

# Natural Language Processing

CS 3216/UG, AI 5203/PG

Week-8

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

# History of Neural models in NLP

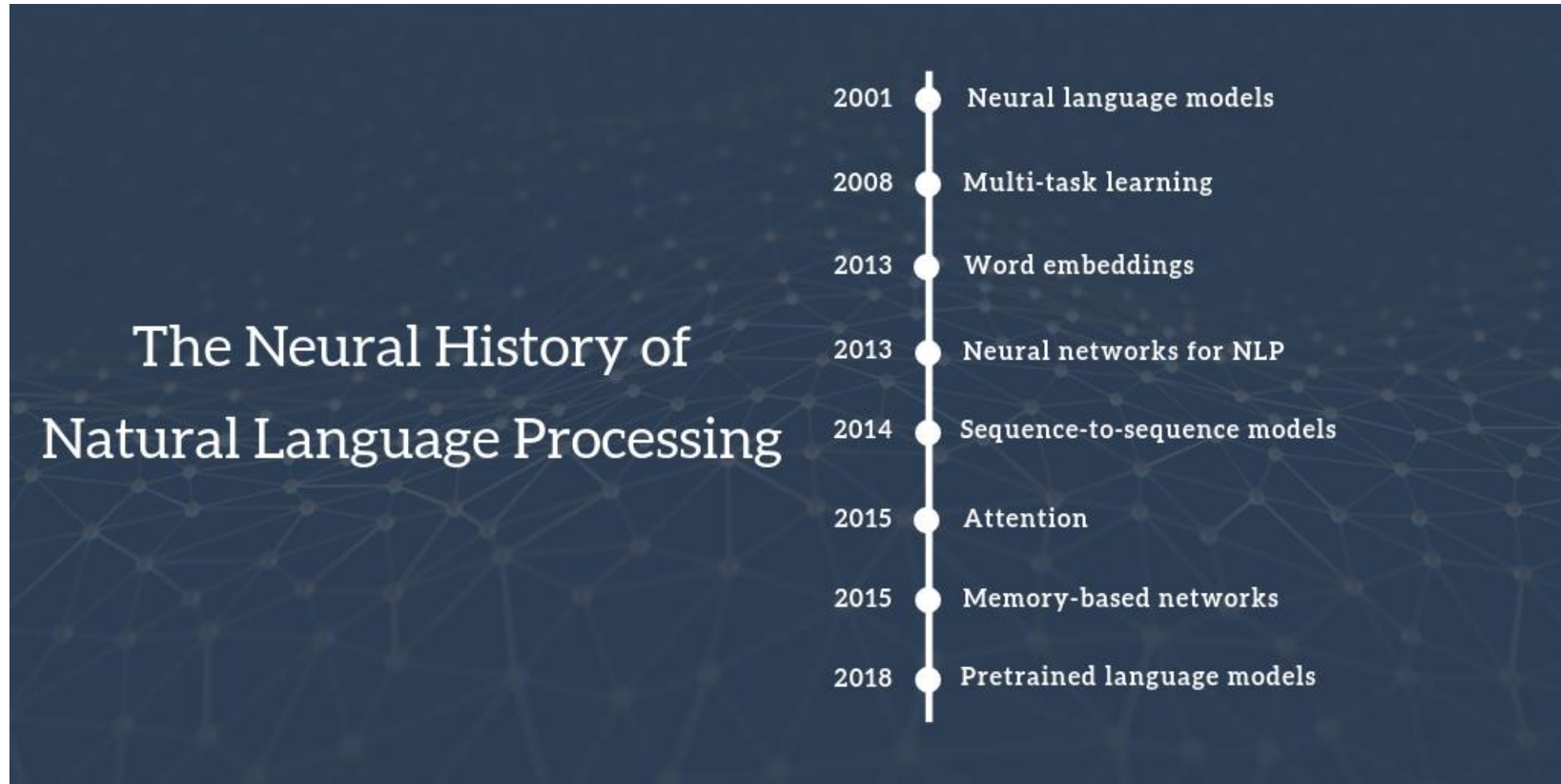
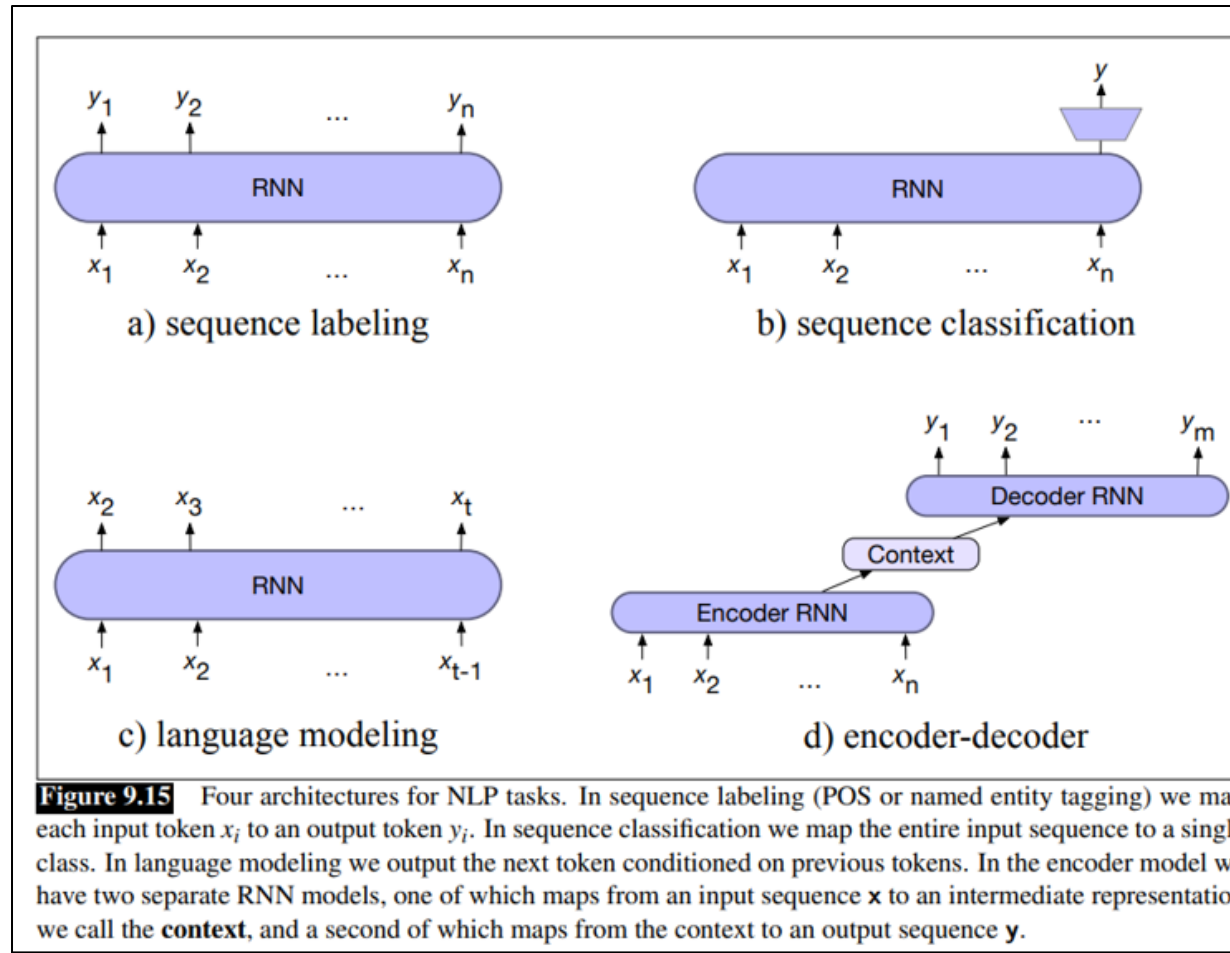


Image source : <https://www.ruder.io/a-review-of-the-recent-history-of-nlp/>

# Different variants of RNN

- Stacked RNN
- Bi-directional RNN
- Many more

# Sequence -to-Sequence learning



---

# Sequence to Sequence Learning with Neural Networks

---

Ilya Sutskever  
Google  
ilyasu@google.com

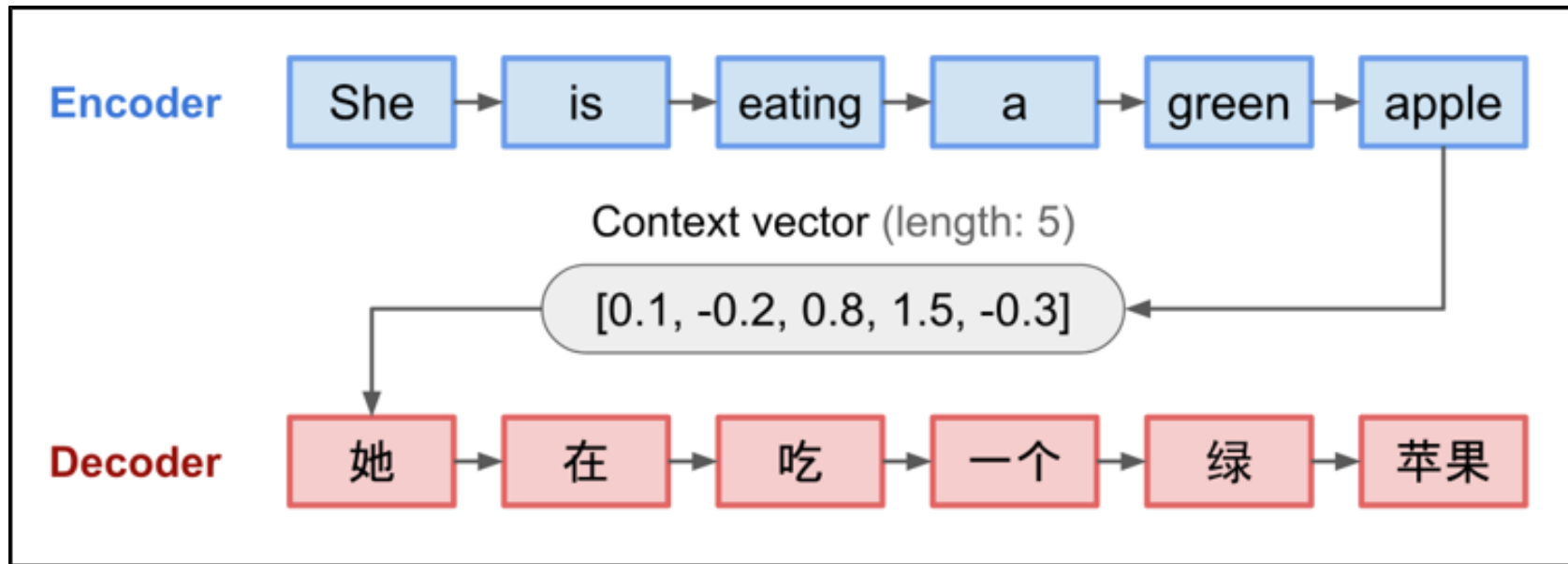
Oriol Vinyals  
Google  
vinyals@google.com

