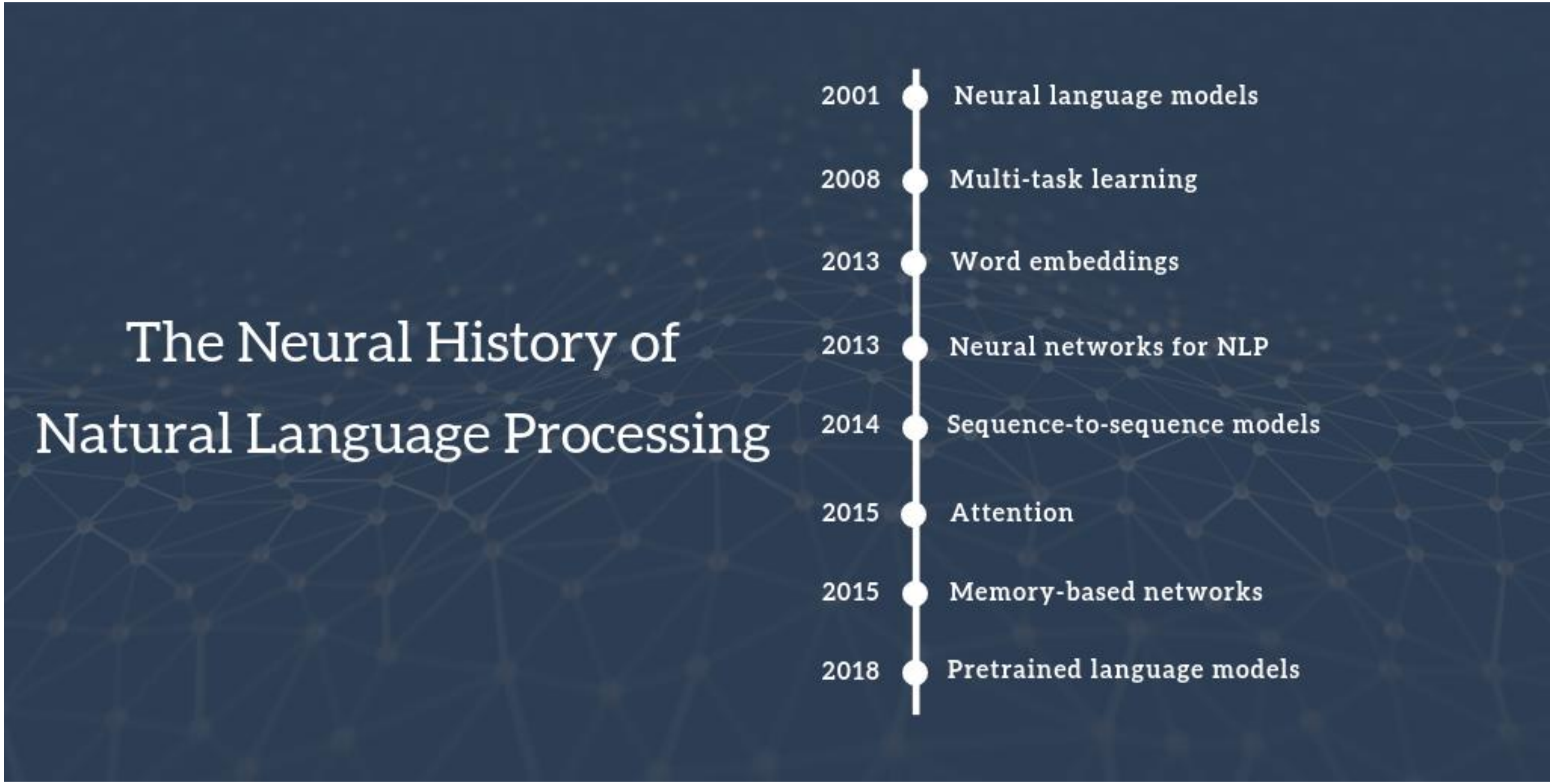


Image source : https://www.ruder.io/a-review-of-the-recent-history-of-nlp/

Different variants of RNN

- Stacked RNN
- Bi-directional RNN
- Many more

Sequence -to-Sequence learning

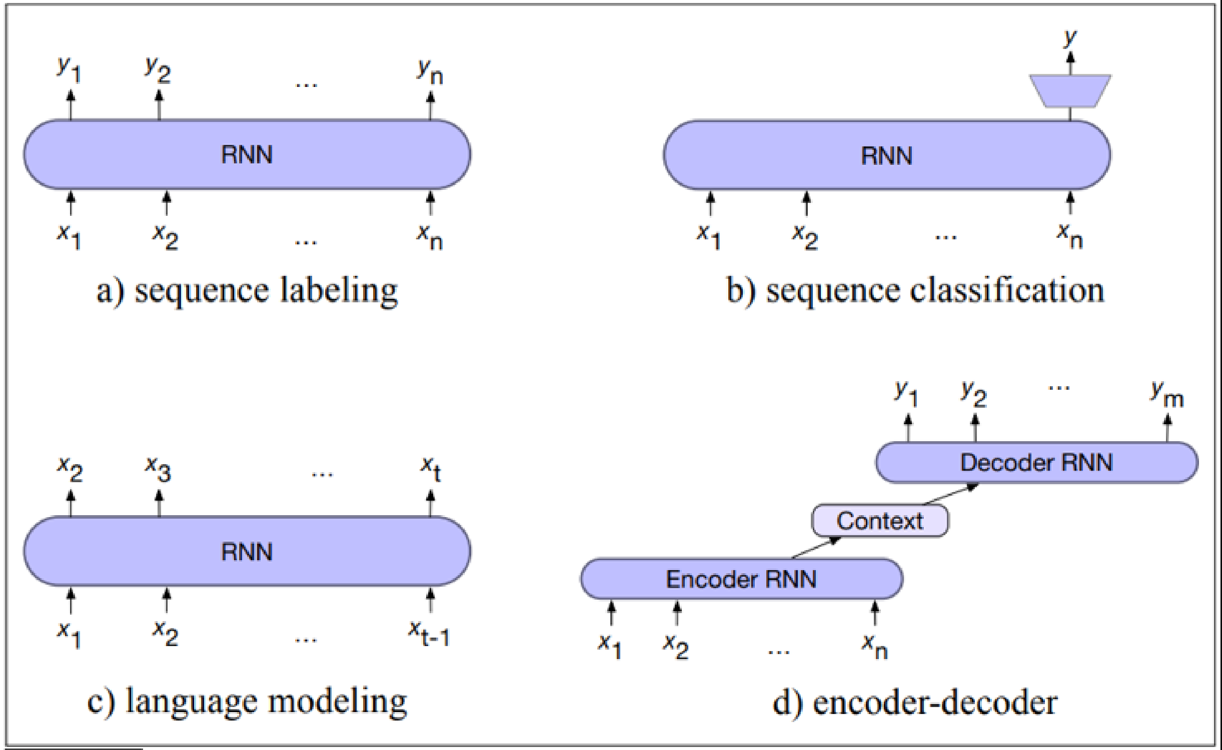


Figure 9.15 Four architectures for NLP tasks. In sequence labeling (POS or named entity tagging) we map each input token x_i to an output token y_i. In sequence classification we map the entire input sequence to a single class. In language modeling we output the next token conditioned on previous tokens. In the encoder model we have two separate RNN models, one of which maps from an input sequence $\mathbf{x}$ to an intermediate representation we call the **context**, and a second of which maps from the context to an output sequence $\mathbf{y}$.

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

Sequence to Sequence Learning with Neural Networks

Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

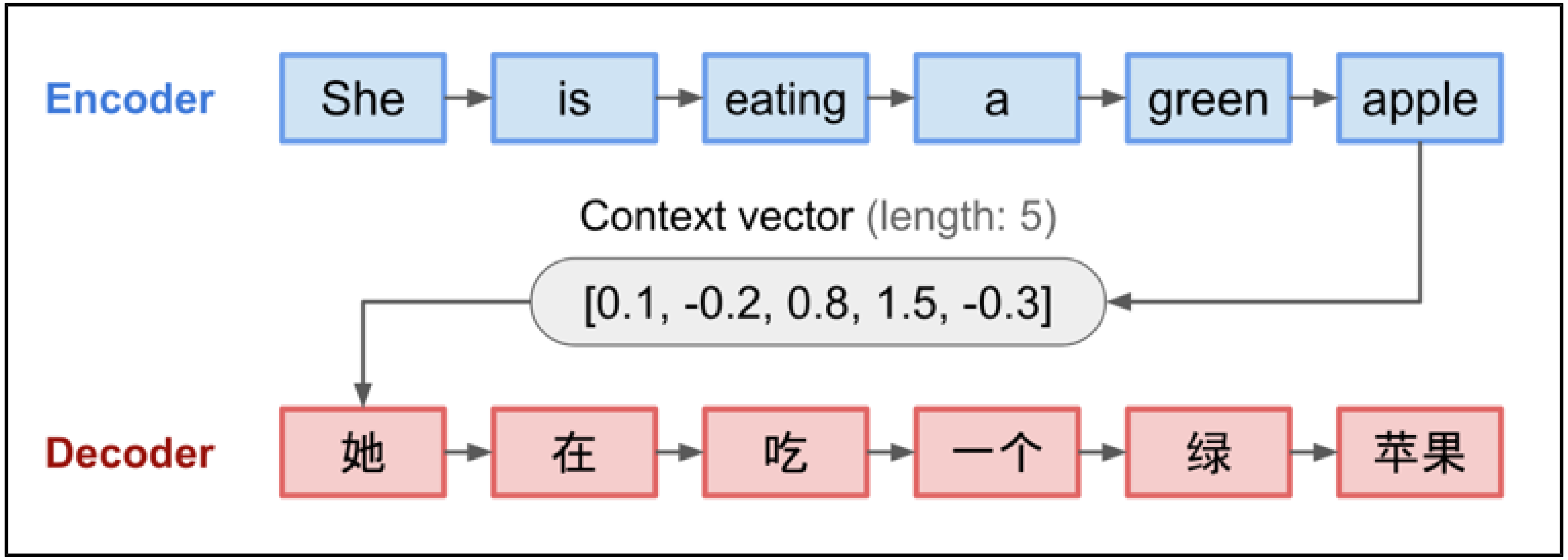
Quoc V. Le
Google
qvl@google.com

Abstract

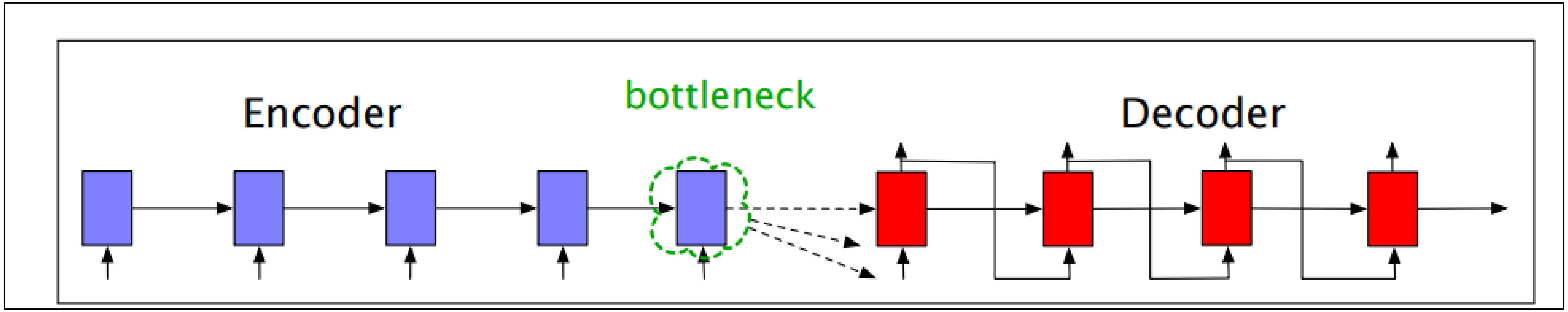
Deep Neural Networks (DNNs) are powerful models that have achieved excellent performance on difficult learning tasks. Although DNNs work well whenever large labeled training sets are available, they cannot be used to map sequences to sequences. In this paper, we present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. Our method uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. Our main result is that on an English to French translation task from the WMT'14 dataset, the translations produced by the LSTM achieve a BLEU score of 34.8 on the entire test set, where the LSTM's BLEU score was penalized on out-of-vocabulary words. Additionally, the LSTM did not have difficulty on long sentences. For comparison, a phrase-based SMT system achieves a BLEU score of 33.3 on the same dataset. When we used the LSTM to rerank the 1000 hypotheses produced by the aforementioned SMT system, its BLEU score increases to 36.5, which is close to the previous best result on this task. The LSTM also learned sensible phrase and sentence representations that are sensitive to word order and are relatively invariant to the active and the pas-

https://arxiv.org/pdf/1409.3215

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.

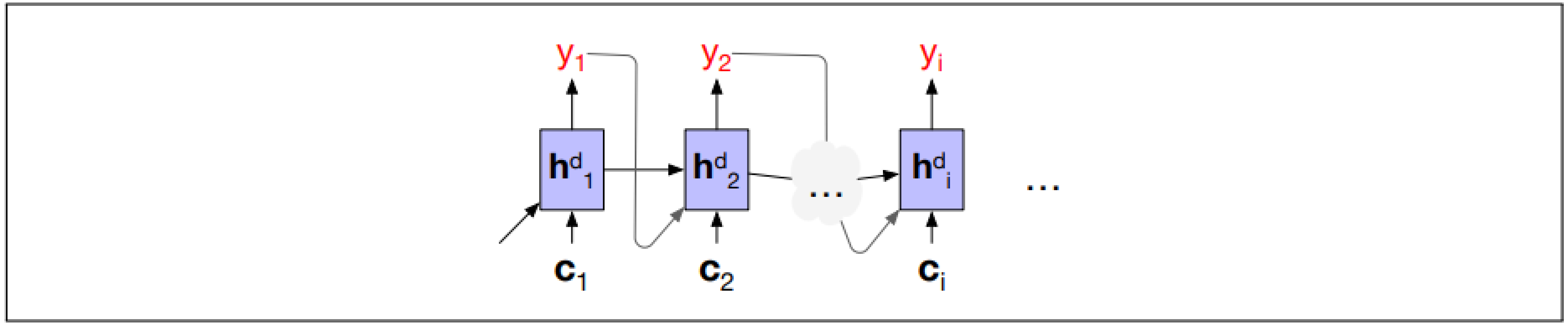


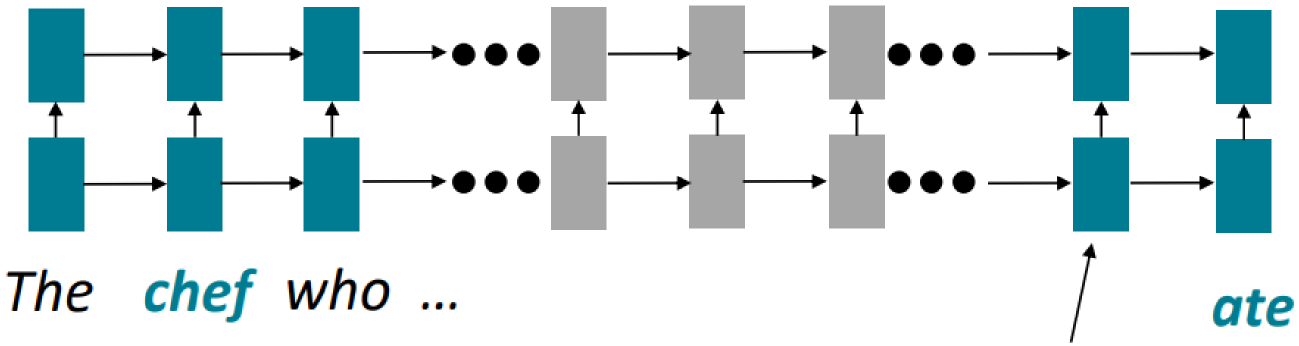
Figure 9.21 The attention mechanism allows each hidden state of the decoder to see a different, dynamic, context, which is a function of all the encoder hidden states.

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...

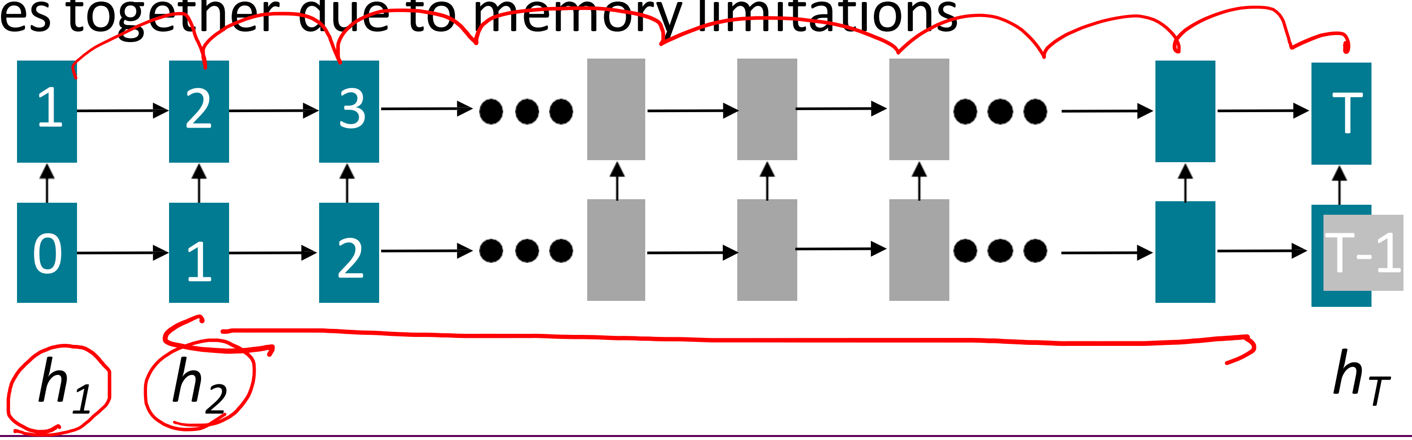


Info of ***chef*** has gone through O(sequence length) many layers!