Quoc V. Le  
Google  
qvl@google.com

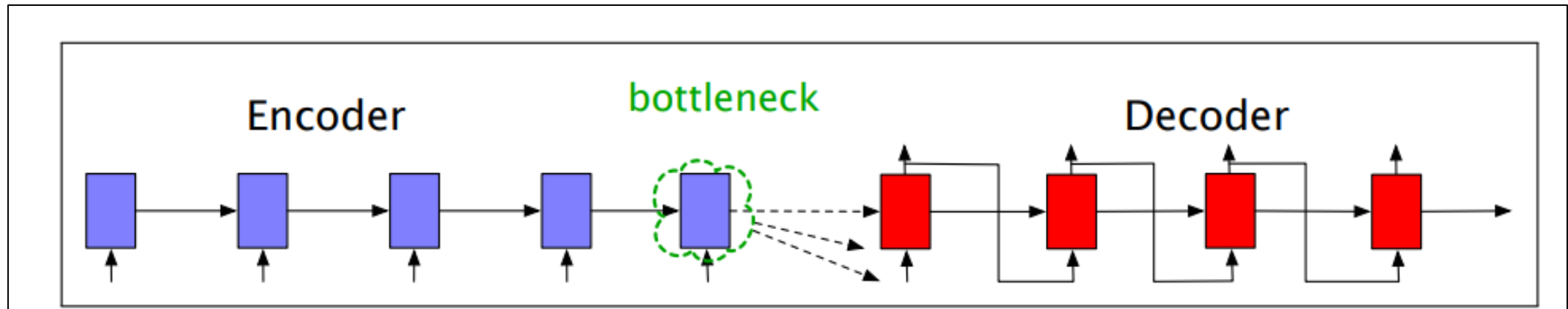
## Abstract

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-

# 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

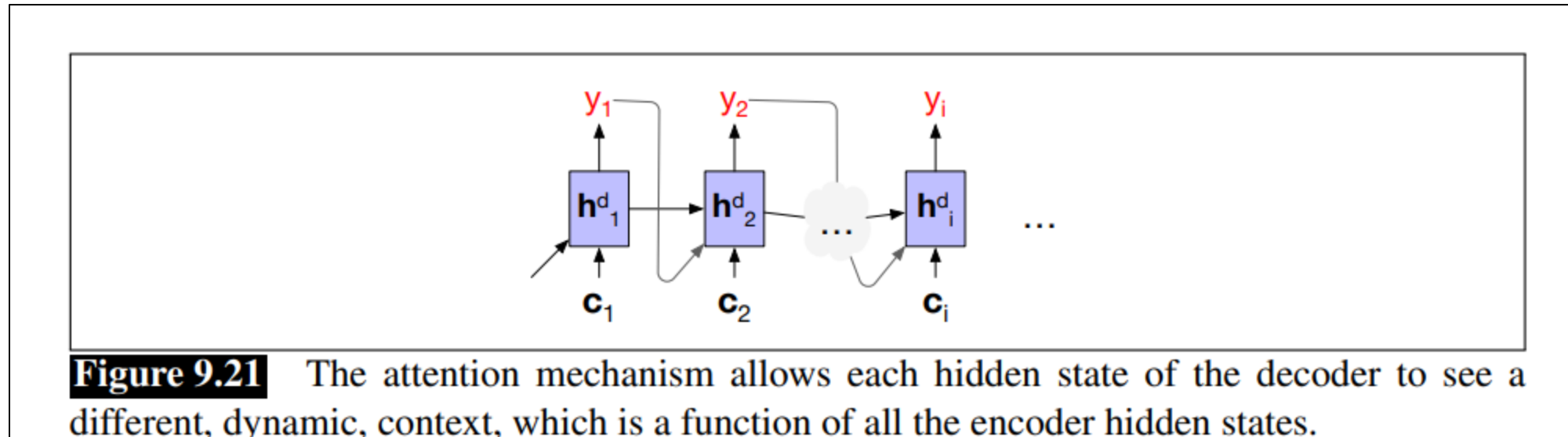
**Imagine the whole universe in all its beauty - try to visualize everything you can find there and how you can describe it in words. Then imagine all of it is compressed into a single vector of size e.g. 512. Do you feel that the universe is still ok?**

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.

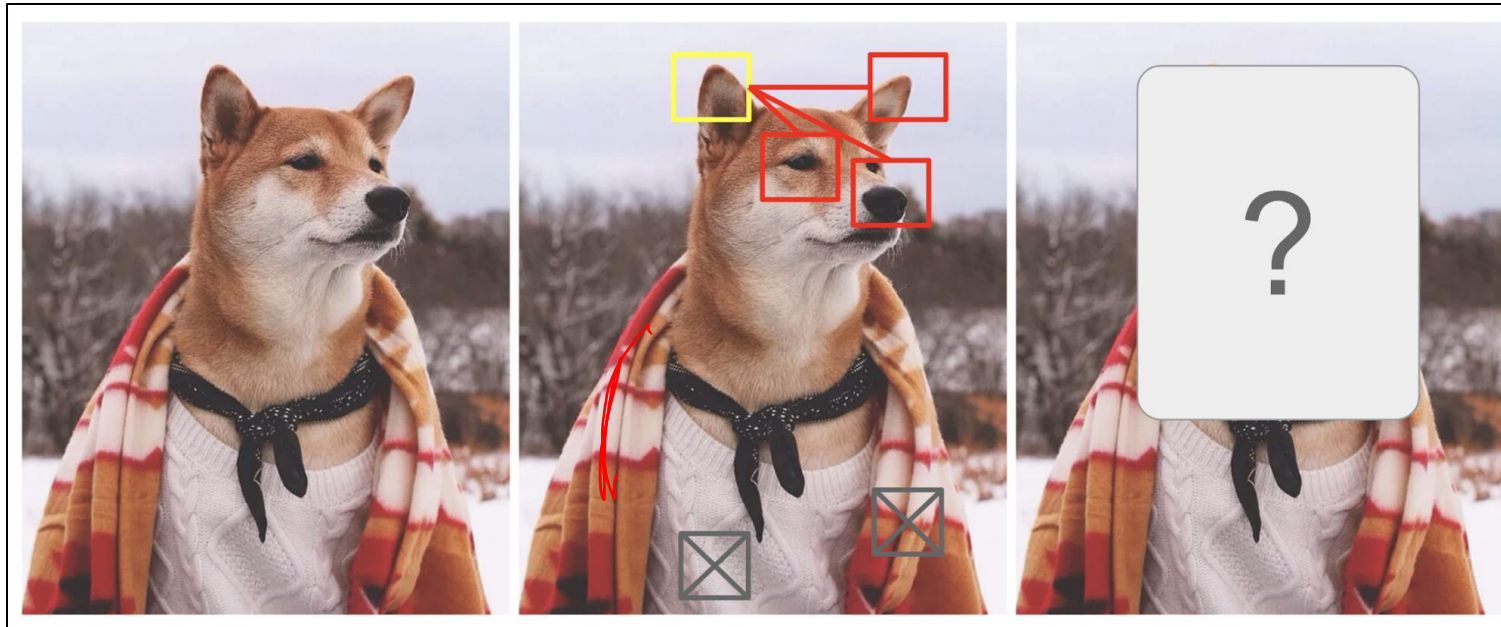
# Solution to bottleneck problem: Attention

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



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

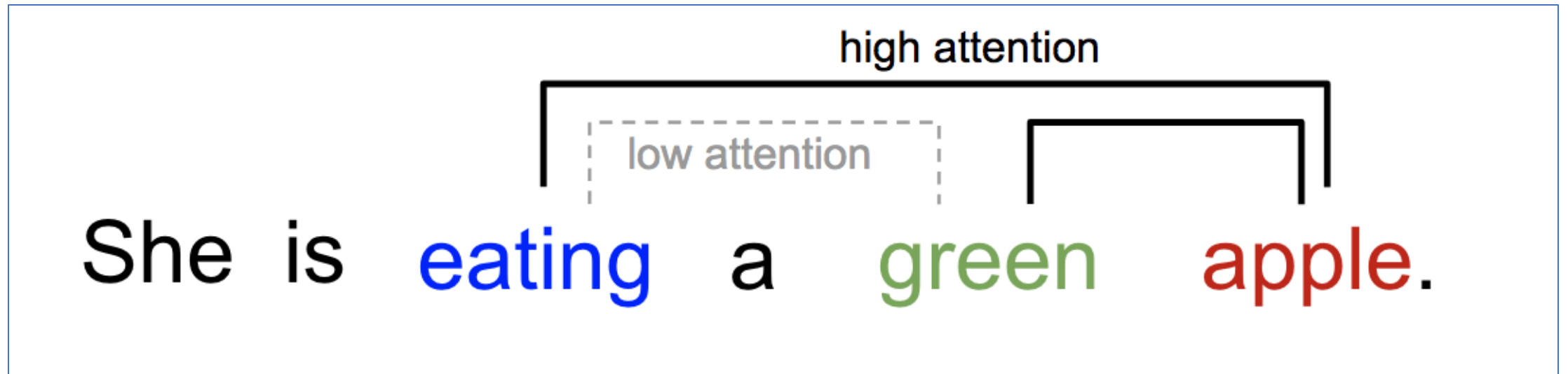
# Attention

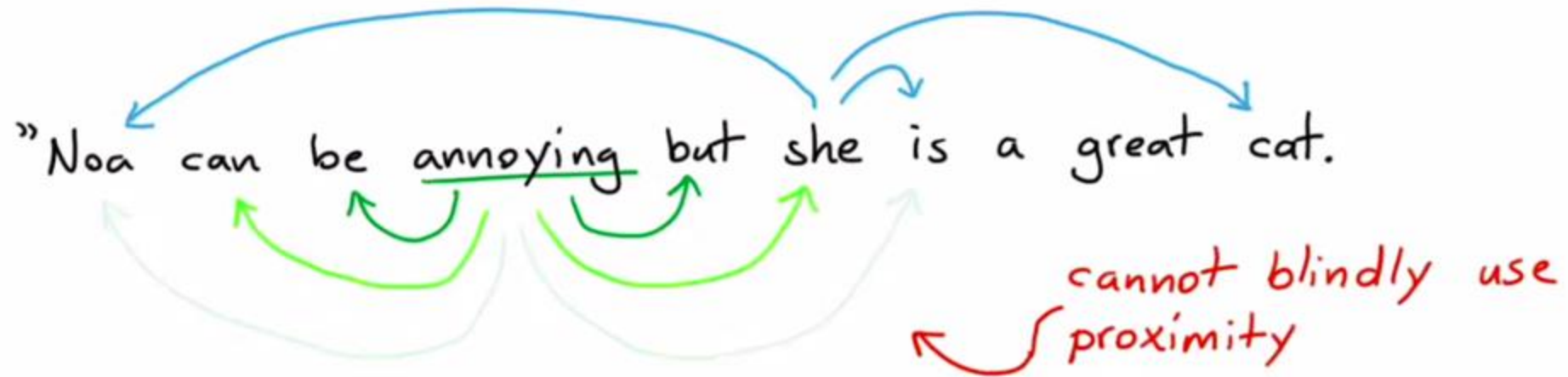


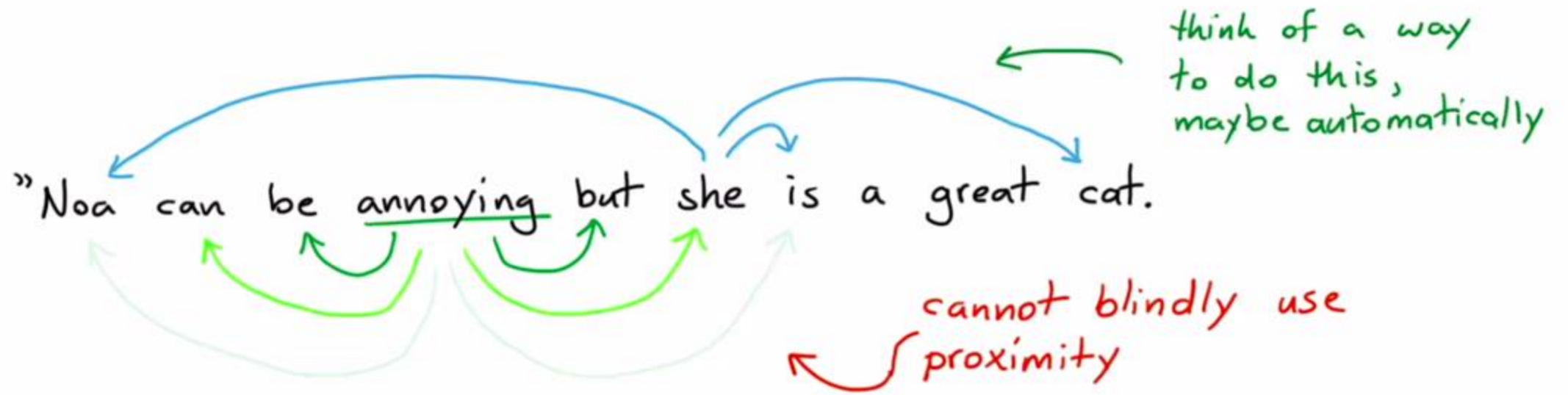
*A Shiba Inu in a men's outfit. The credit of the original photo goes to Instagram @mensweardog.*

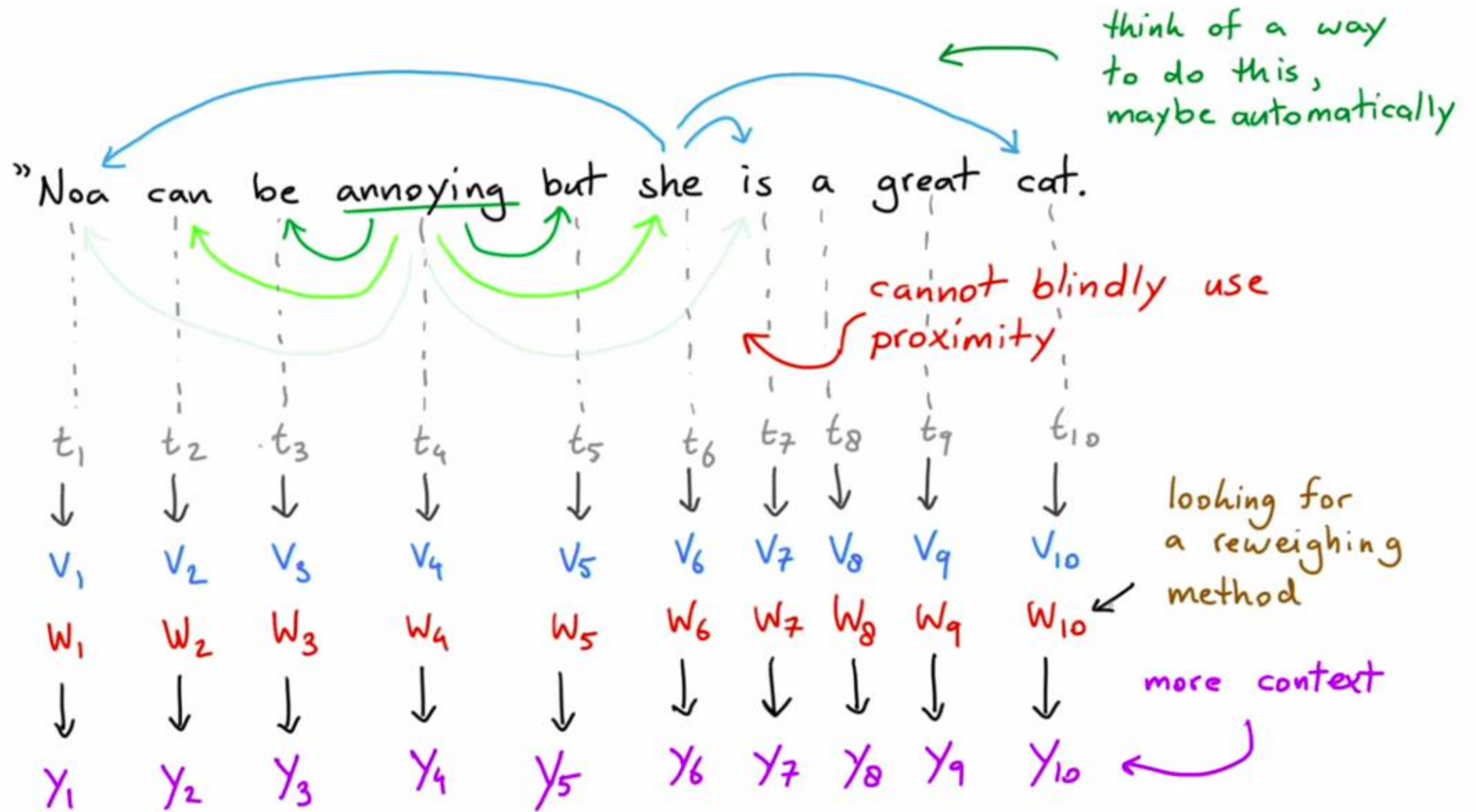
Source: <https://lilianweng.github.io/posts/2018-06-24-attention/>