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

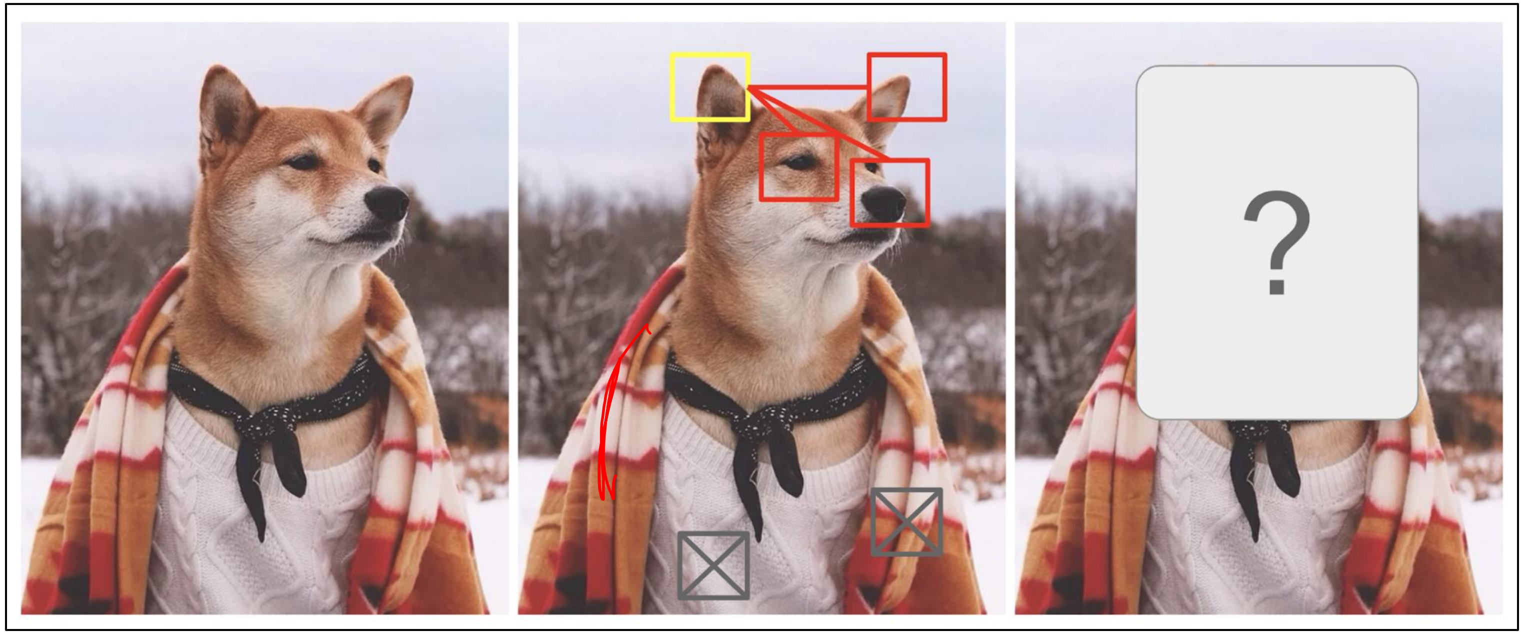


Numbers indicate min # of steps before a state can be computed

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

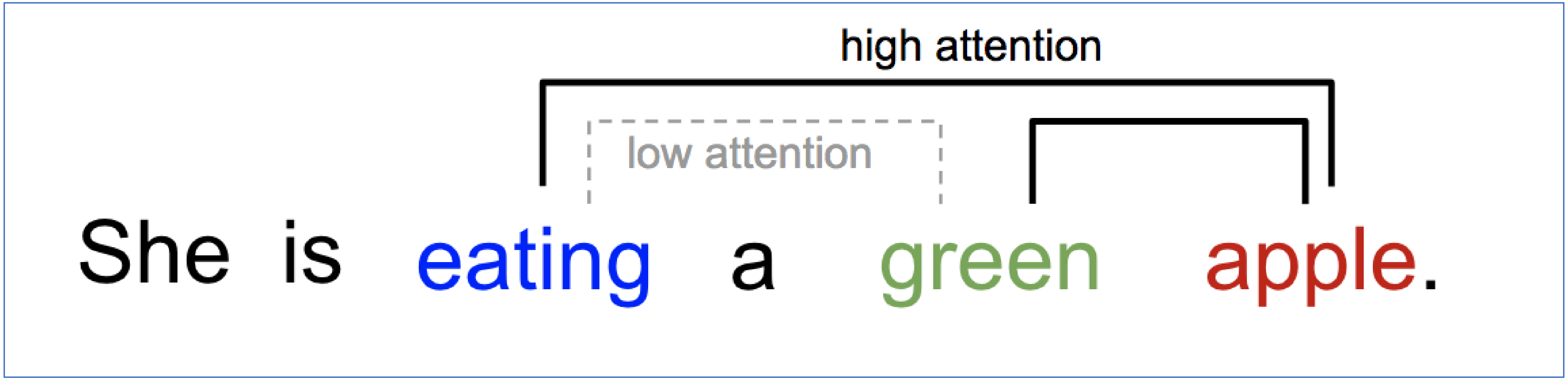
LET'S HAVE
A MOMENT
OF SILENCE
FOR ALL
THOSE
PEOPLE
DYING FOR
ATTENTION

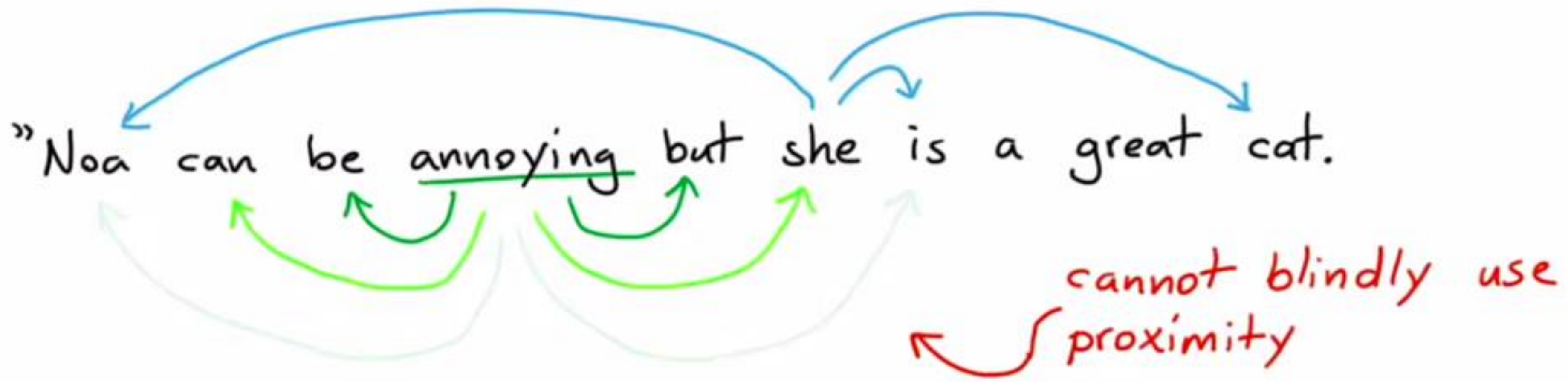
Attention



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





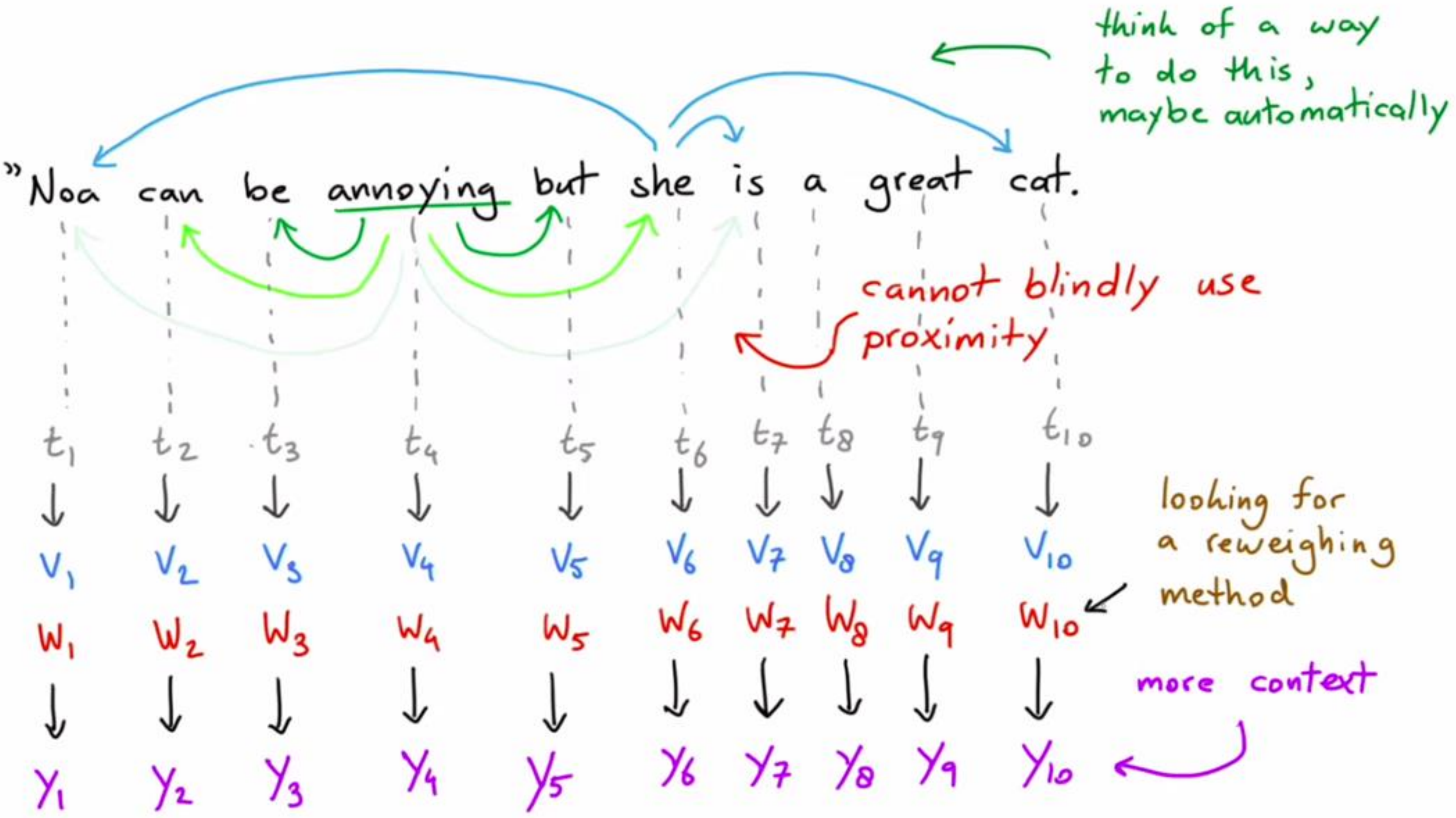
https://learning.rasa.com/transformers/self-attention/

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

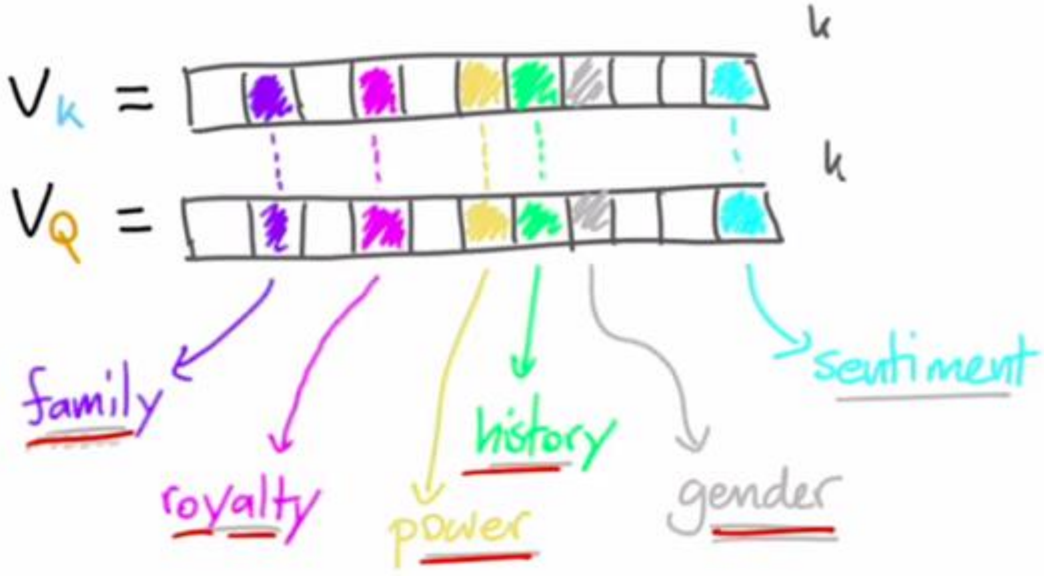
think of a way to do this, maybe automatically

cannot blindly use proximity

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



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



son
daughter
cousin
land
country
army
own

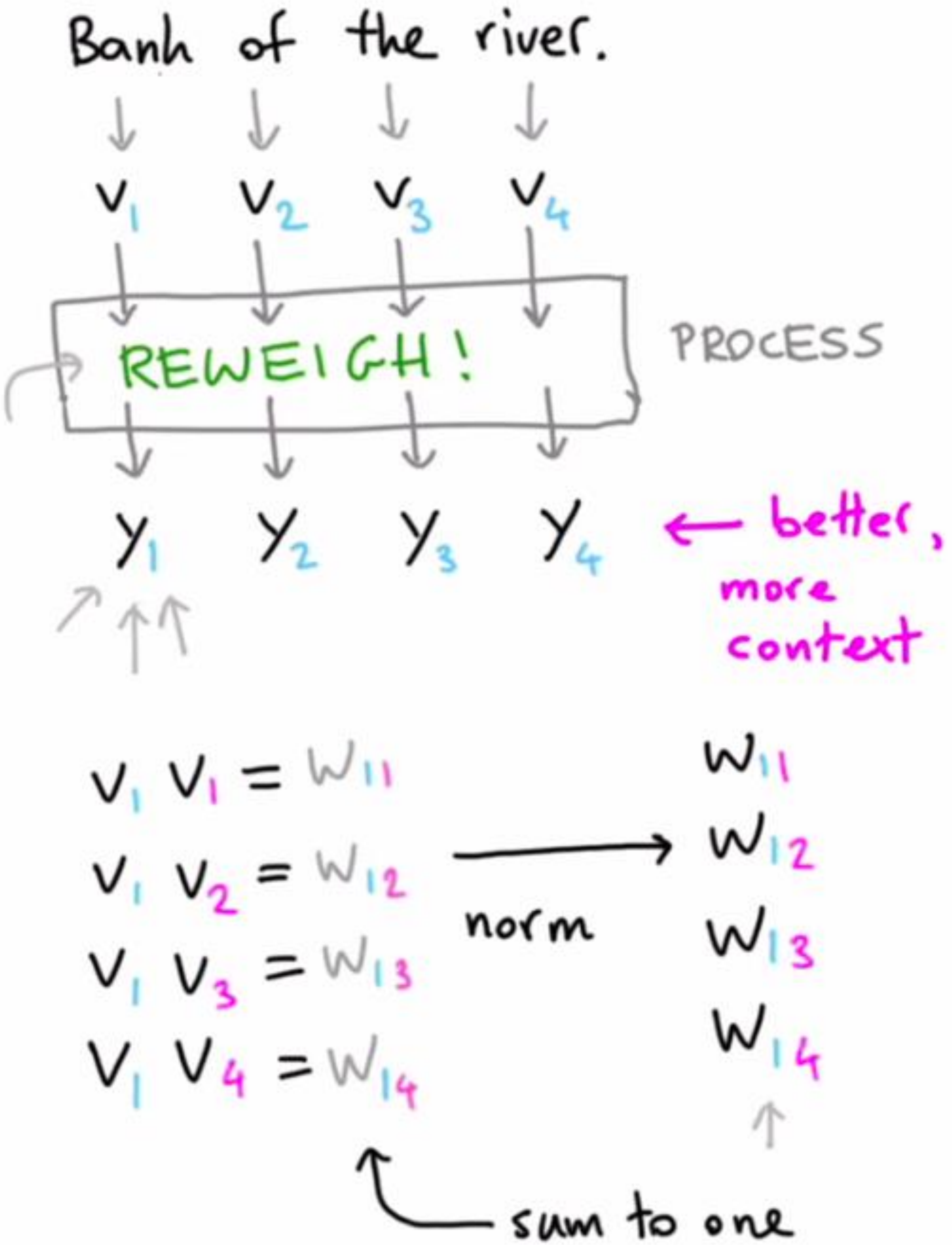
dog
cat
running
swimming
help
nurse
computer

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

Reweighing Plan

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

Let's explore this



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

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

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

$v_1 \, v_1 = w_{11}$

$v_1 \, v_2 = w_{12}$

$v_1 \, v_3 = w_{13}$

$v_1 \, v_4 = w_{14}$

$\xrightarrow{\text{norm}}$

w_{11}

w_{12}

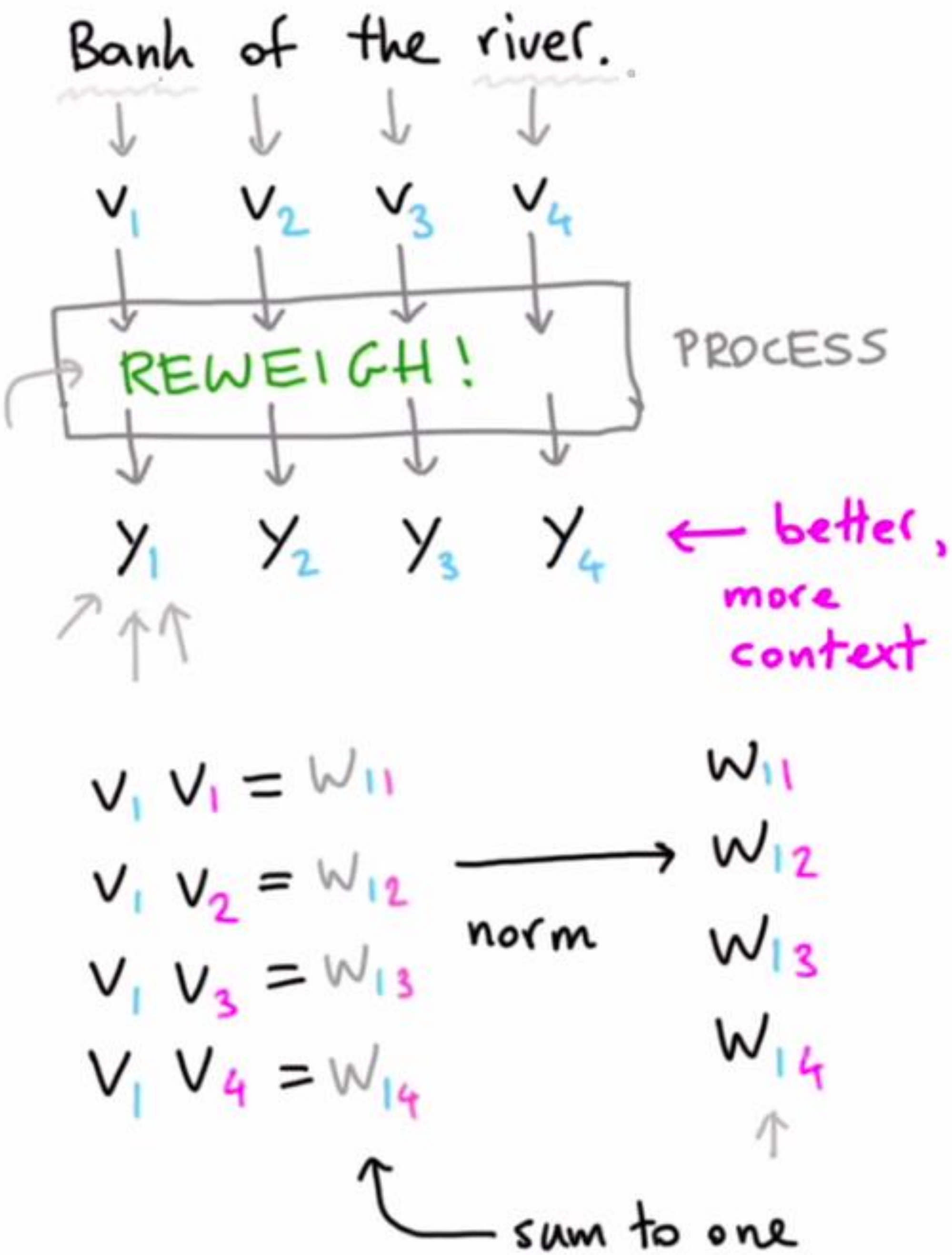
w_{13}

w_{14}

sum to one

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

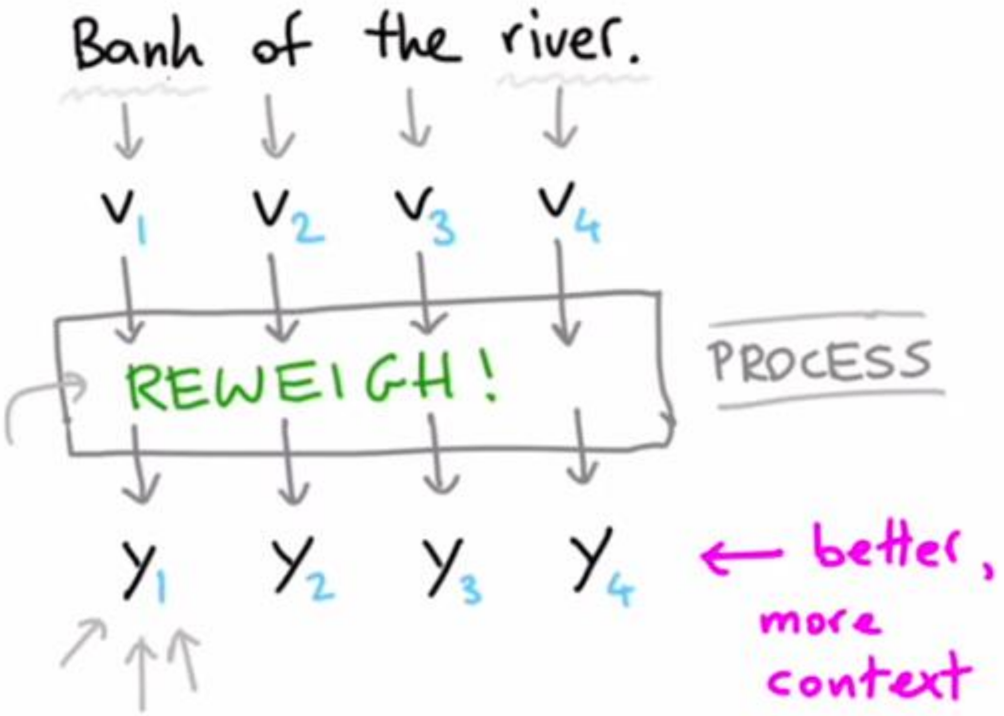
https://learning.rasa.com/transformers/self-attention/