# Attention

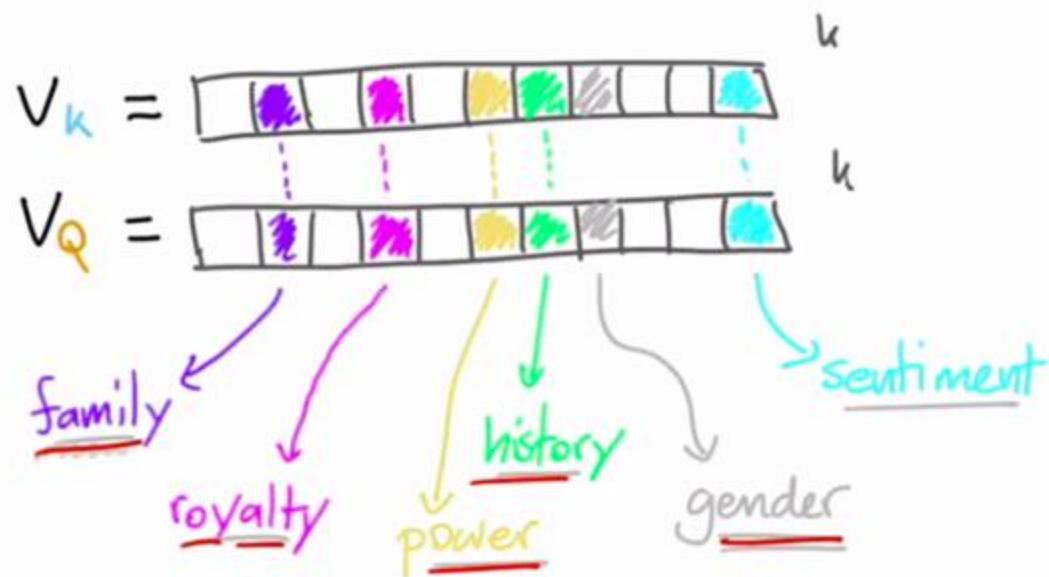












## Reweighting Plan

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

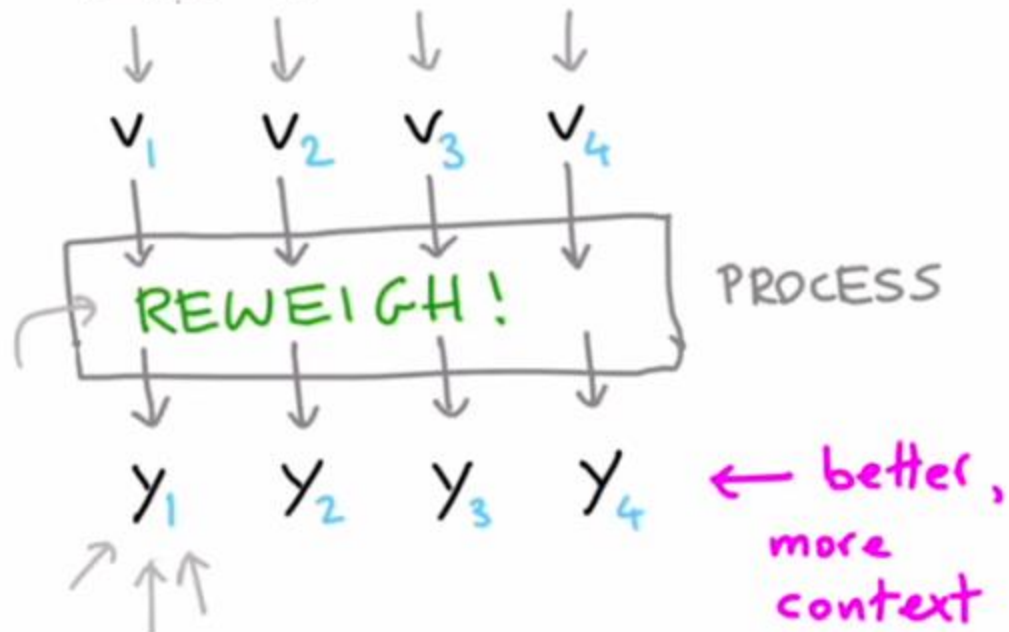
Let's explore this

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!

Bank of the river.

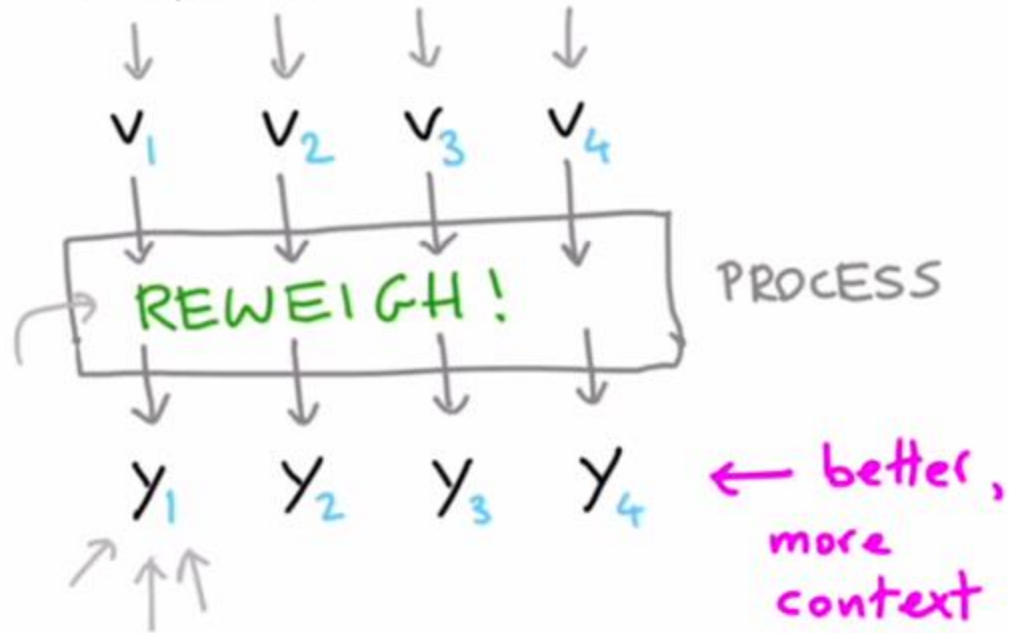


$$\begin{array}{lcl} V_1 V_1 = W_{11} & & W_{11} \\ V_1 V_2 = W_{12} & \xrightarrow{\text{norm}} & W_{12} \\ V_1 V_3 = W_{13} & & W_{13} \\ V_1 V_4 = W_{14} & & W_{14} \end{array}$$

↑  
sum to one

$$W_{11}V_1 + W_{12}V_2 + W_{13}V_3 + W_{14}V_4 = \gamma_1$$

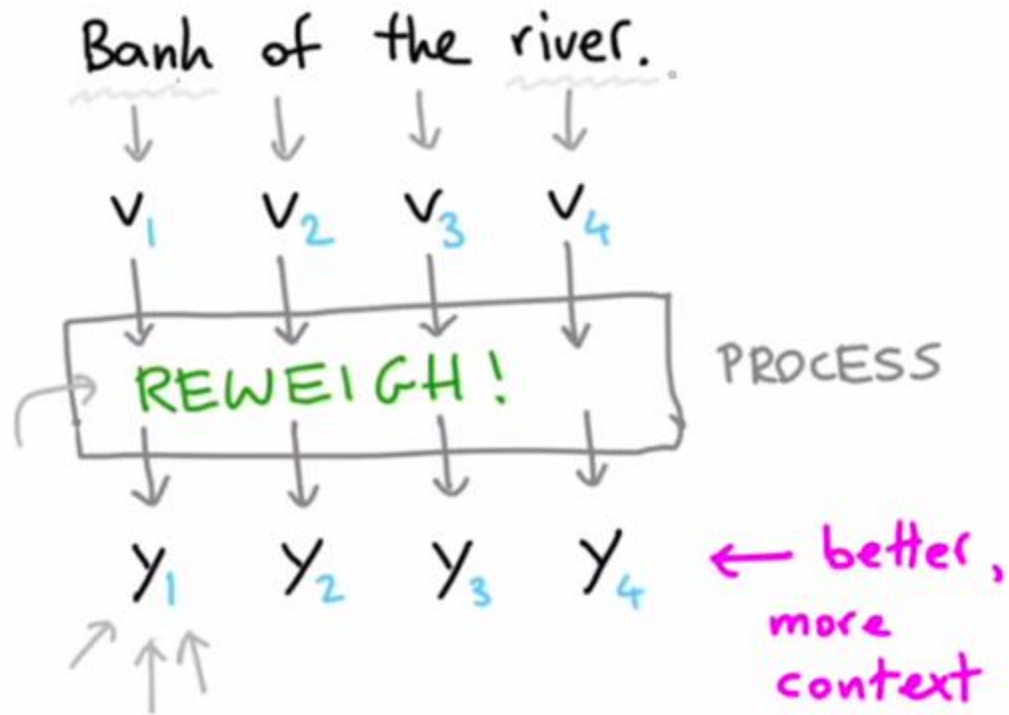
Bank of the river.



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

↑  
sum to one

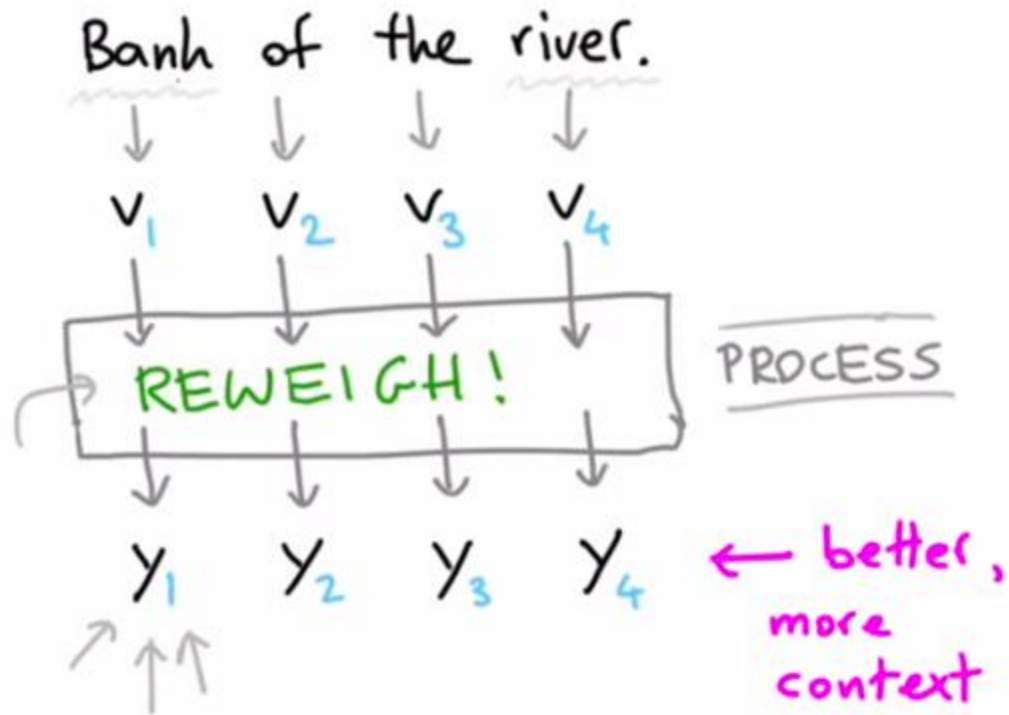
$$w_{11}v_1 + w_{12}v_2 + w_{13}v_3 + w_{14}v_4 = y_1$$



$$\begin{array}{lcl}
 v_1 v_1 = w_{11} & & w_{11} \\
 v_1 v_2 = w_{12} & \xrightarrow{\text{norm}} & w_{12} \\
 v_1 v_3 = w_{13} & & w_{13} \\
 v_1 v_4 = w_{14} & & w_{14} \\
 & & \uparrow \\
 & \nwarrow \text{sum to one} & 
 \end{array}$$

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

↑ ↑ ↑ ↑  
reweigh all vectors towards  $v_1$

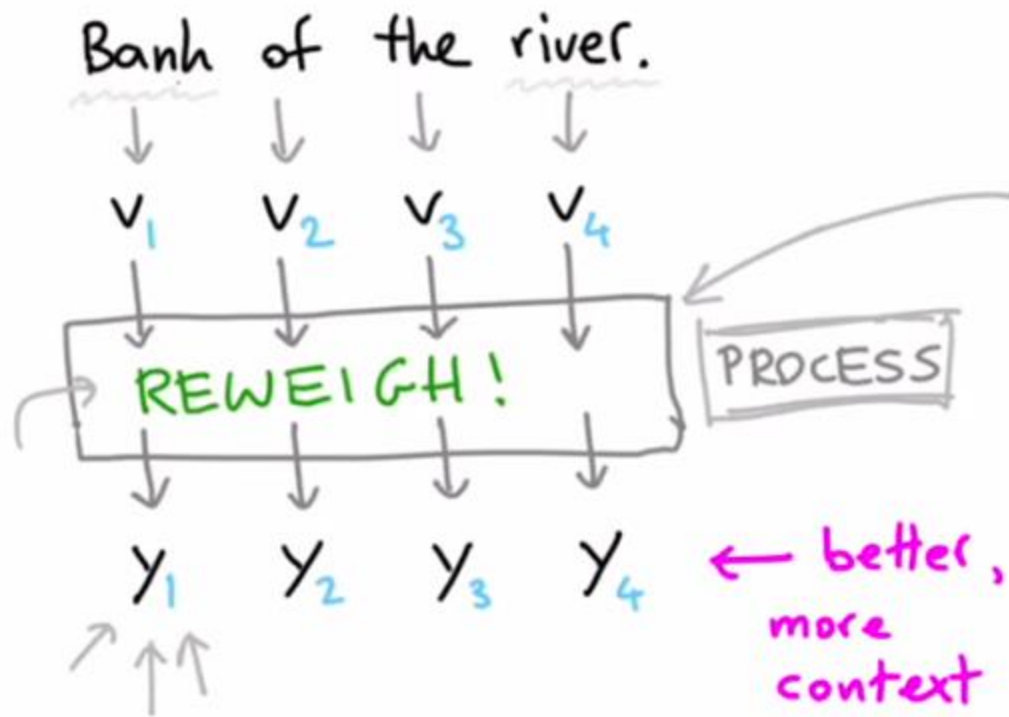


$$\begin{aligned}
 v_1 v_1 &= w_{11} \\
 v_1 v_2 &= w_{12} \\
 v_1 v_3 &= w_{13} \\
 v_1 v_4 &= w_{14}
 \end{aligned}
 \xrightarrow{\text{norm}}
 \begin{aligned}
 w_{11} \\
 w_{12} \\
 w_{13} \\
 w_{14}
 \end{aligned}$$

↑  
sum to one

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





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

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

SELF ATTENTION

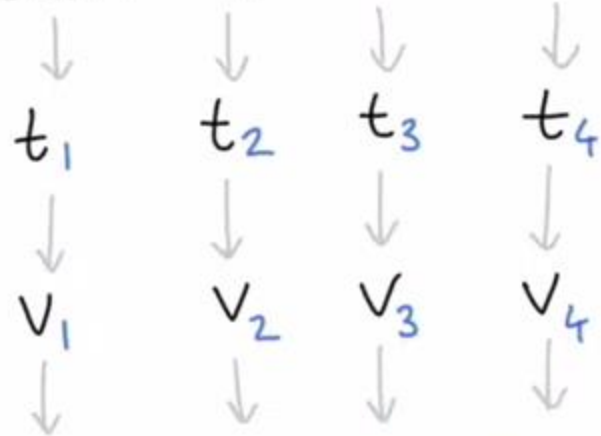
$$\begin{aligned}
 v_1 v_1 &= w_{11} \\
 v_1 v_2 &= w_{12} \\
 v_1 v_3 &= w_{13} \\
 v_1 v_4 &= w_{14}
 \end{aligned}$$

norm

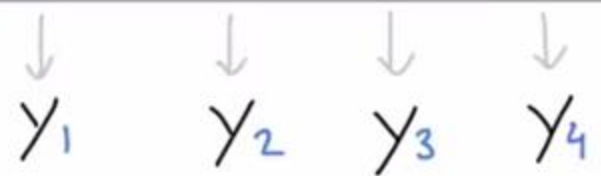
$$\begin{aligned}
 w_{11} \\
 w_{12} \\
 w_{13} \\
 w_{14}
 \end{aligned}$$

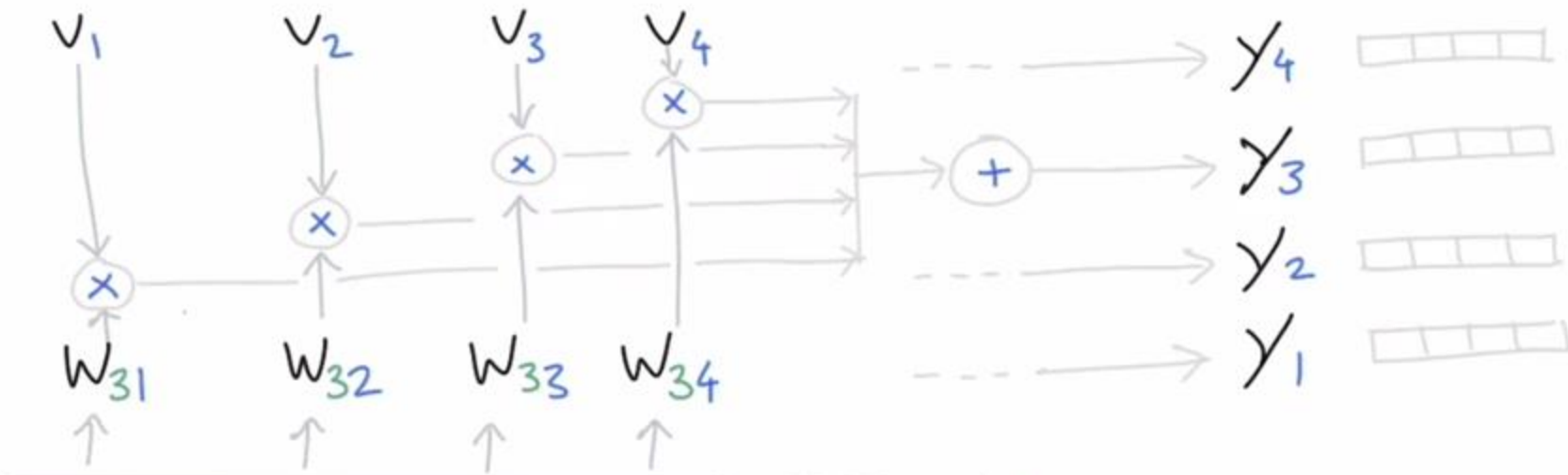
sum to one

Bank of the river.