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

reweigh all vectors towards v_1

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



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

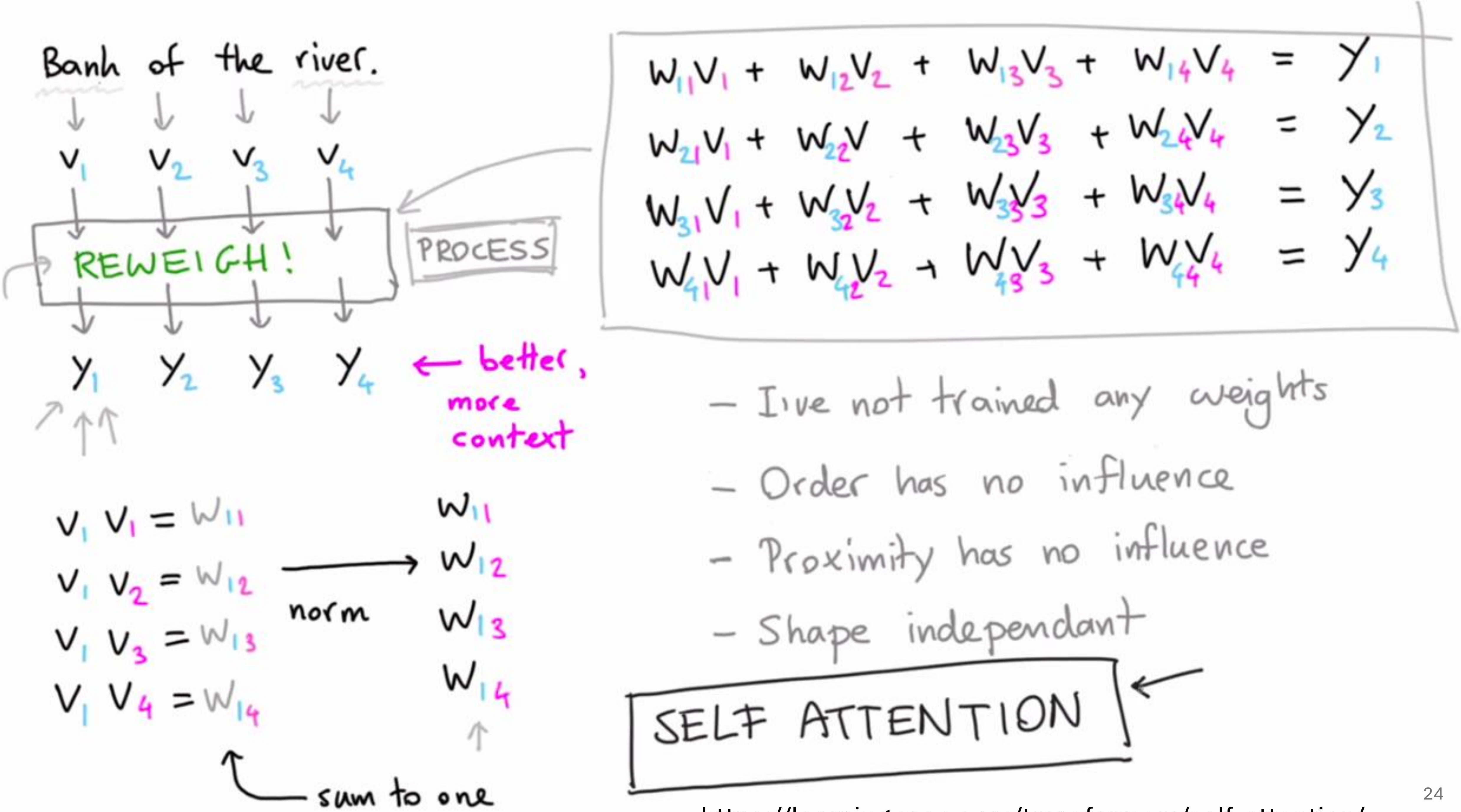
$$v_1 v_1 = w_{11}$$
$$v_1 v_2 = w_{12}$$
$$v_1 v_3 = w_{13}$$
$$v_1 v_4 = w_{14}$$

$\xrightarrow{\text{norm}}$

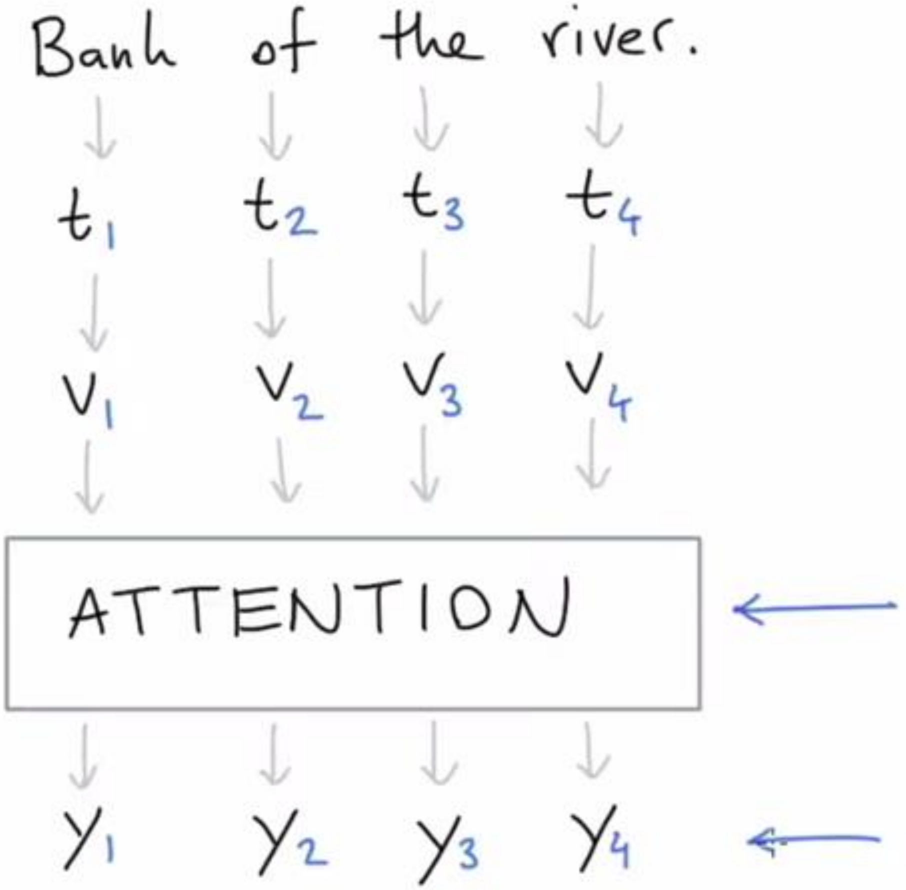
$$w_{11}$$
$$w_{12}$$
$$w_{13}$$
$$w_{14}$$

sum to one

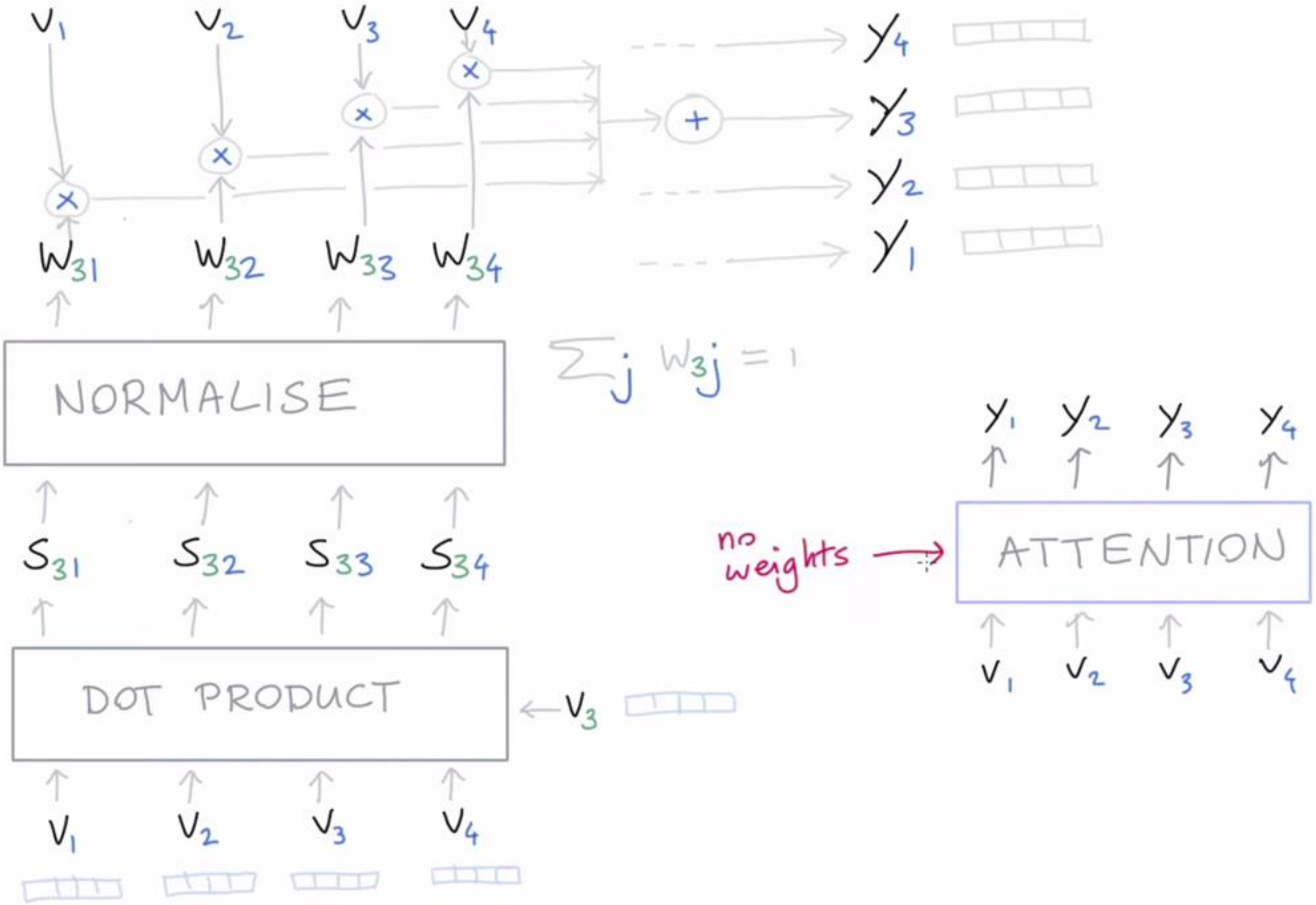
https://learning.rasa.com/transformers/self-attention/



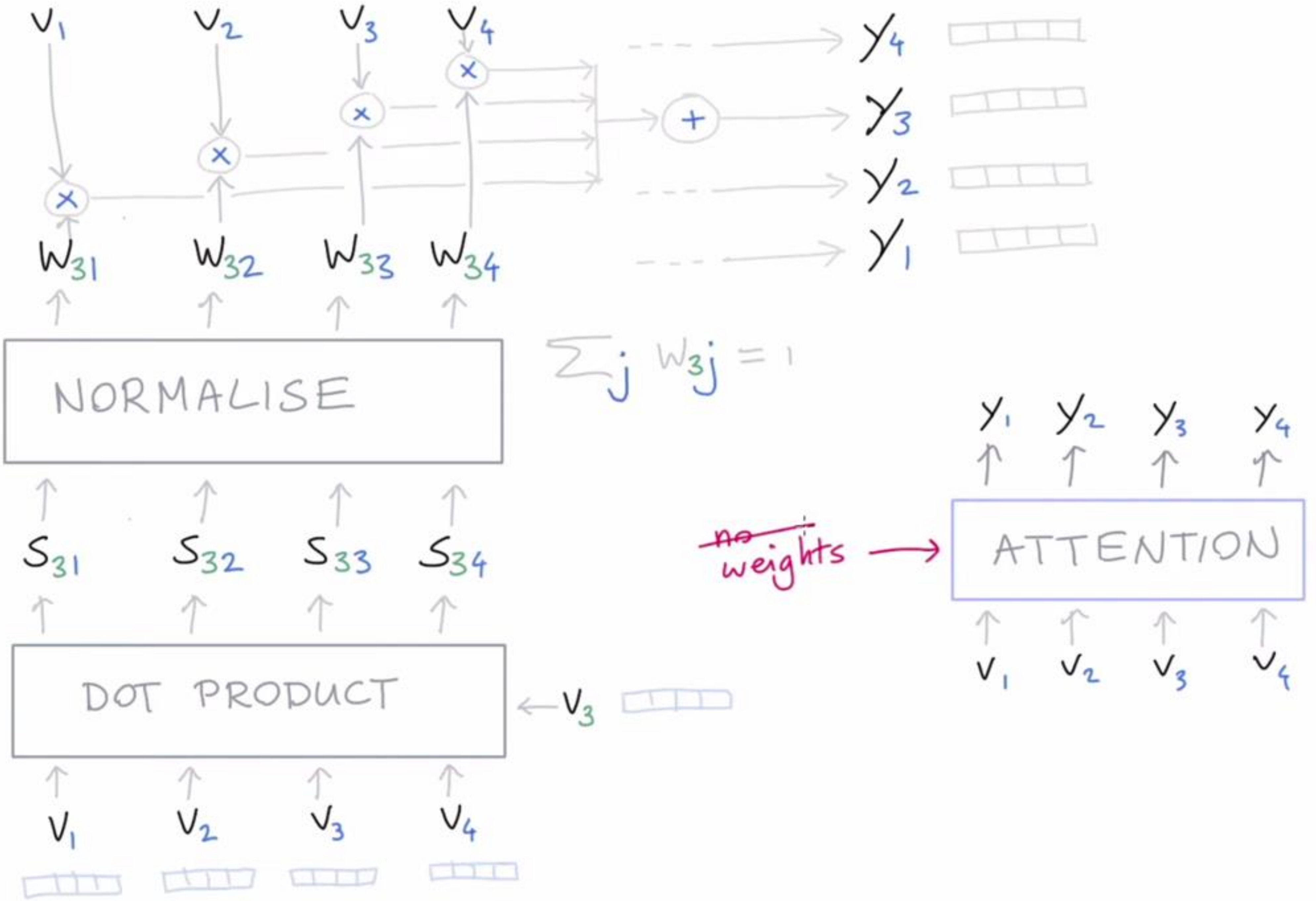
https://learning.rasa.com/transformers/self-attention/



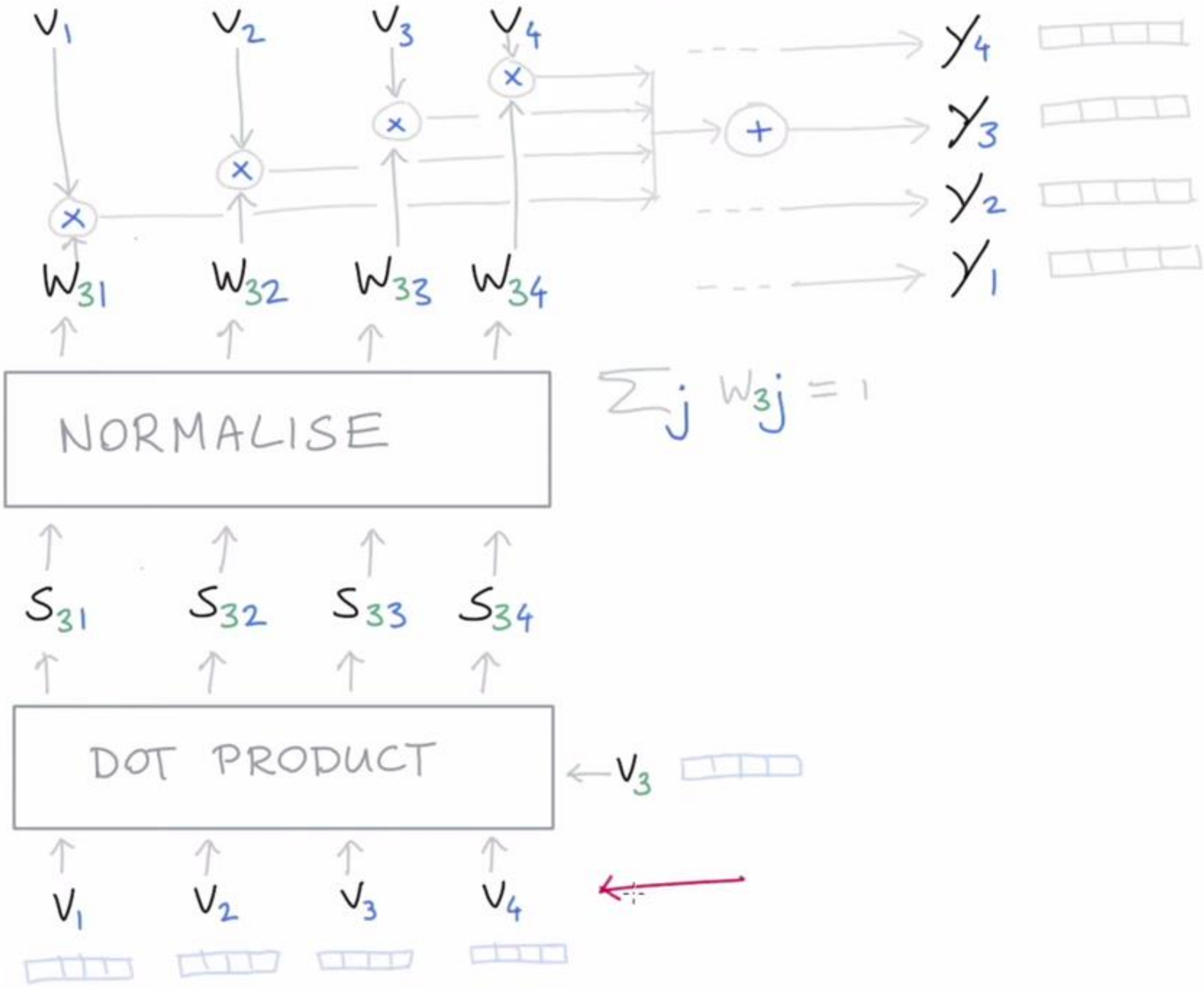
Bank of the river.
t1 t2 t3 t4
v1 v2 v3 v4
ATTENTION
y1 y2 y3 y4



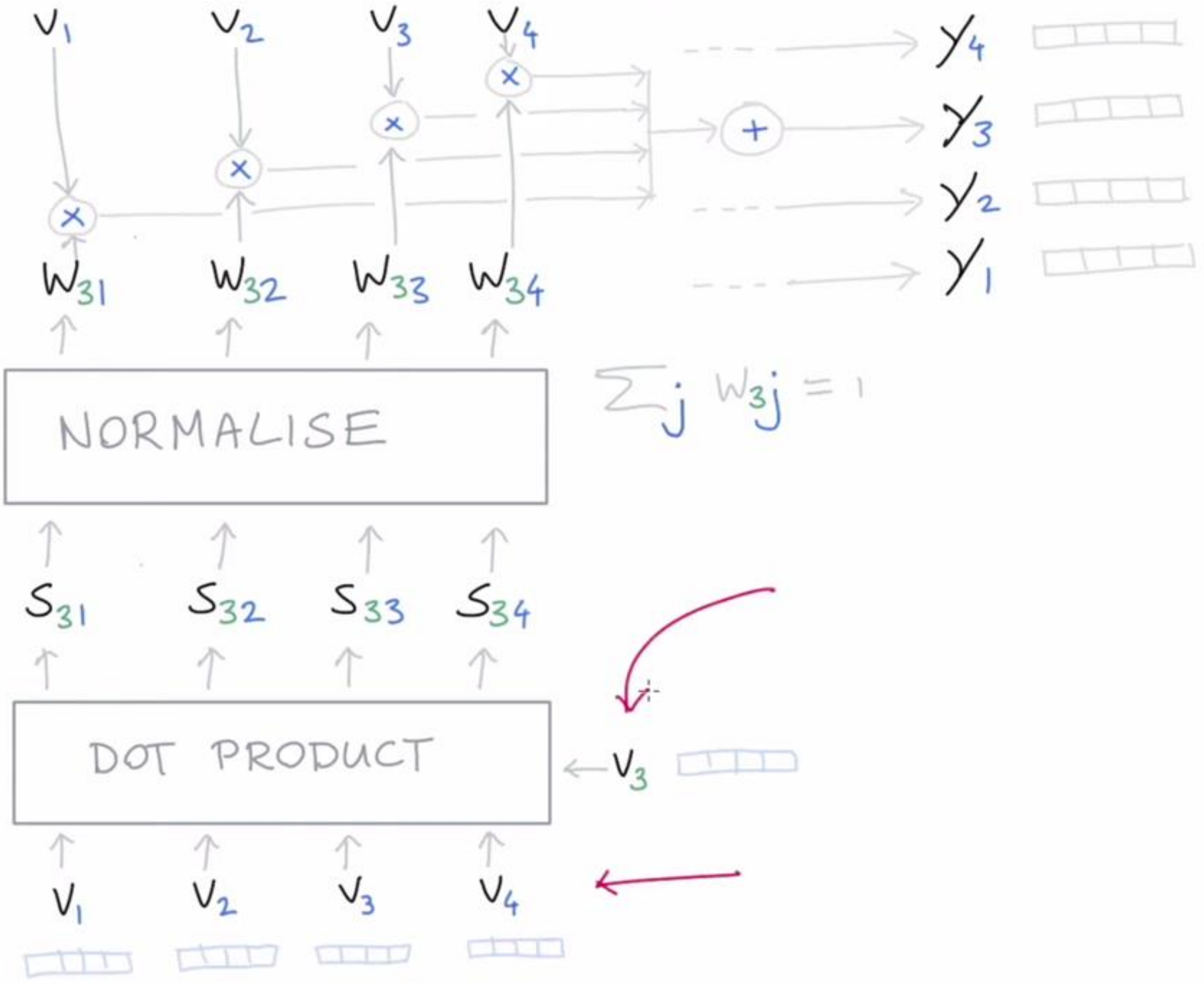
v1 v2 v3 v4
y4
y3
y2
y1
w31 w32 w33 w34
NORMALISE
Σj w3j = 1
y1 y2 y3 y4
s31 s32 s33 s34
no weights
ATTENTION
DOT PRODUCT
v3
v1 v2 v3 v4
v1 v2 v3 v4



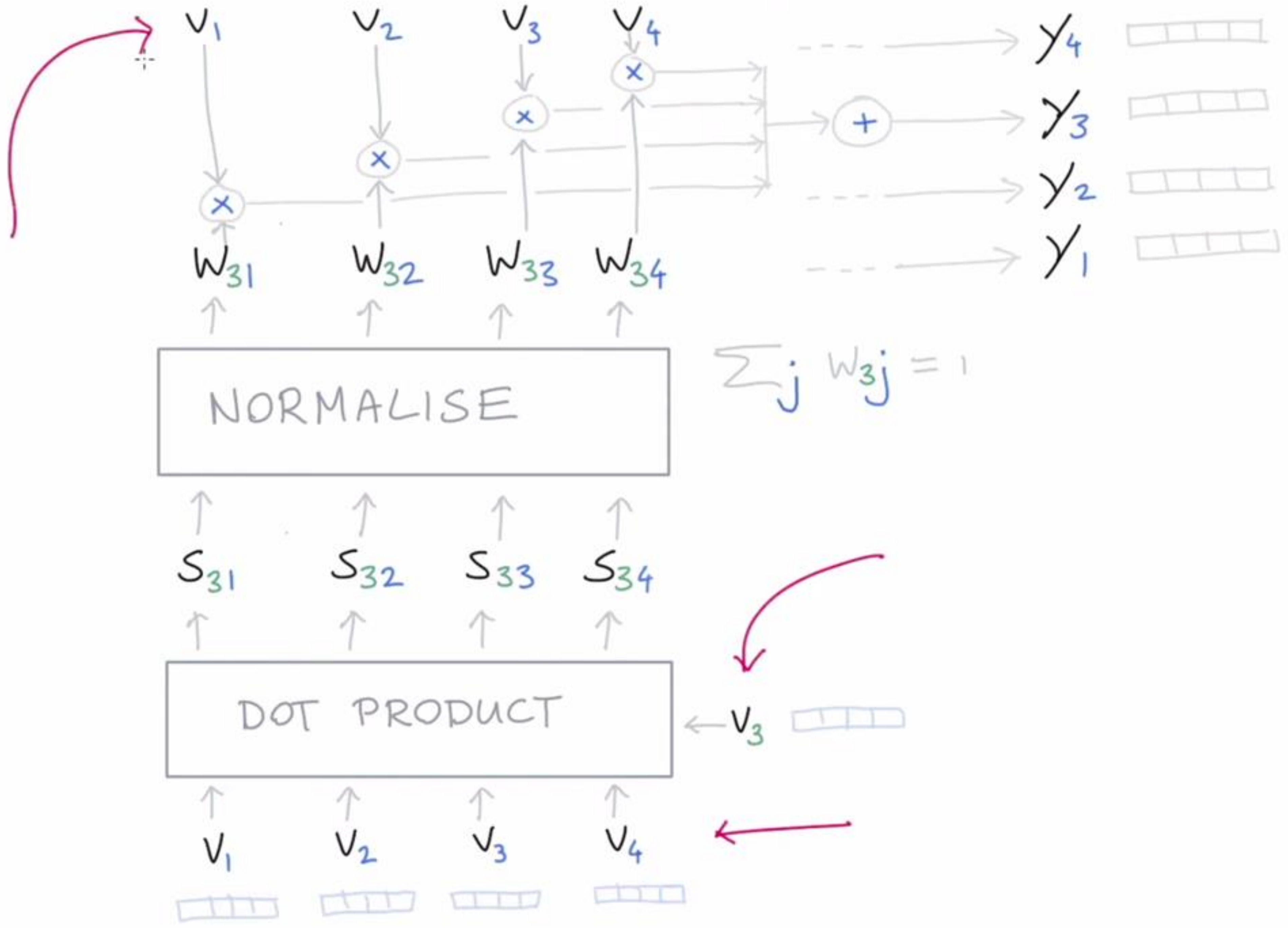
v1 v2 v3 v4
y4
y3
y2
y1
w31 w32 w33 w34
NORMALISE
Σj w3j = 1
y1 y2 y3 y4
s31 s32 s33 s34
~~no~~ weights
ATTENTION
DOT PRODUCT
v3
v1 v2 v3 v4



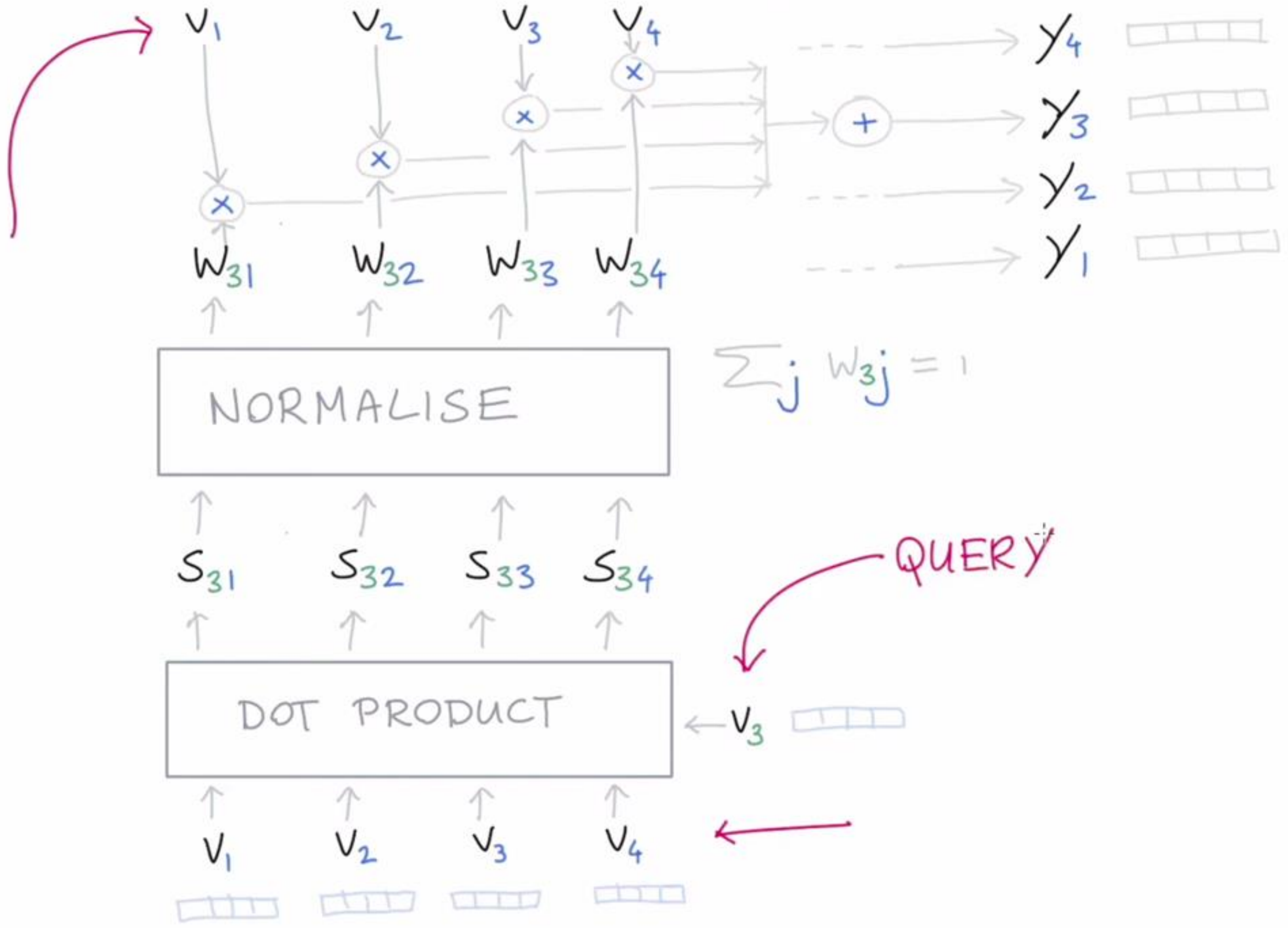
v1 v2 v3 v4
y4
y3
y2
y1
W31 W32 W33 W34
NORMALISE
Σj w3j = 1
S31 S32 S33 S34
DOT PRODUCT
v3
v1 v2 v3 v4



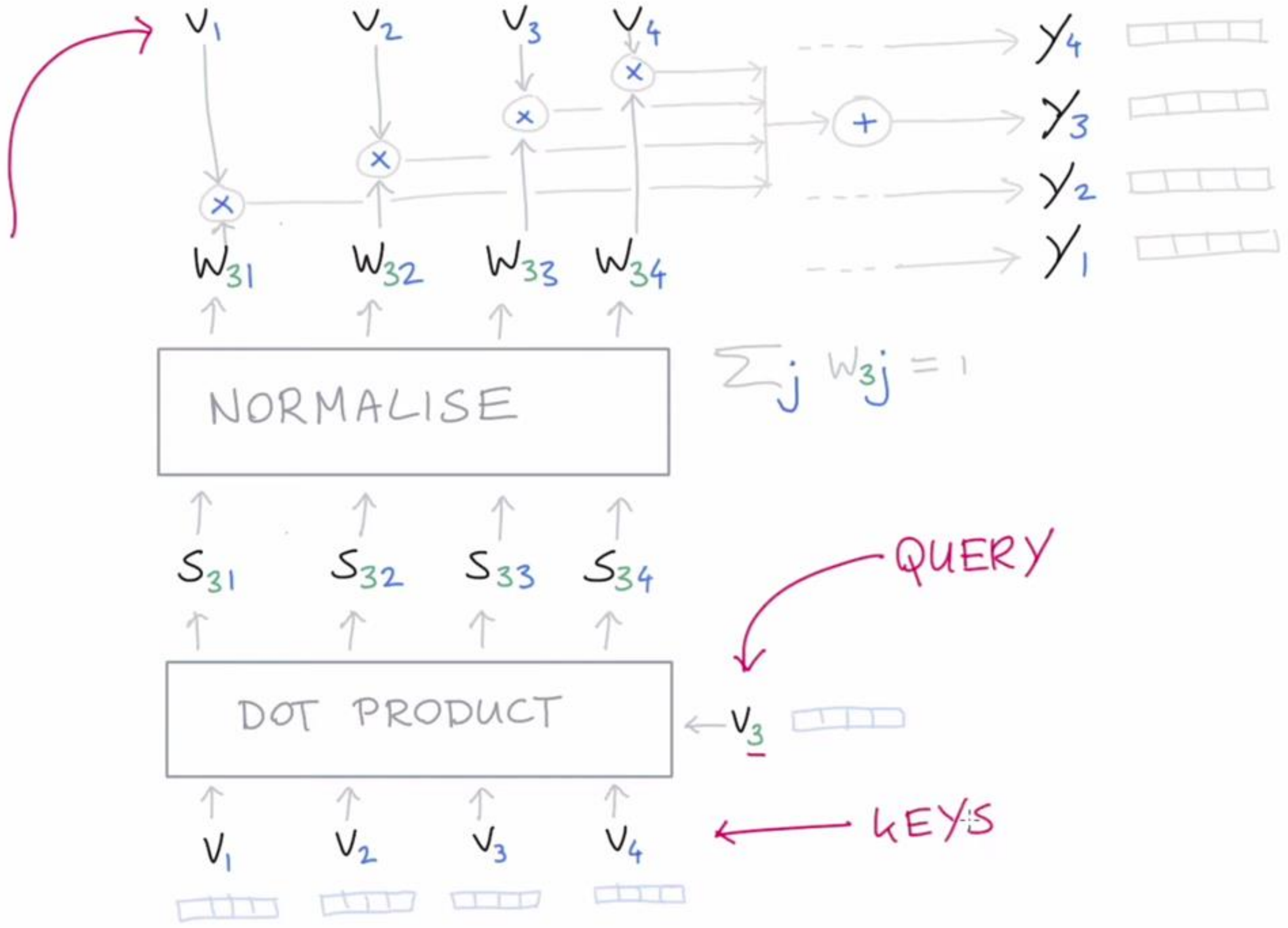
v_1 v_2 v_3 v_4
y_4
y_3
y_2
y_1
w_{31} w_{32} w_{33} w_{34}
NORMALISE
$\sum_j w_{3j} = 1$
s_{31} s_{32} s_{33} s_{34}
DOT PRODUCT
v_3
v_1 v_2 v_3 v_4

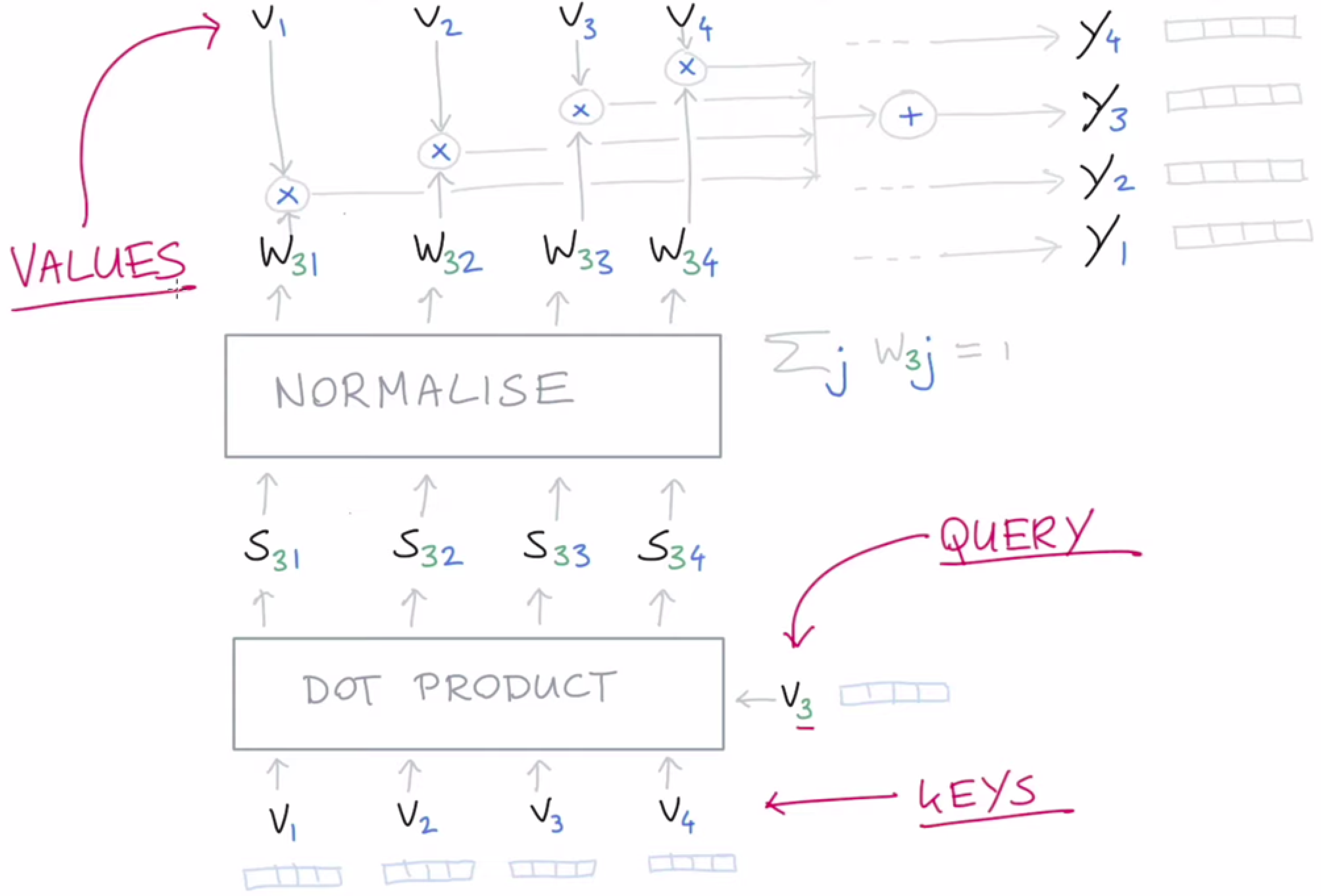


v_1 v_2 v_3 v_4
y_4
y_3
y_2
y_1
w_{31} w_{32} w_{33} w_{34}
NORMALISE
$\sum_j w_{3j} = 1$
s_{31} s_{32} s_{33} s_{34}
DOT PRODUCT
v_3
v_1 v_2 v_3 v_4