ATTENTION





NORMALISE

$$\sum_j w_{3j} = 1$$

$s_{31}$   $s_{32}$   $s_{33}$   $s_{34}$

DOT PRODUCT

$v_1$   $v_2$   $v_3$   $v_4$

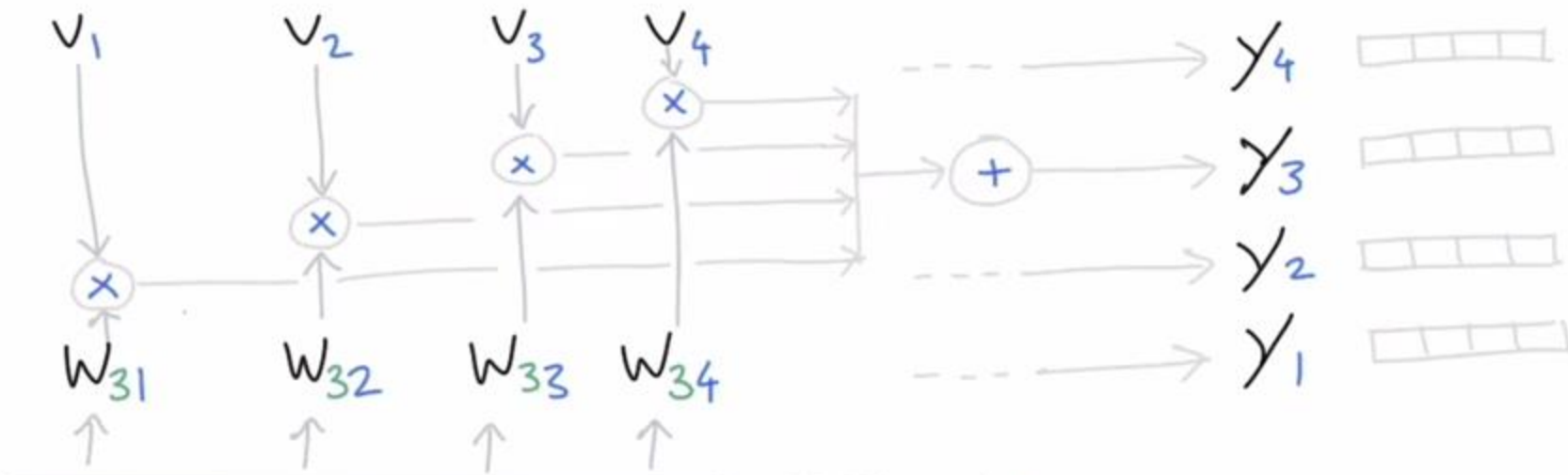
no weights  $\rightarrow$

ATTENTION

$y_1$   $y_2$   $y_3$   $y_4$

$v_1$   $v_2$   $v_3$   $v_4$