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
v_1 v_2 v_3 v_4

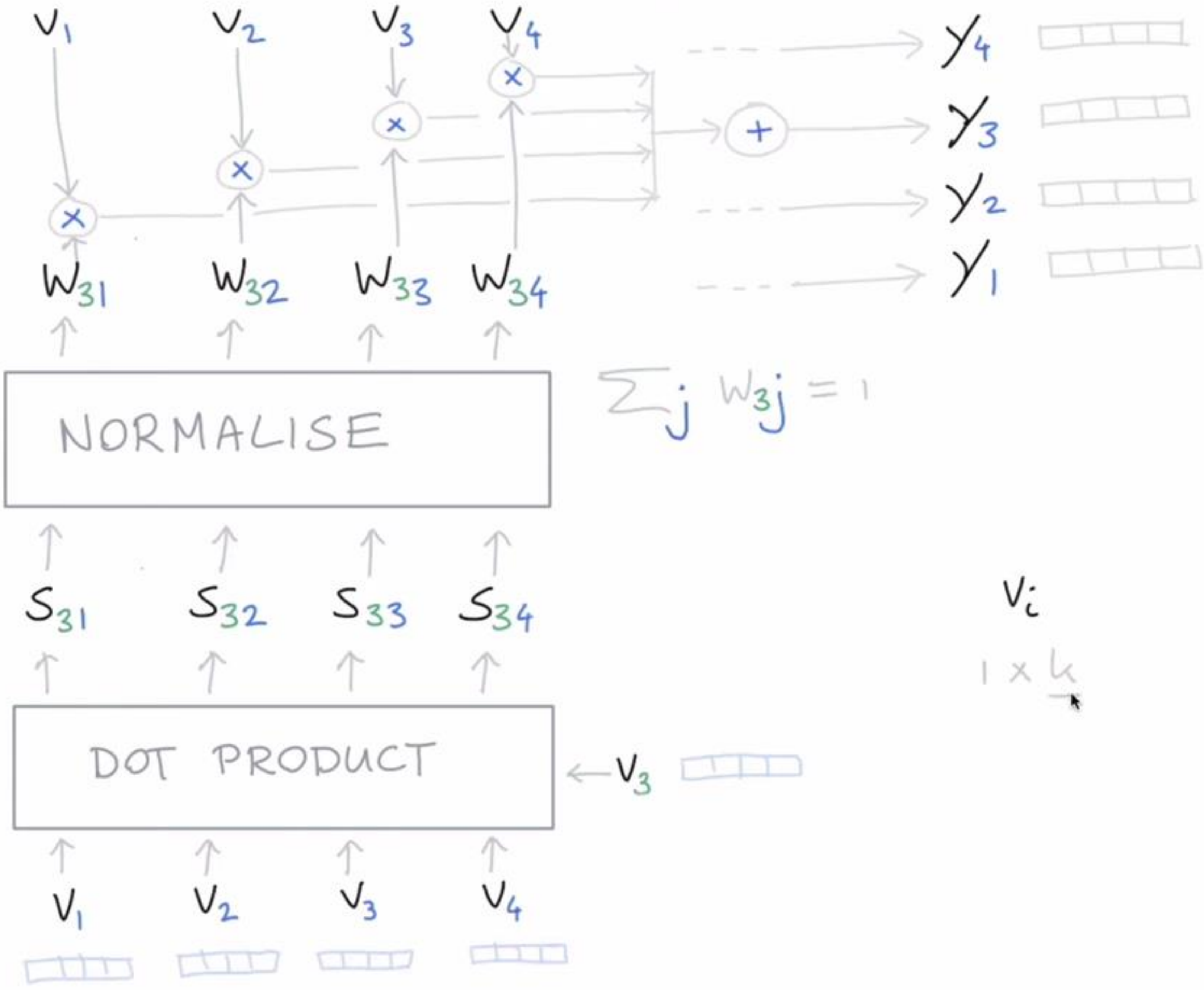

v1 v2 v3 v4
y4
y3
y2
y1
w31 w32 w33 w34
NORMALISE
Σj w3j = 1
s31 s32 s33 s34
QUERY
DOT PRODUCT
v3
KEYS
v1 v2 v3 v4



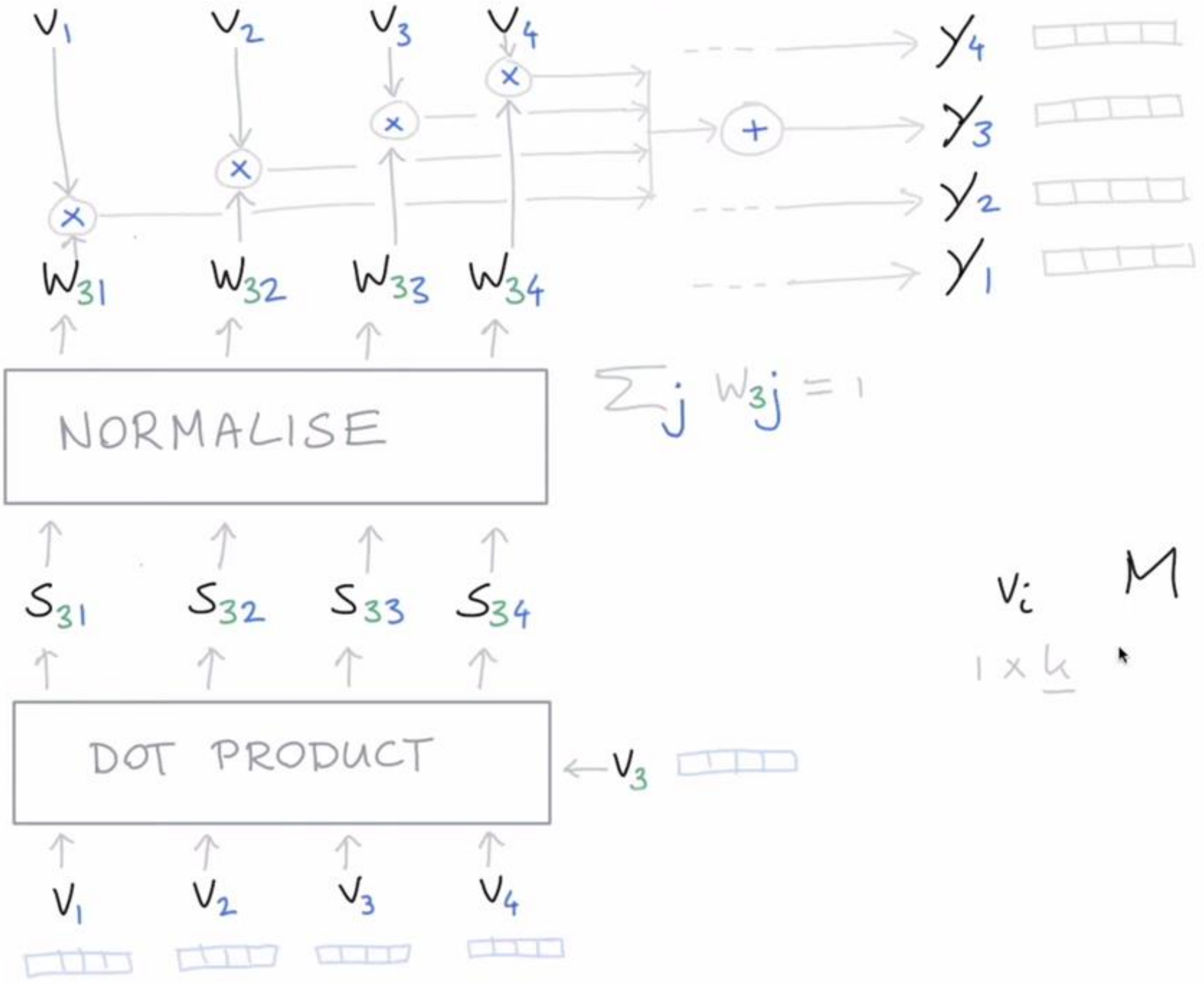
VALUES
v_1 v_2 v_3 v_4
w_{31} w_{32} w_{33} w_{34}
y_4 y_3 y_2 y_1
NORMALISE
$\sum_j w_{3j} = 1$
s_{31} s_{32} s_{33} s_{34}
QUERY
DOT PRODUCT
v_3
KEYS
v_1 v_2 v_3 v_4

Key, Value and Query

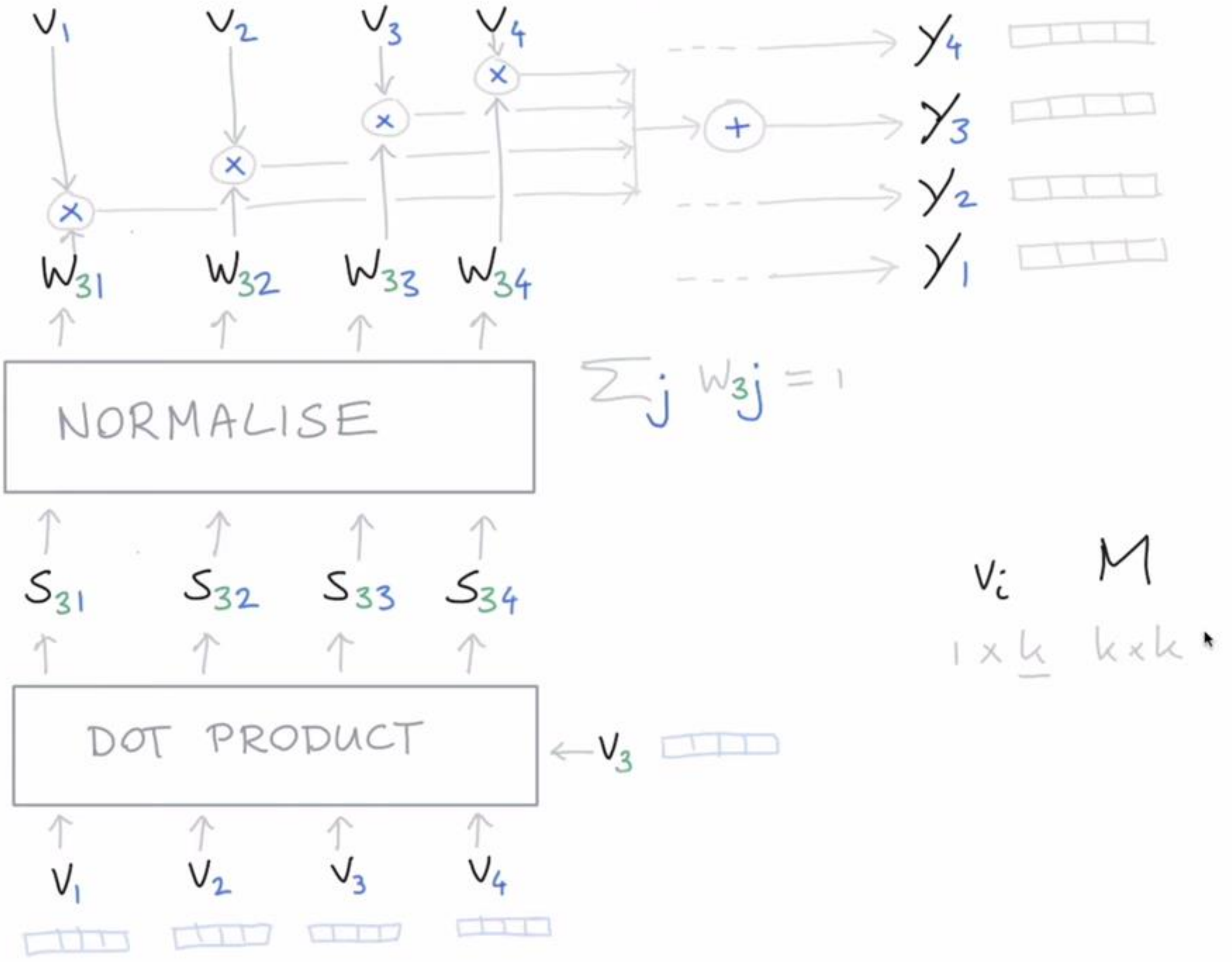
- The major component in the transformer is the unit of multi-head self-attention mechanism.



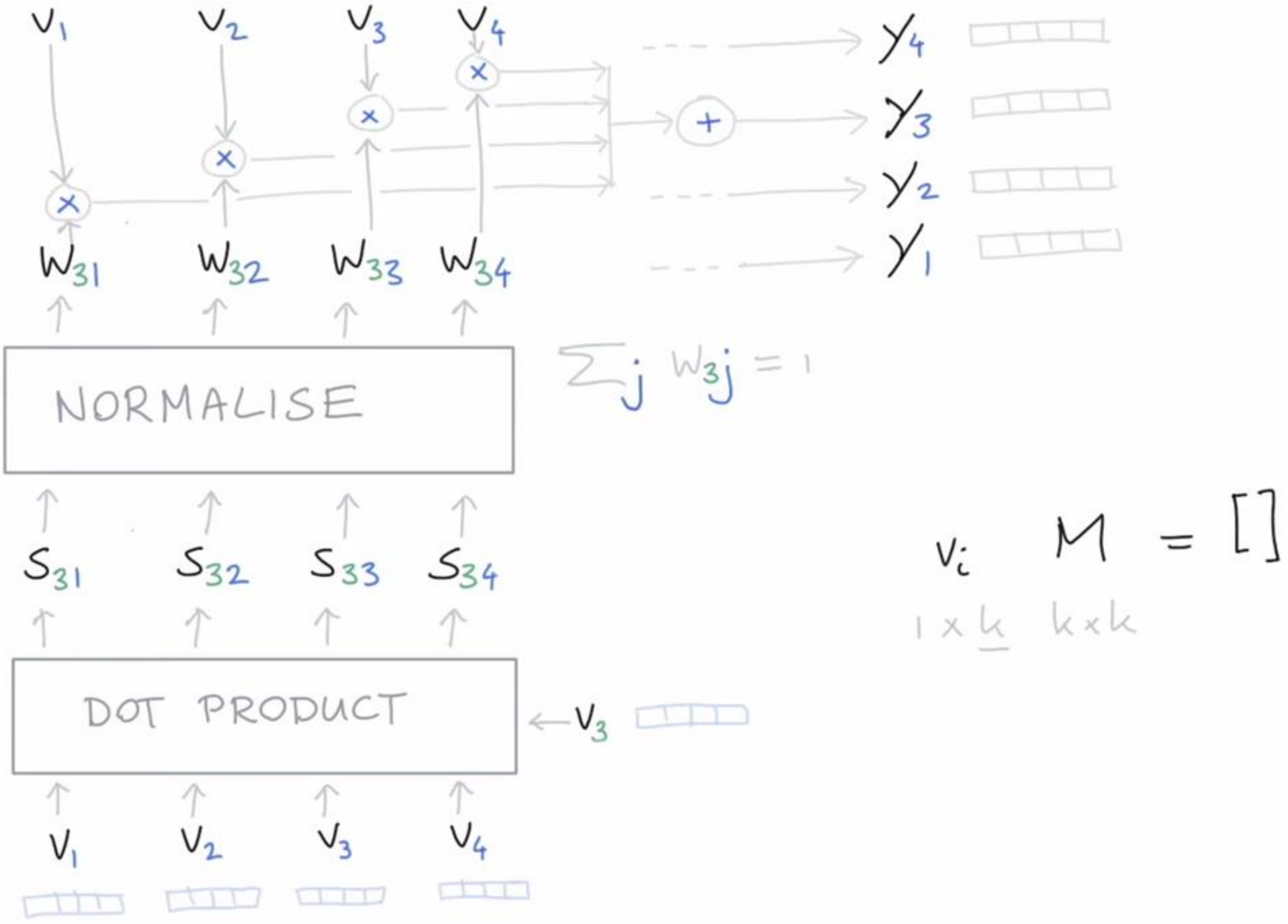
v_1 v_2 v_3 v_4
y_4
y_3
y_2
y_1
w_{31} w_{32} w_{33} w_{34}
NORMALISE
$\sum_j w_{3j} = 1$
s_{31} s_{32} s_{33} s_{34}
v_i
$1 \times k$
DOT PRODUCT
v_3
v_1 v_2 v_3 v_4



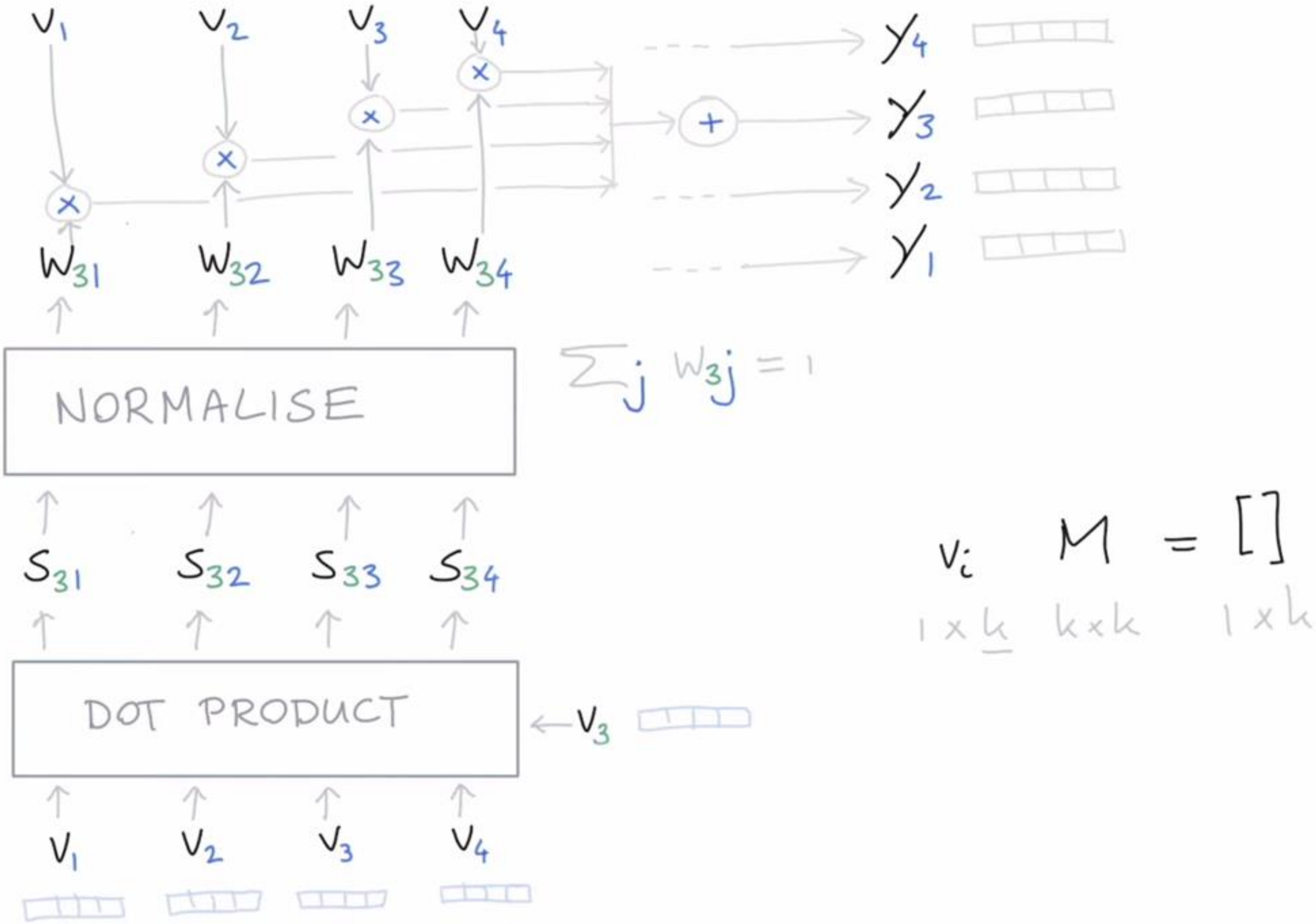
v_1 v_2 v_3 v_4
y_4 y_3 y_2 y_1
w_{31} w_{32} w_{33} w_{34}
NORMALISE
$\sum_j w_{3j} = 1$
s_{31} s_{32} s_{33} s_{34}
DOT PRODUCT
v_3
v_1 v_2 v_3 v_4
v_i M
$1 \times k$