NORMALISE

$$\sum_j w_{3j} = 1$$

$s_{31}$   $s_{32}$   $s_{33}$   $s_{34}$

DOT PRODUCT

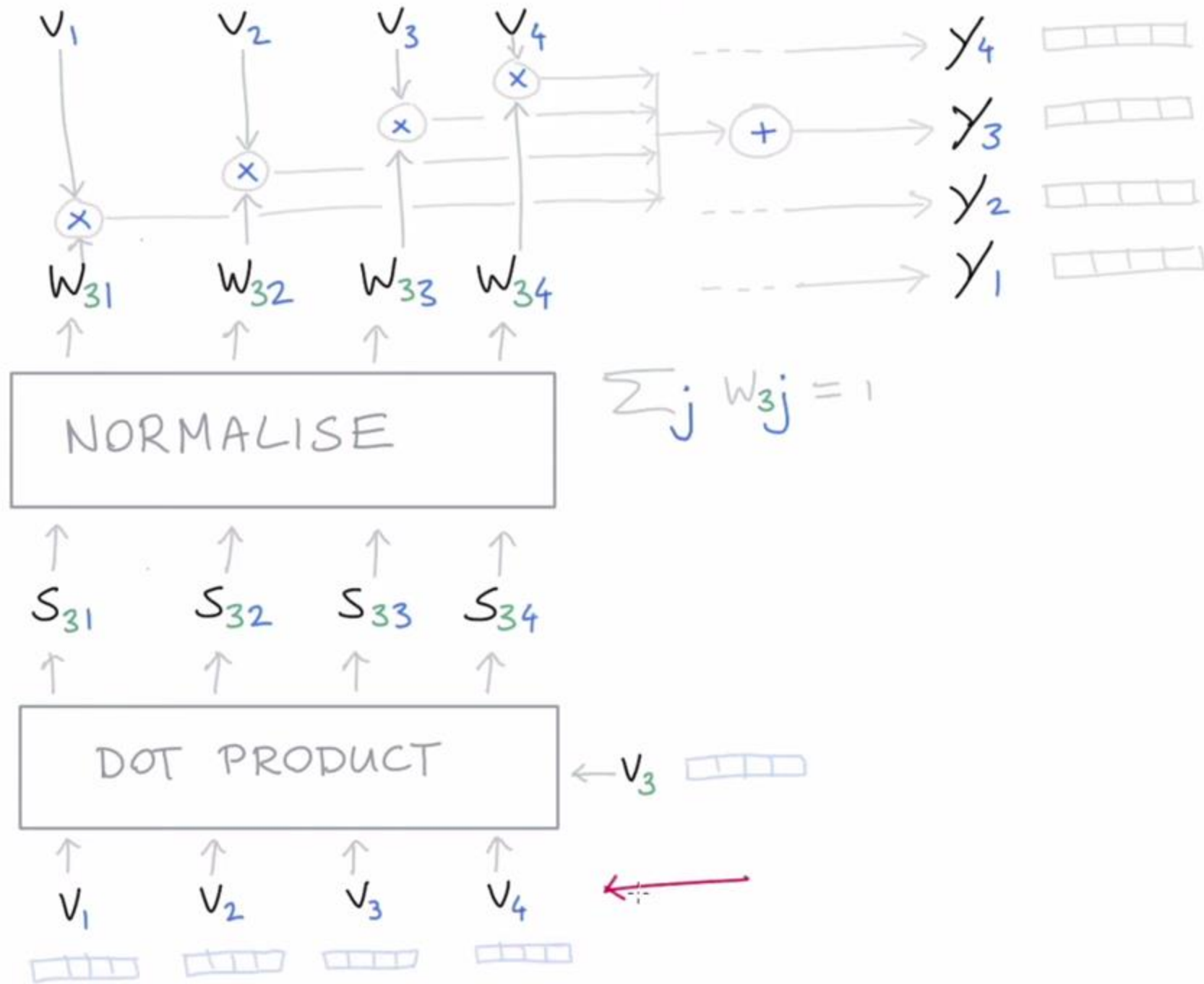
$v_1$   $v_2$   $v_3$   $v_4$

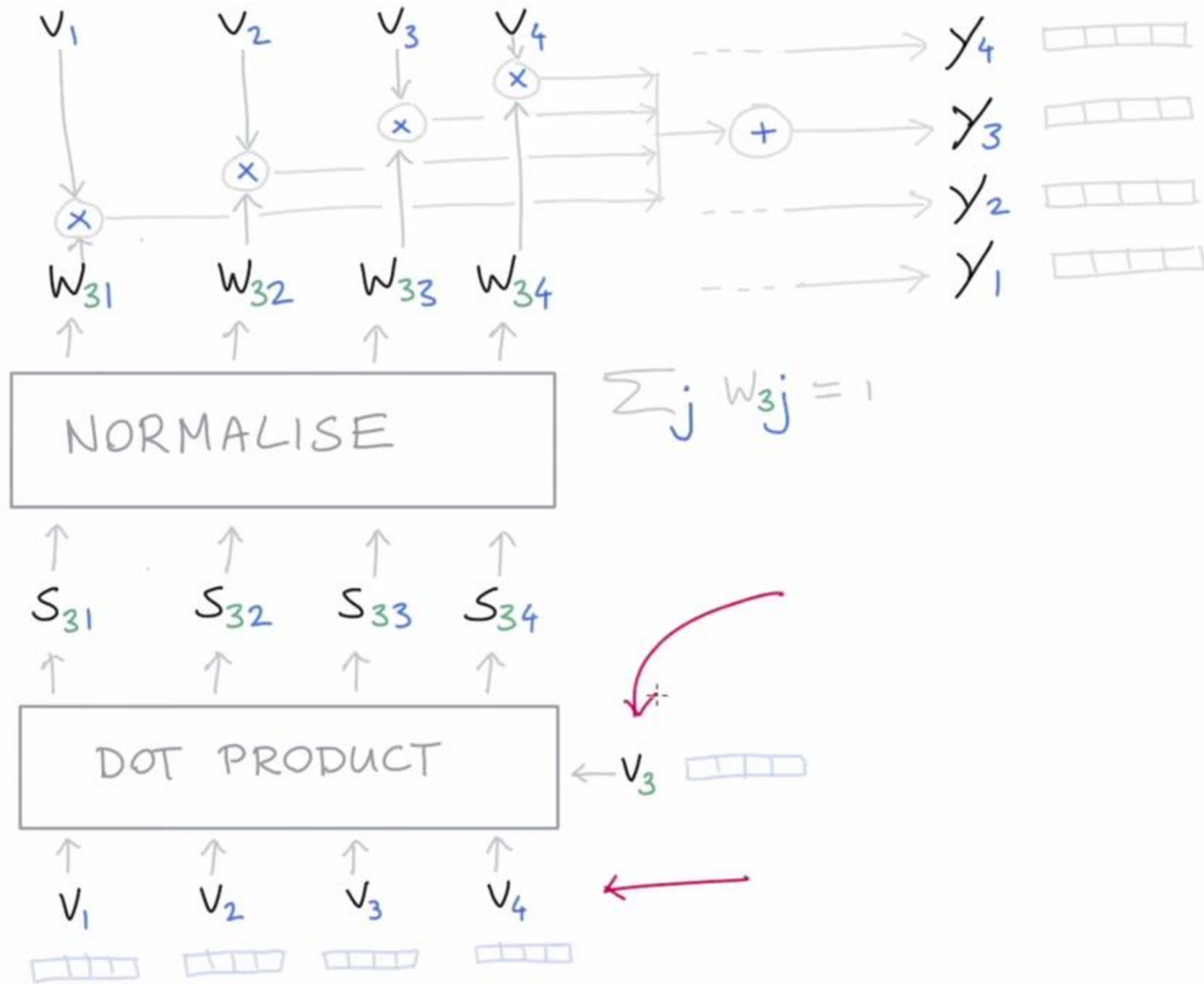
~~no~~ weights →

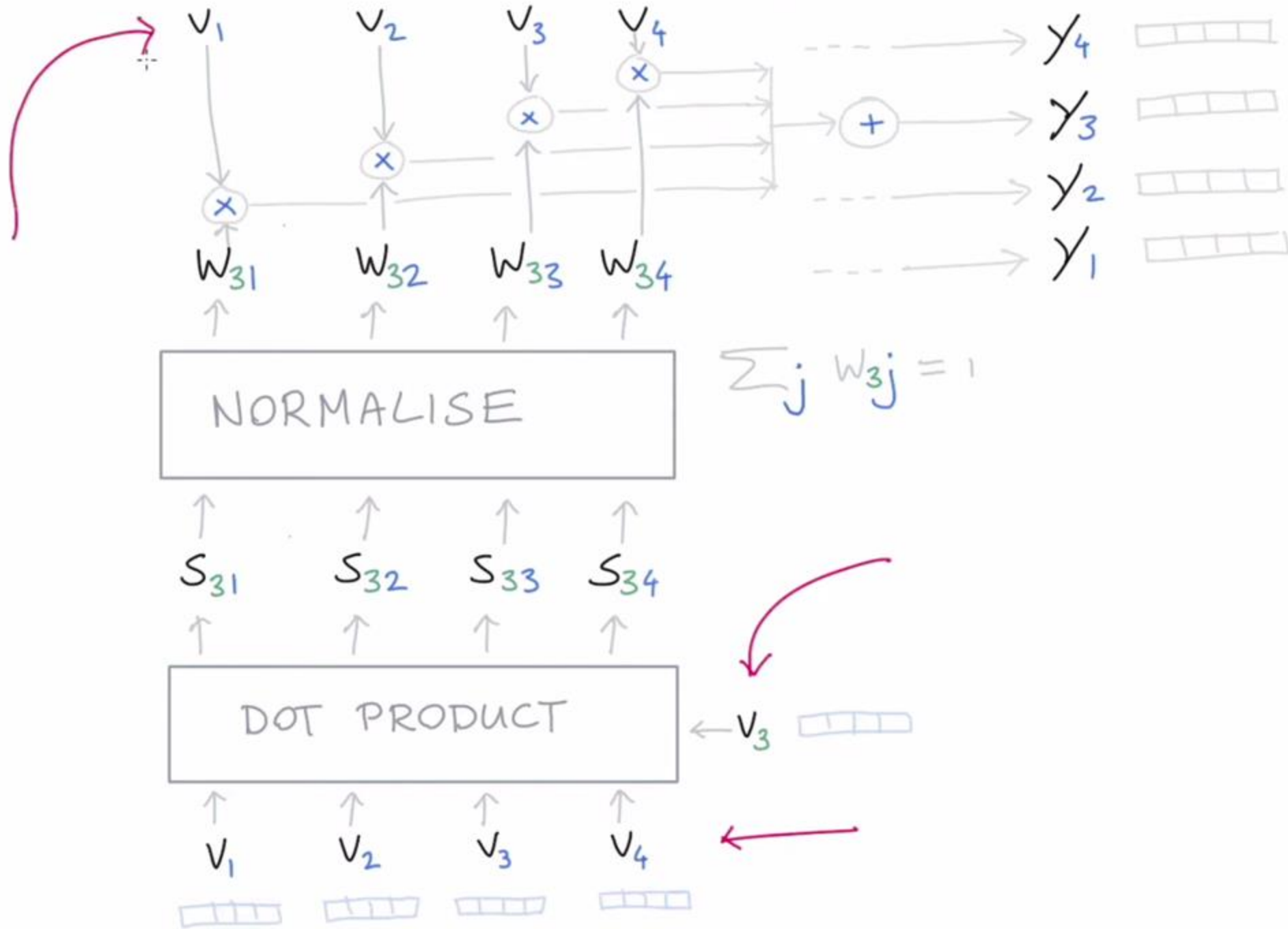
ATTENTION