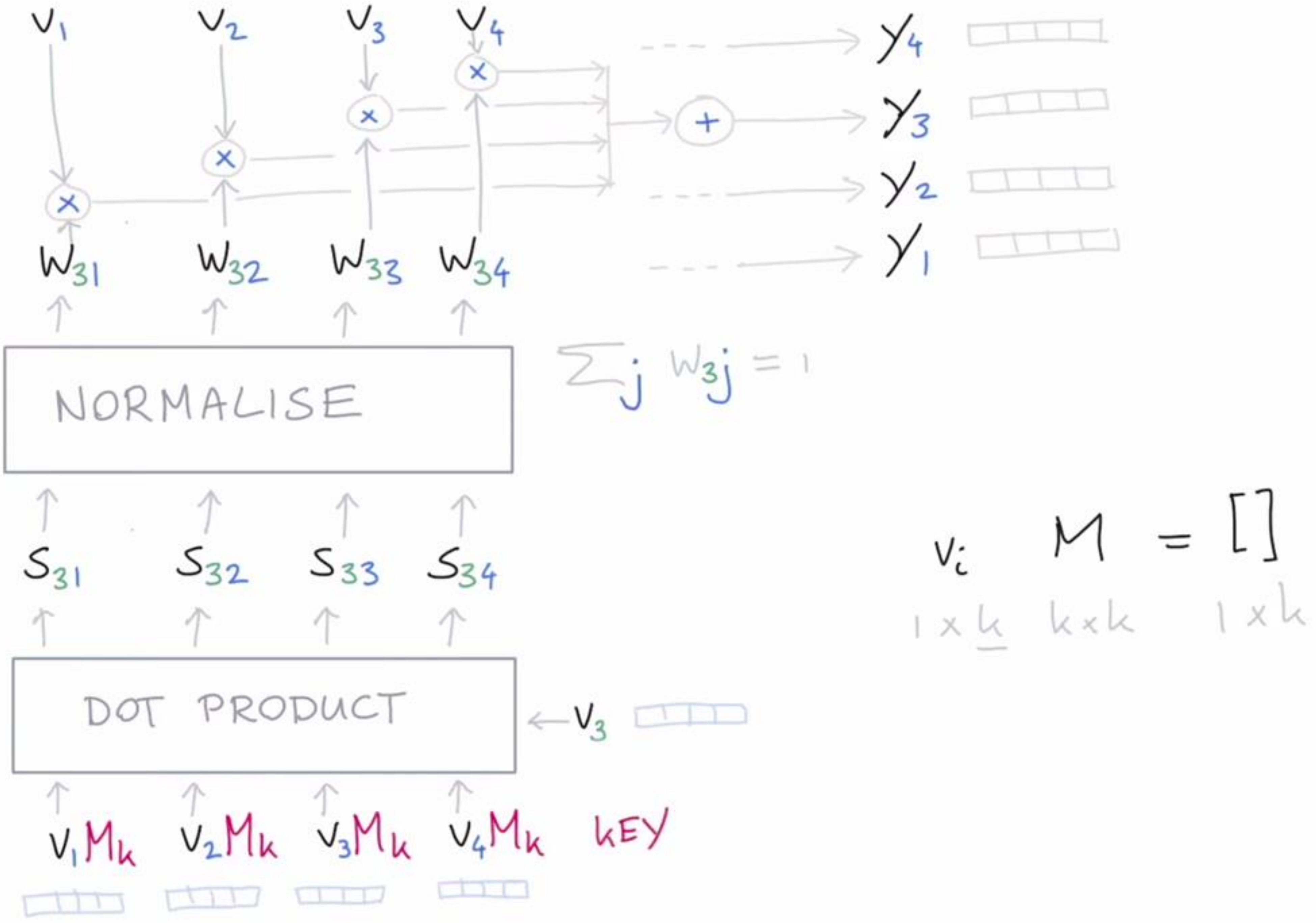
v1 v2 v3 v4
×
+
y4 y3 y2 y1
W31 W32 W33 W34
NORMALISE
Σj W3j = 1
S31 S32 S33 S34
DOT PRODUCT
v3
v1 v2 v3 v4
vi M
1×k k×k



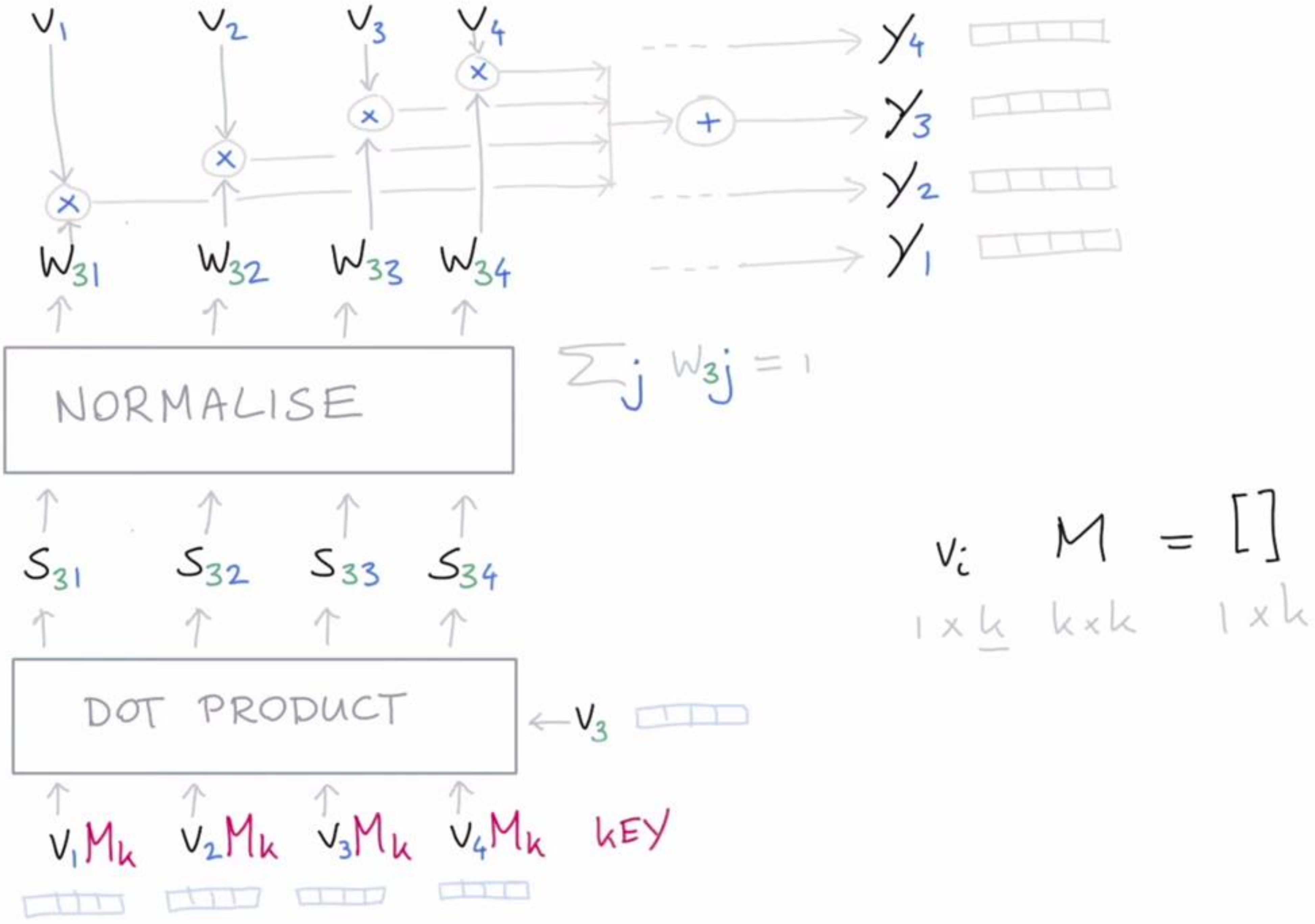
v1 v2 v3 v4
W31 W32 W33 W34
y4 y3 y2 y1
NORMALISE
Σj W3j = 1
S31 S32 S33 S34
DOT PRODUCT
v3
v1 v2 v3 v4
vi M = []
1×k k×k

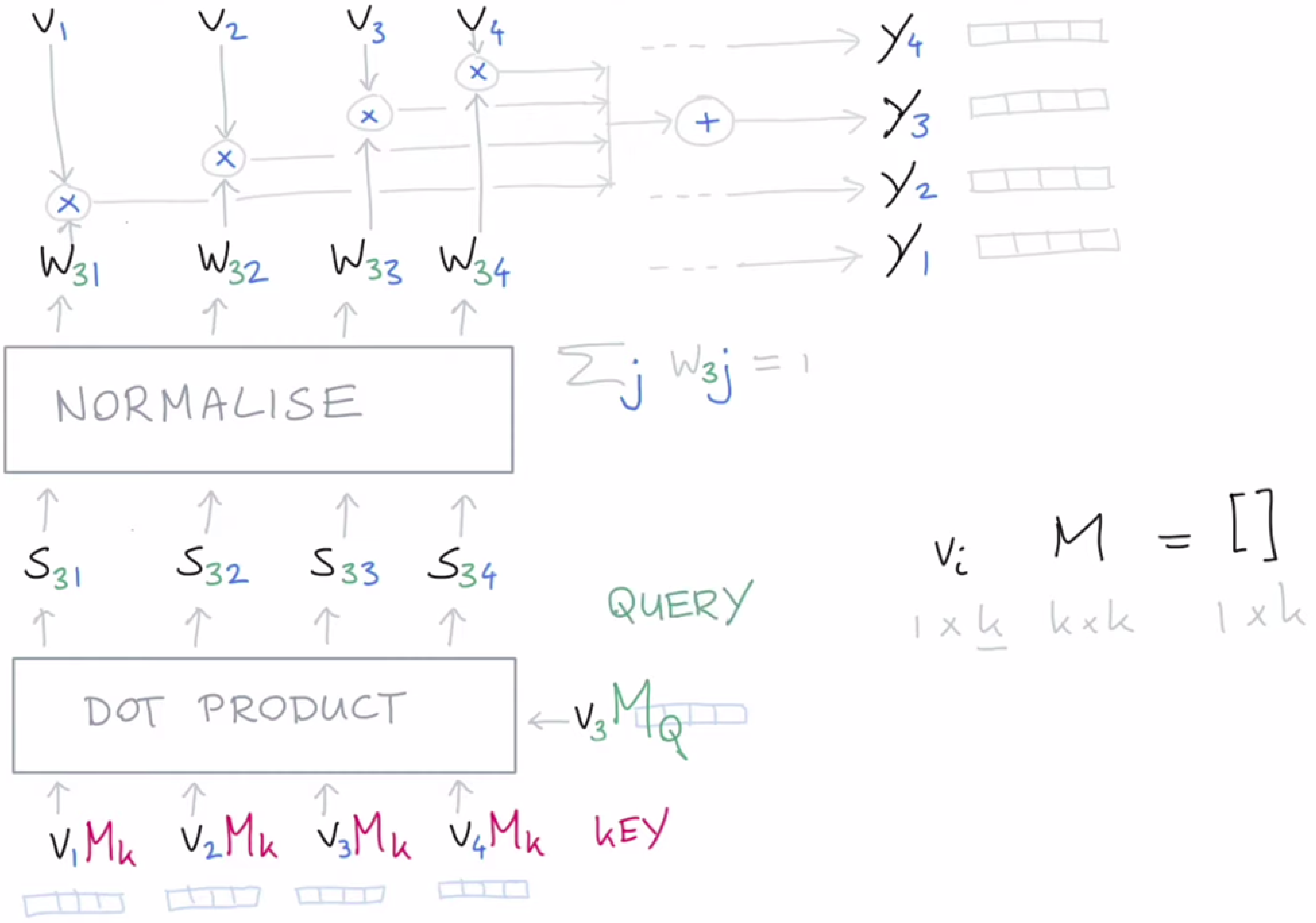


v_1 v_2 v_3 v_4
y_4 y_3 y_2 y_1
w_{31} w_{32} w_{33} w_{34}
NORMALISE
$\sum_j w_{3j} = 1$
s_{31} s_{32} s_{33} s_{34}
$v_i \quad M = [\,]$
$1 \times k \quad k \times k \quad 1 \times k$
DOT PRODUCT
v_3
v_1 v_2 v_3 v_4

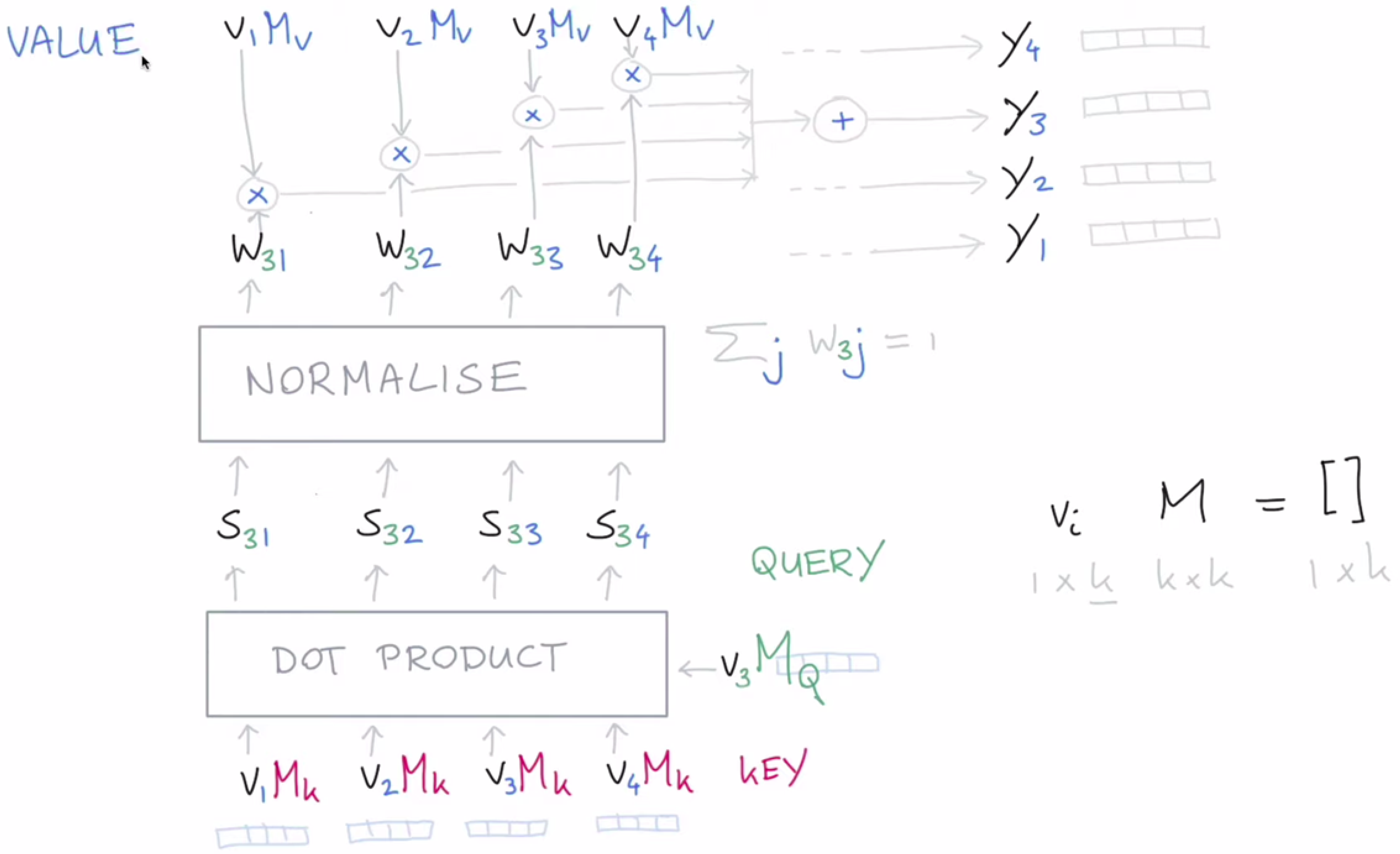


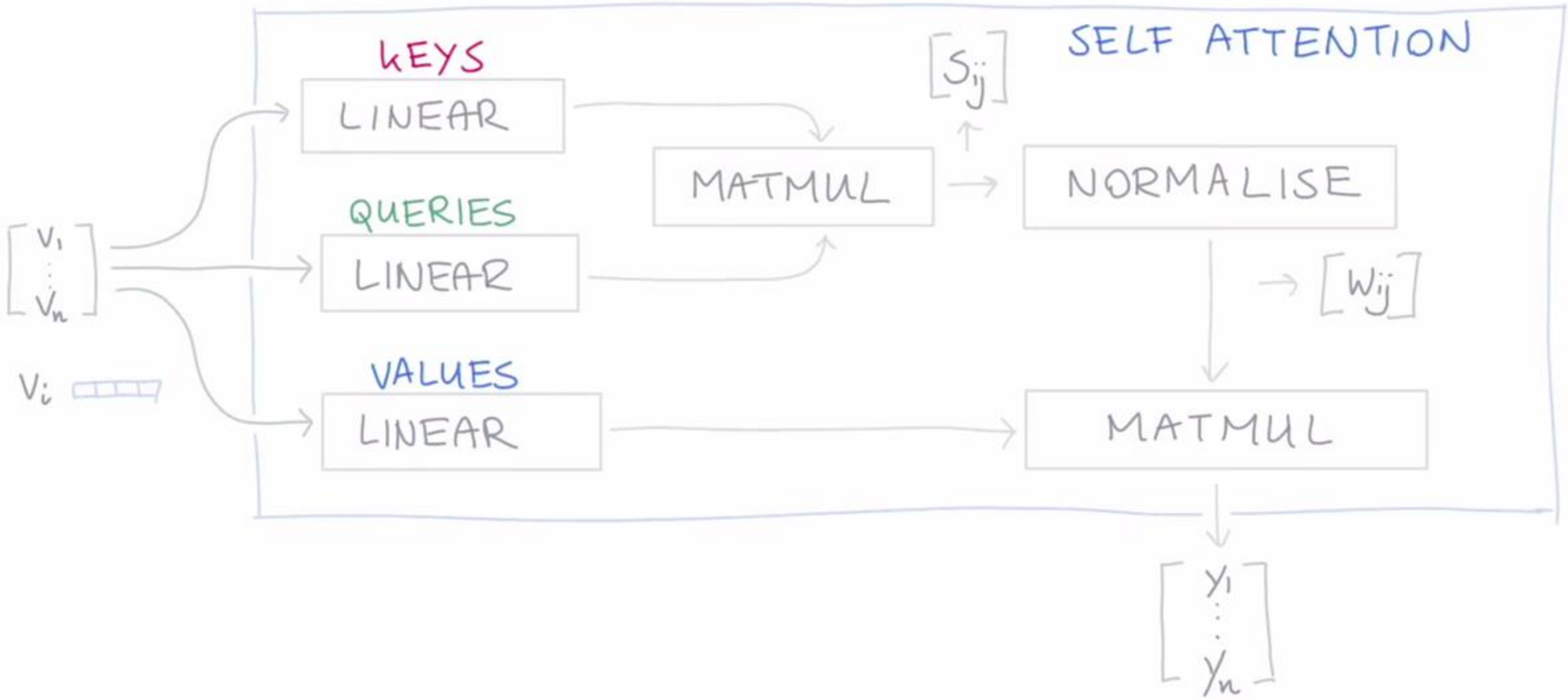
v_1 v_2 v_3 v_4
y_4
y_3
y_2
y_1
w_{31} w_{32} w_{33} w_{34}
NORMALISE
$\sum_j w_{3j} = 1$
$v_i \quad M = [\,]$
$1 \times \underline{k} \quad k \times k \quad 1 \times k$
s_{31} s_{32} s_{33} s_{34}
DOT PRODUCT
$\leftarrow v_3$
$v_1 M_k$ $v_2 M_k$ $v_3 M_k$ $v_4 M_k$ KEY

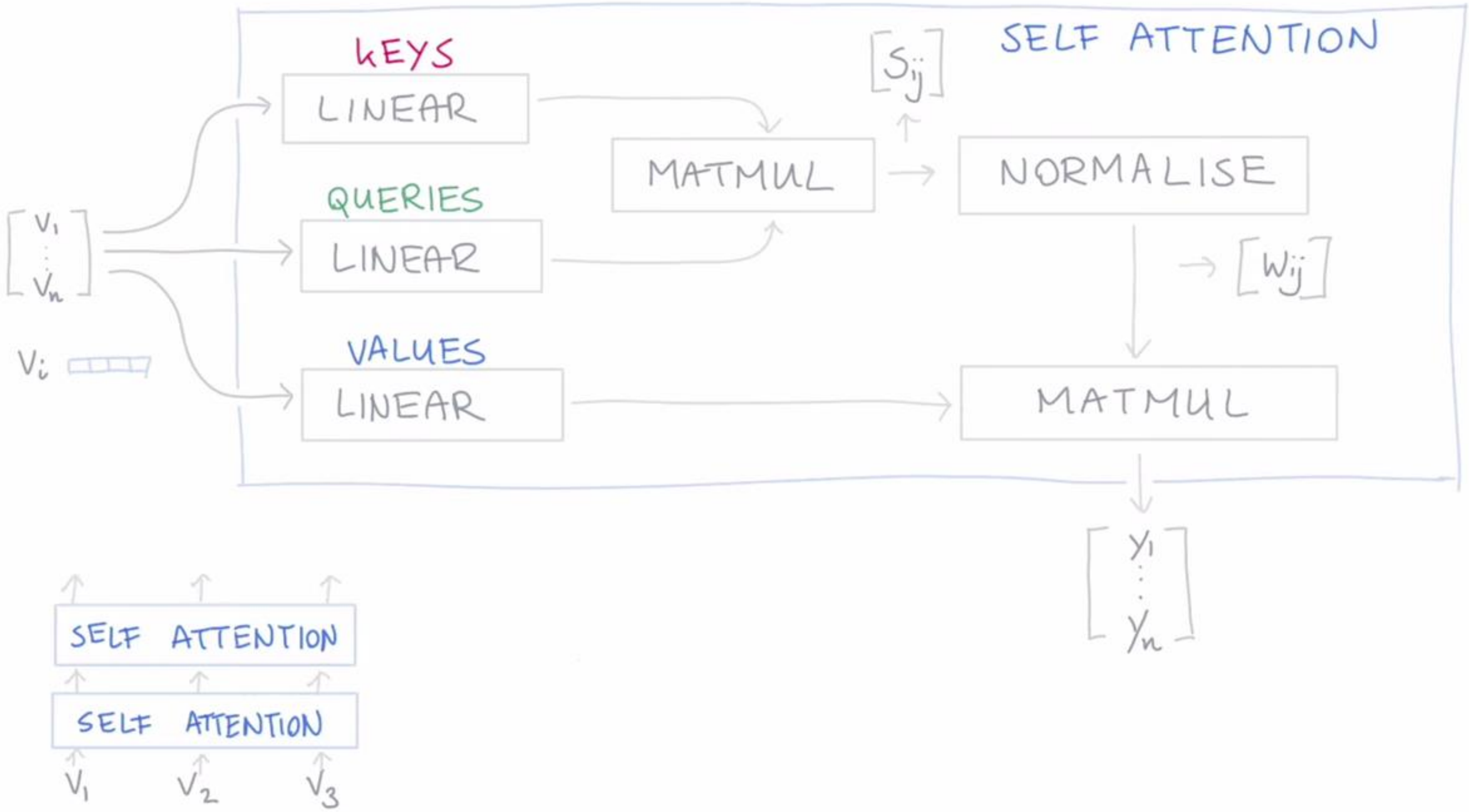

v1 v2 v3 v4
y4
y3
y2
y1
W31 W32 W33 W34
NORMALISE
Σj W3j = 1
vi M = []
1 x k k x k 1 x k
S31 S32 S33 S34
DOT PRODUCT
v3
v1Mk v2Mk v3Mk v4Mk KEY



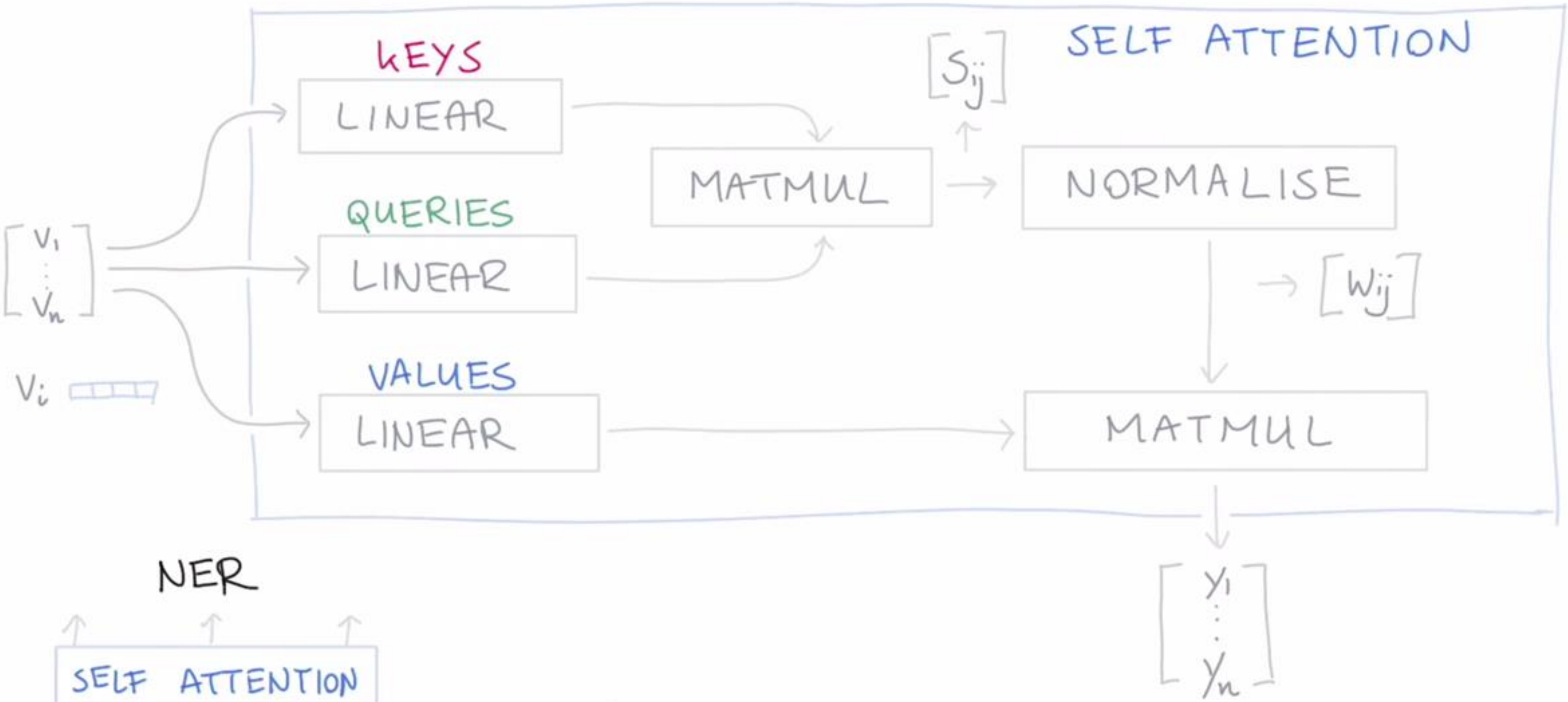
v1 v2 v3 v4
×
+
y4 y3 y2 y1
W31 W32 W33 W34
NORMALISE
Σj W3j = 1
S31 S32 S33 S34
QUERY
DOT PRODUCT
v3MQ
v1Mk v2Mk v3Mk v4Mk
KEY
vi M = []
1×k k×k 1×k


VALUE
$v_1 M_v$
$v_2 M_v$
$v_3 M_v$
$v_4 M_v$
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
$v_3 M_Q$
$v_1 M_k$
$v_2 M_k$
$v_3 M_k$
$v_4 M_k$
KEY
$v_i \quad M \quad = [\,]$
$1 \times k \quad k \times k \quad 1 \times k$

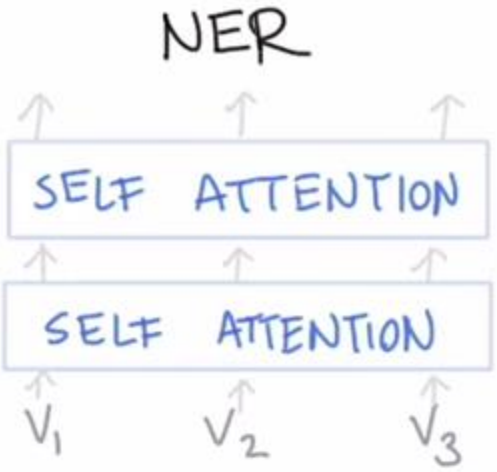

SELF ATTENTION
KEYS
LINEAR
QUERIES
LINEAR
VALUES
LINEAR
MATMUL
NORMALISE
MATMUL
[s_{ij}]
[w_{ij}]
$v_1 \dots v_n$
v_i
$y_1 \dots y_n$



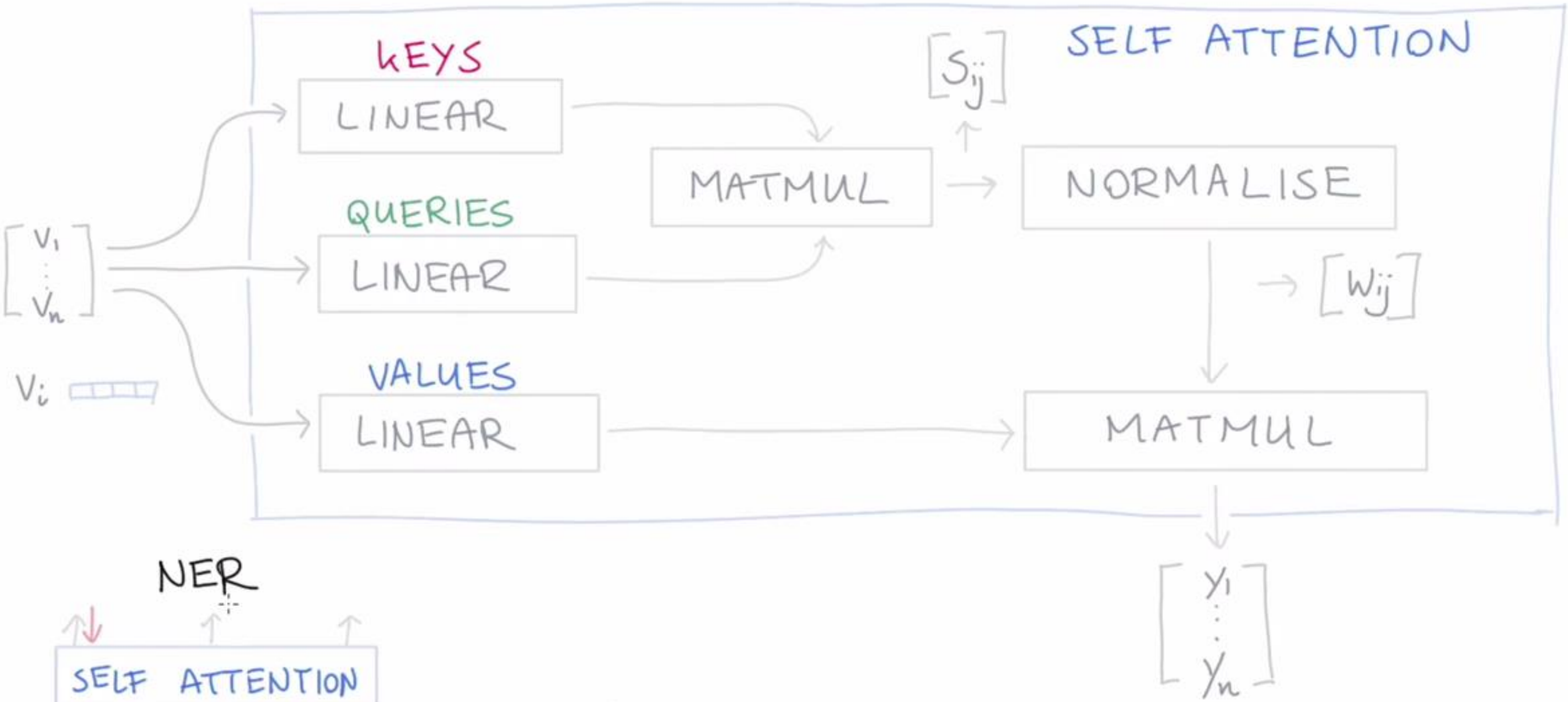
KEYS
LINEAR
QUERIES
LINEAR
VALUES
LINEAR
MATMUL
$[S_{ij}]$
SELF ATTENTION
NORMALISE
$[W_{ij}]$
MATMUL
V_1 … V_n
V_i
y_1 … y_n
SELF ATTENTION
SELF ATTENTION
V_1 V_2 V_3

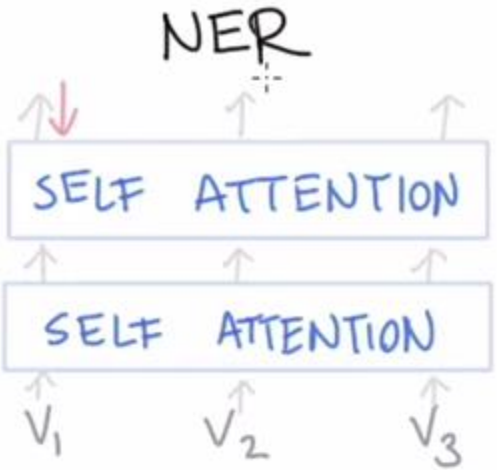


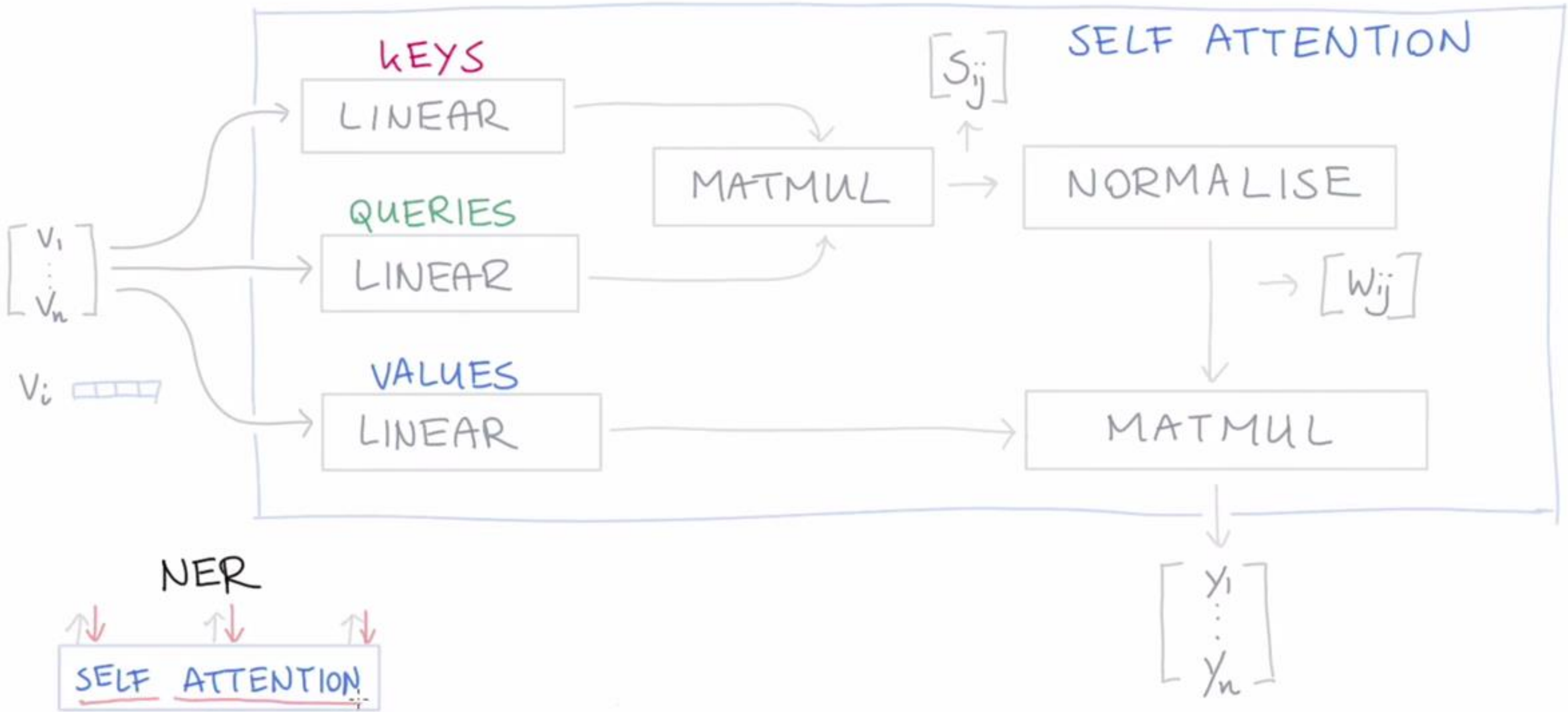
SELF ATTENTION
KEYS
LINEAR
QUERIES
LINEAR
VALUES
LINEAR
MATMUL
$[S_{ij}]$
NORMALISE
$[W_{ij}]$
MATMUL
V_1
V_n
V_i
y_1
y_n

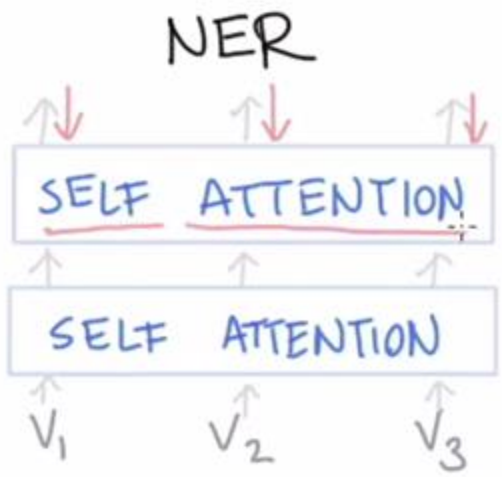


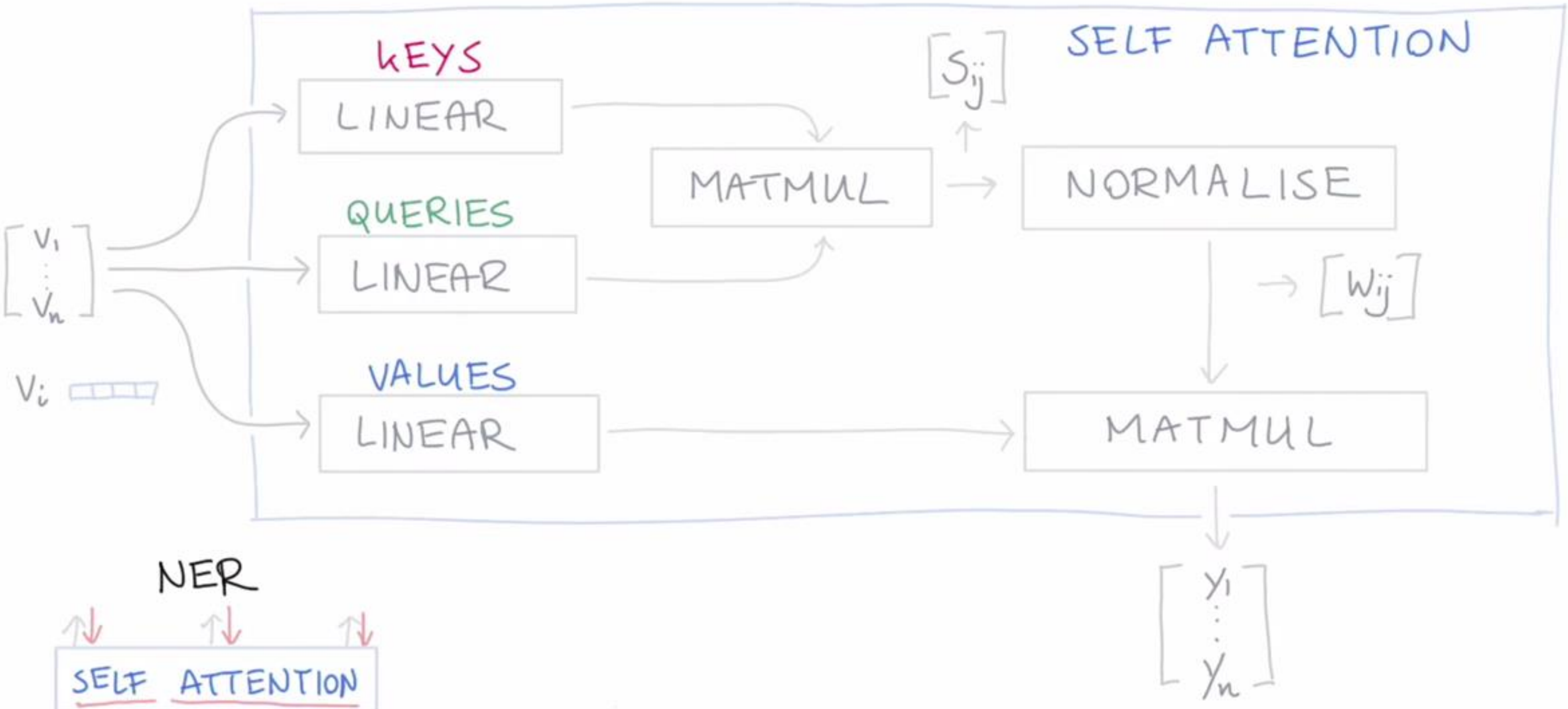
NER
SELF ATTENTION
SELF ATTENTION
V_1
V_2
V_3

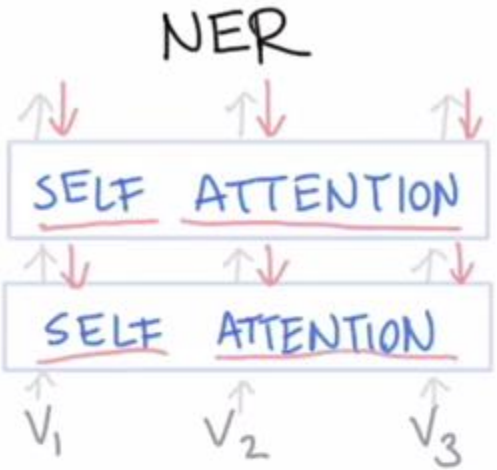

SELF ATTENTION
KEYS
LINEAR
QUERIES
LINEAR
VALUES
LINEAR
MATMUL
$[S_{ij}]$
NORMALISE
$[W_{ij}]$
MATMUL
v_1 ... v_n
v_i
y_1 ... y_n

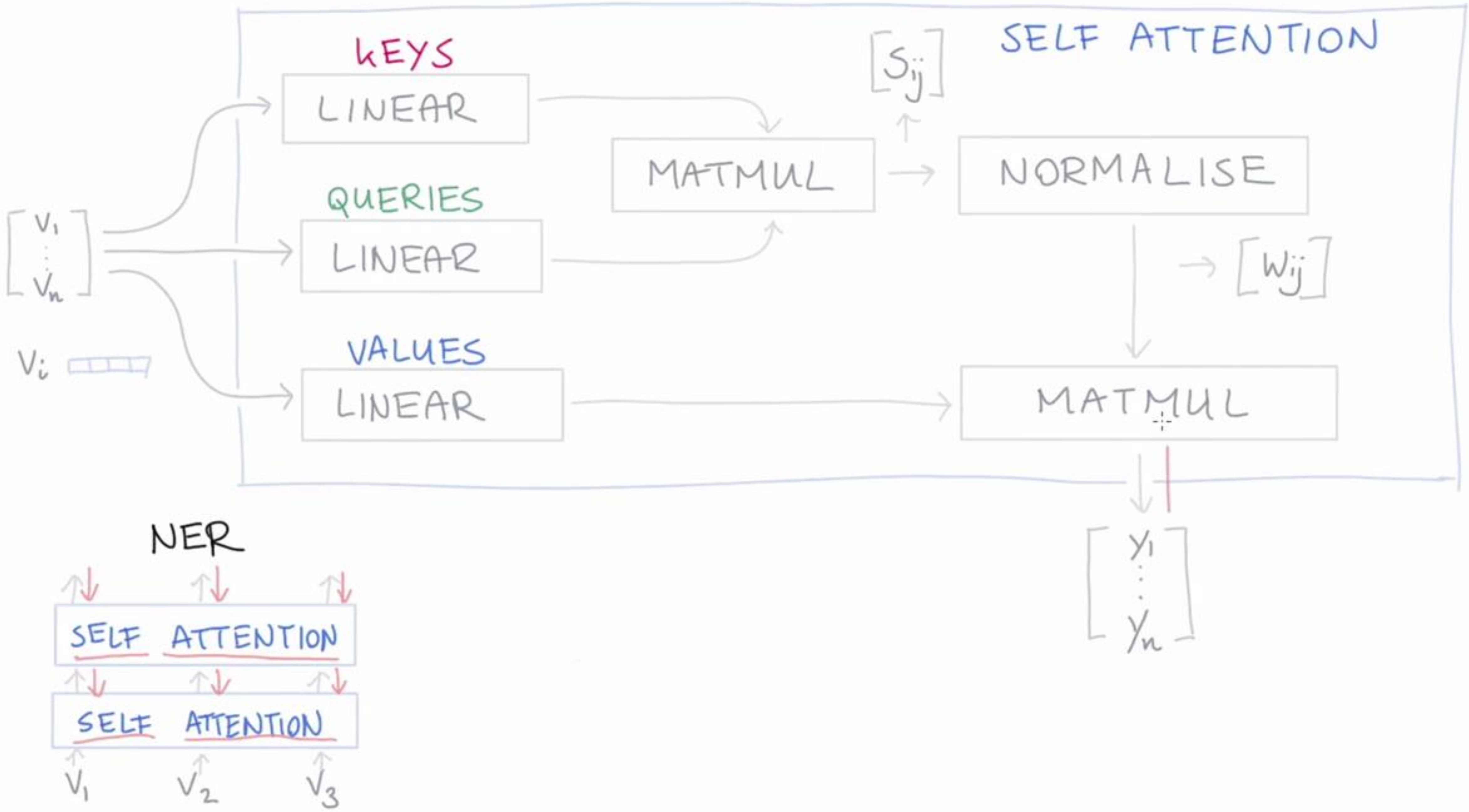

NER
SELF ATTENTION
SELF ATTENTION
v_1
v_2
v_3

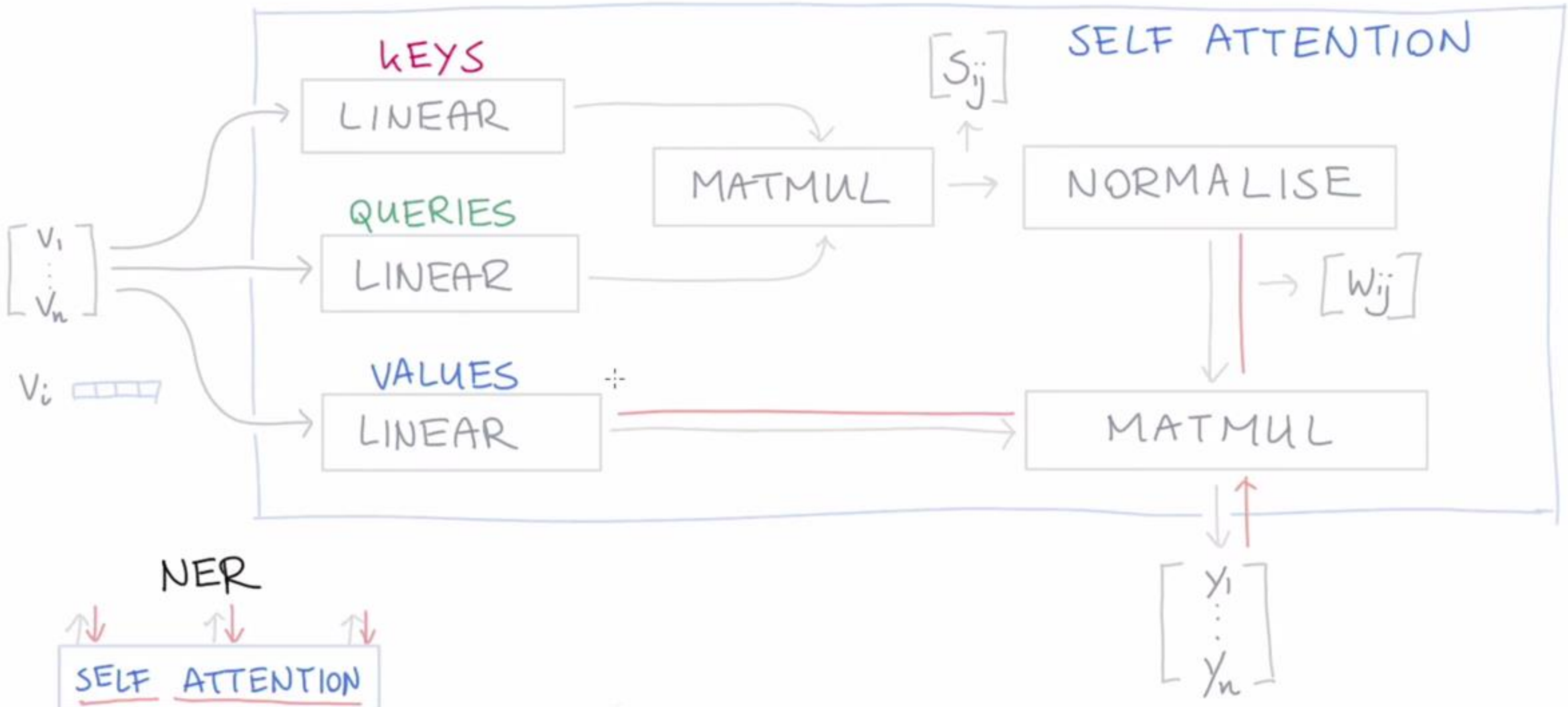

SELF ATTENTION
KEYS
LINEAR
QUERIES
LINEAR
VALUES
LINEAR
MATMUL
$[S_{ij}]$
NORMALISE
$[W_{ij}]$
MATMUL
V_1 … V_n
V_i
y_1 … y_n

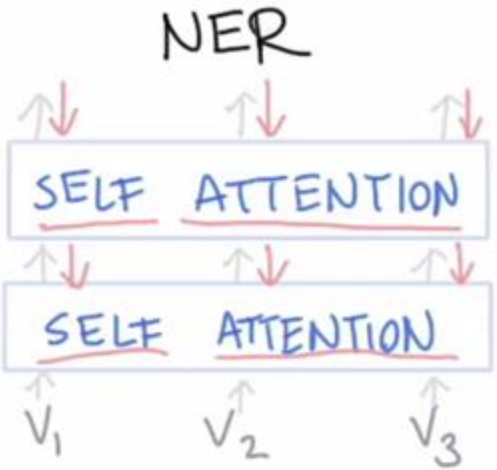

NER
SELF ATTENTION
SELF ATTENTION
V_1
V_2
V_3


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

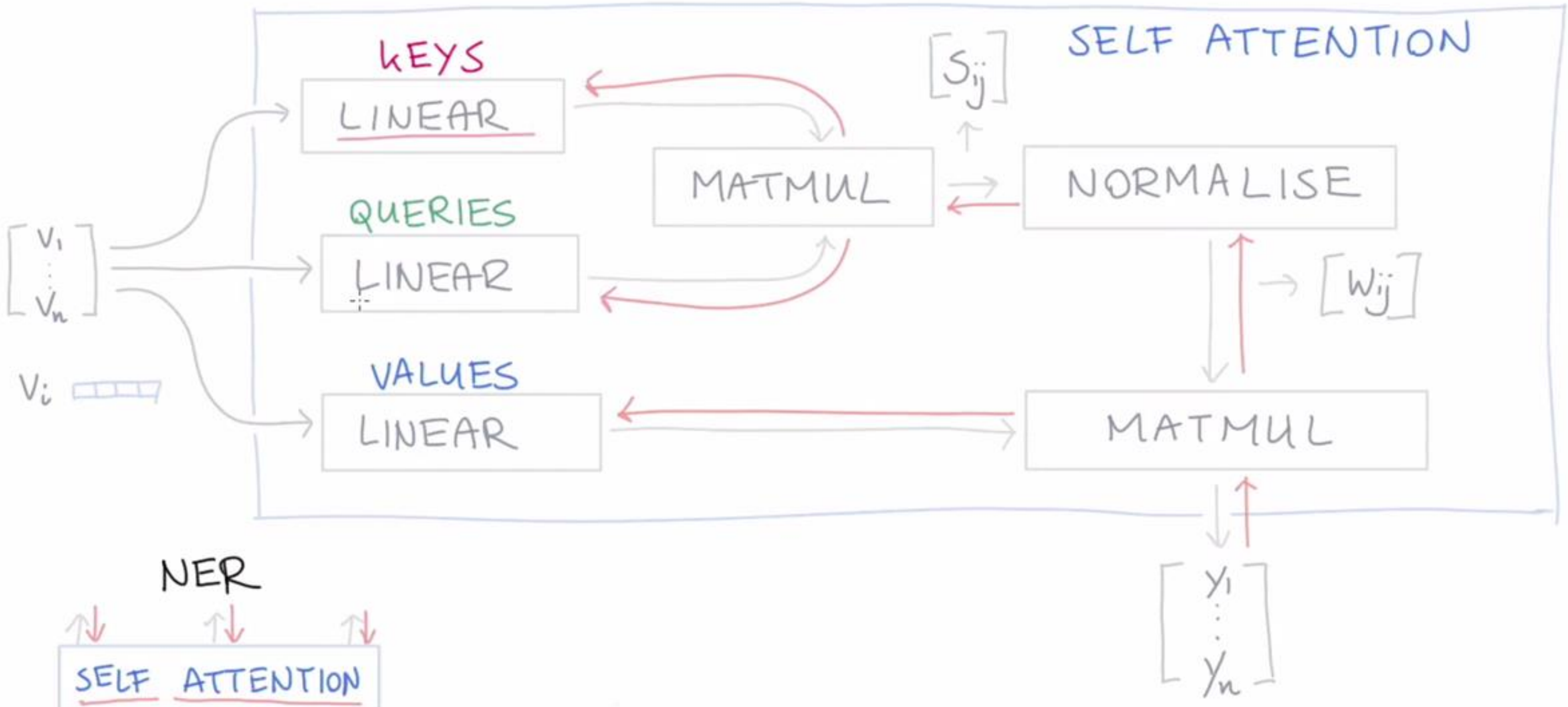

NER
SELF ATTENTION
SELF ATTENTION
V_1
V_2
V_3


SELF ATTENTION
KEYS
LINEAR
QUERIES
LINEAR
VALUES
LINEAR
MATMUL
$[S_{ij}]$
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

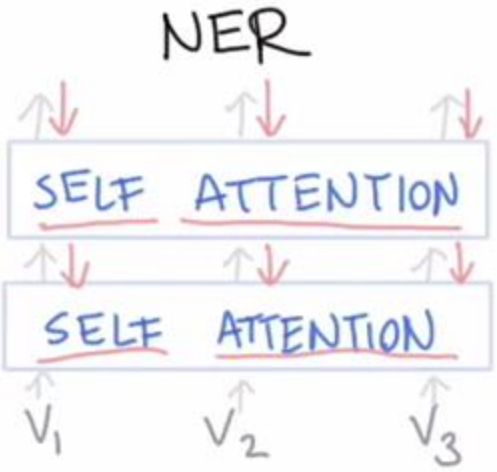




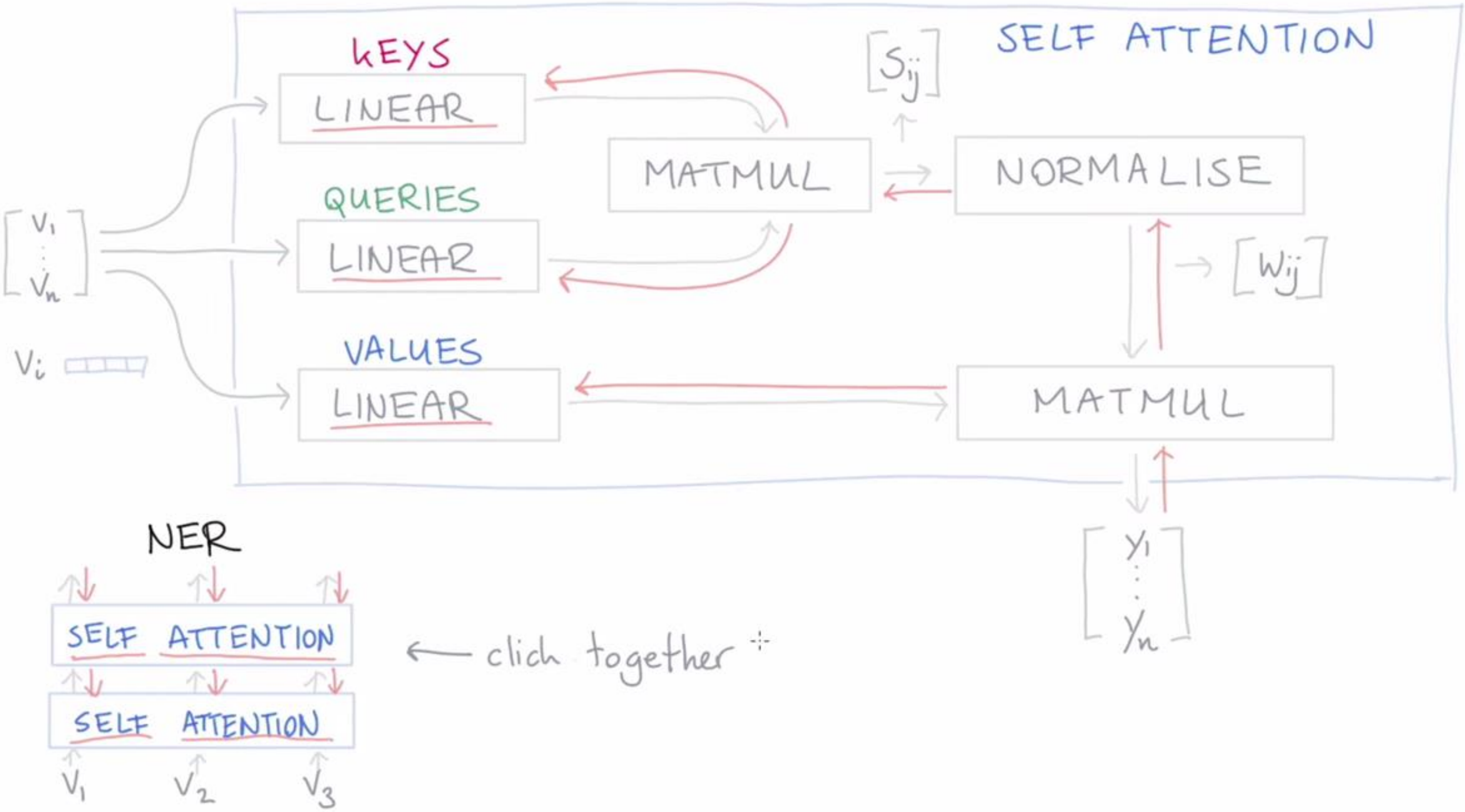
https://learning.rasa.com/transformers/self-attention/



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


SELF ATTENTION
KEYS
QUERIES
VALUES
LINEAR
LINEAR
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
← click together

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

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

[*]Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and

https://arxiv.org/abs/1706.03762

The annotated Transformer (Code)

- The Annotated Transformer (harvard.edu)

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/

- 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