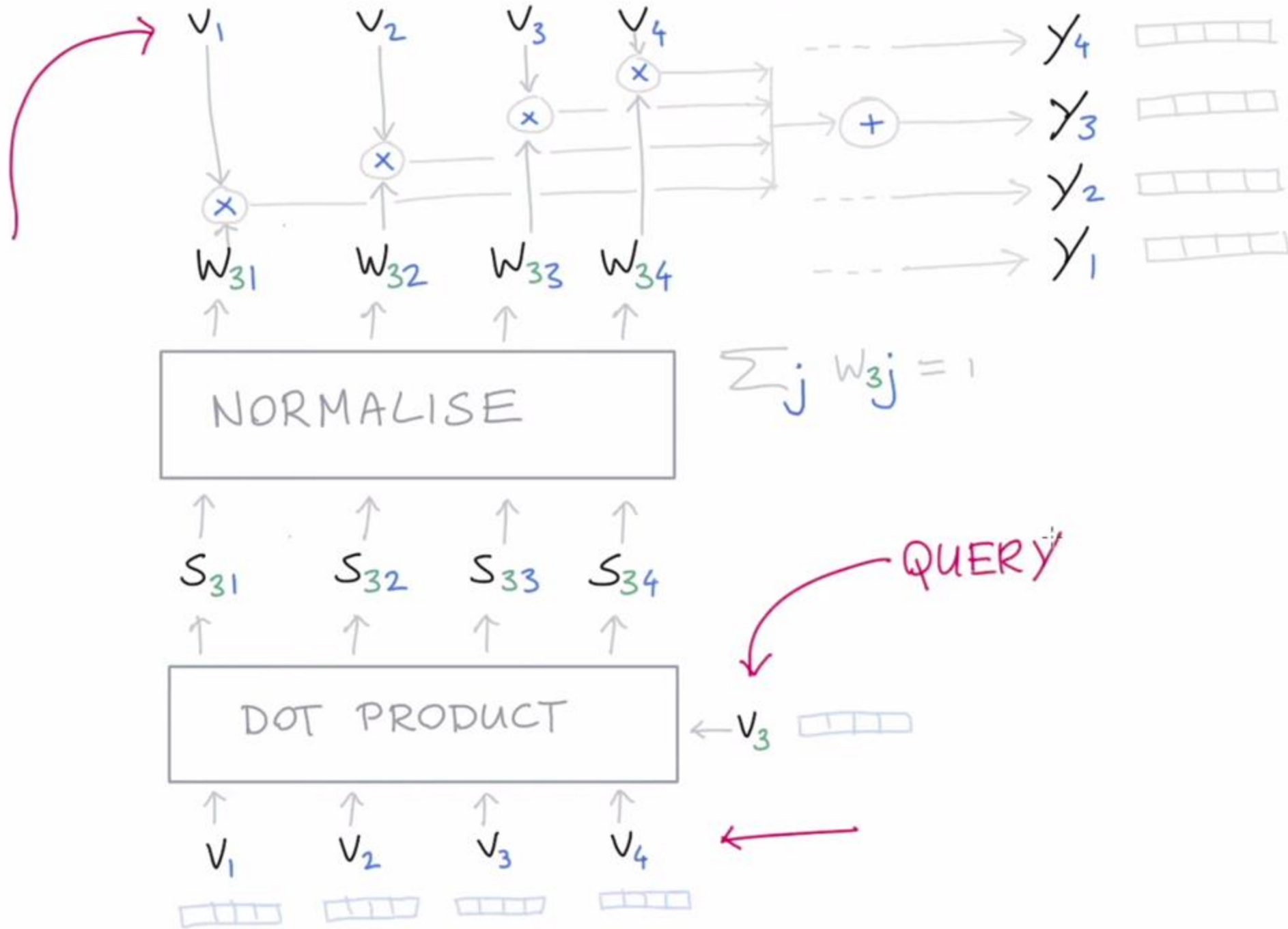
$y_1$   $y_2$   $y_3$   $y_4$

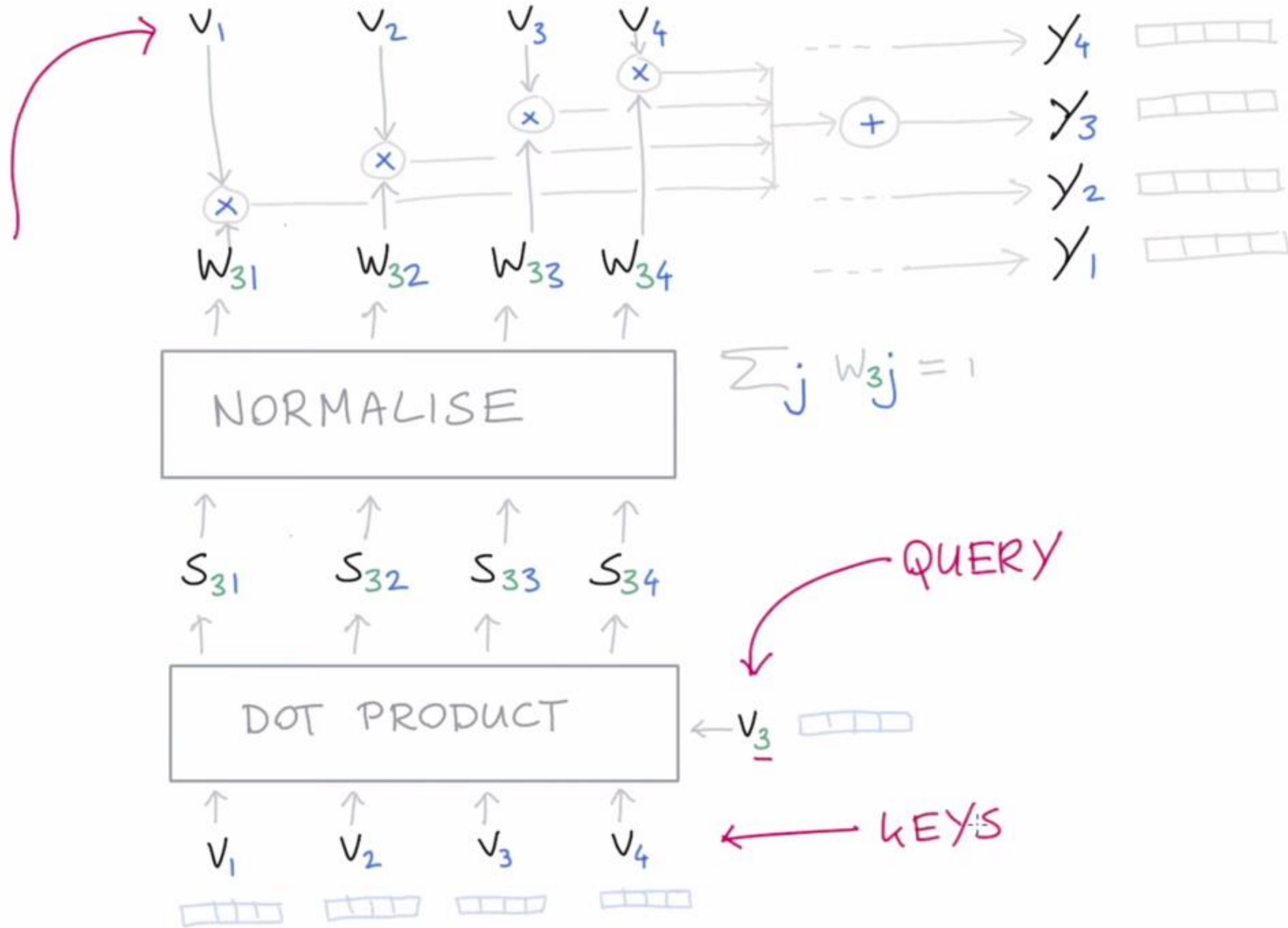
$v_1$   $v_2$   $v_3$   $v_4$



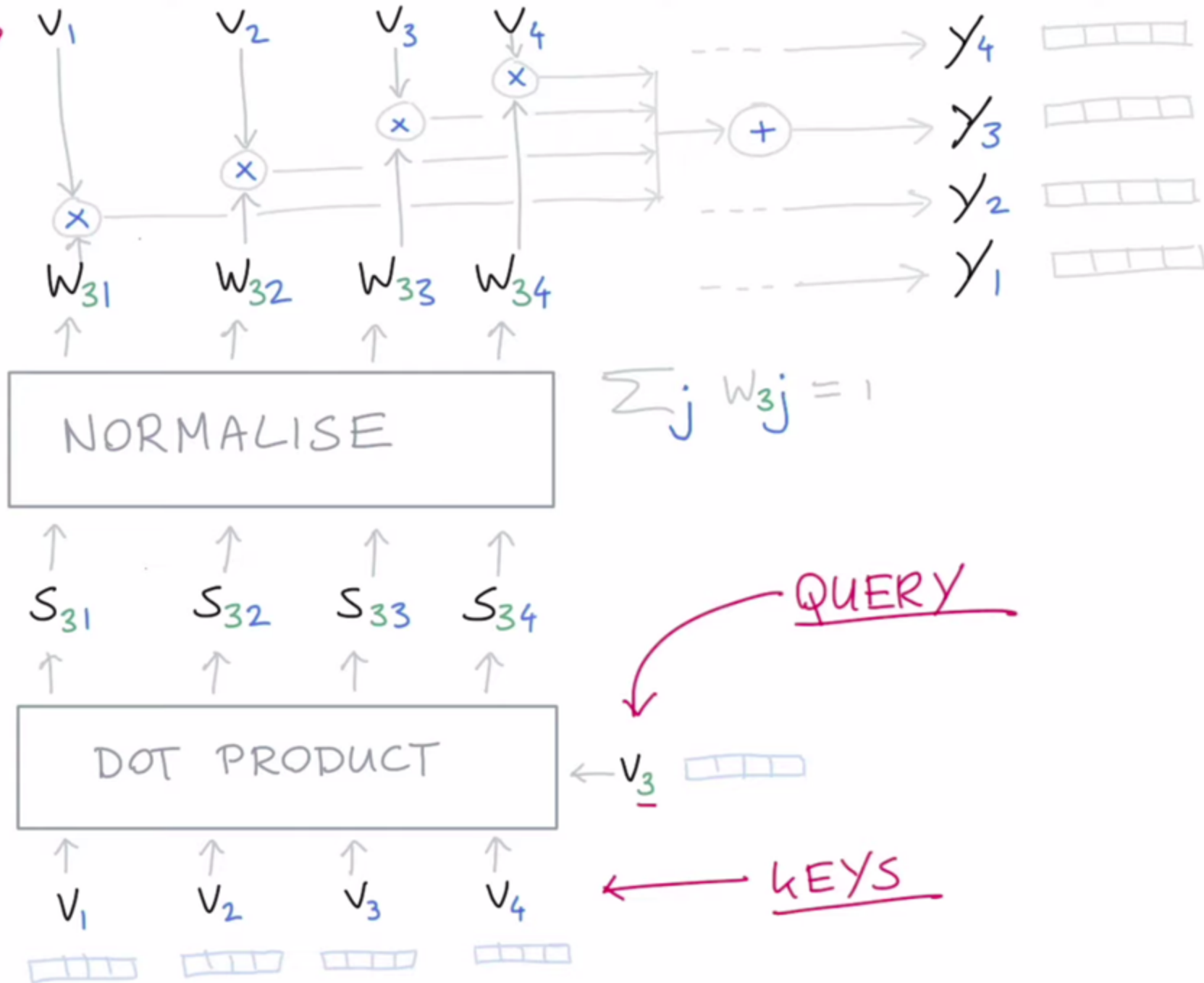








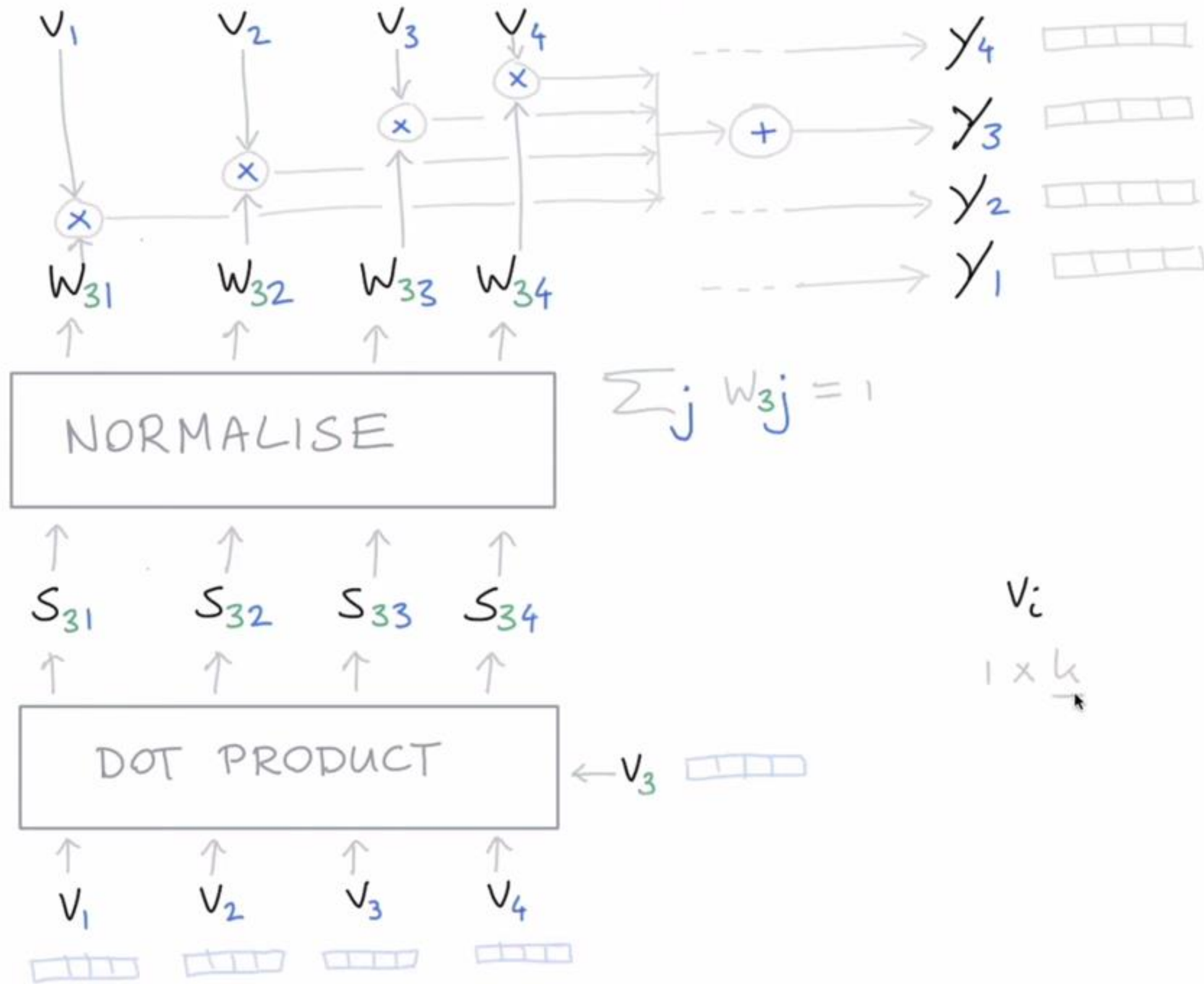
VALUES

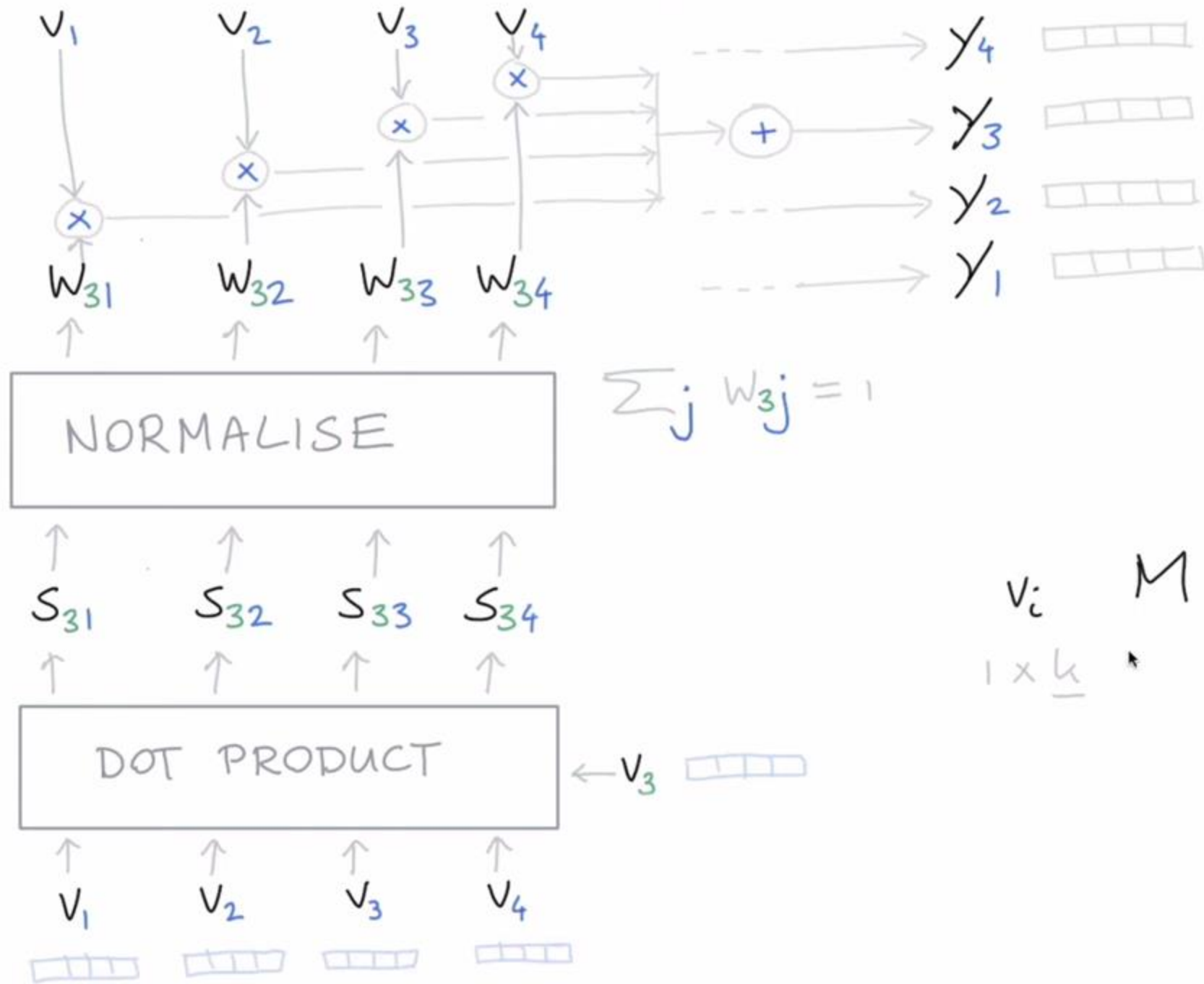


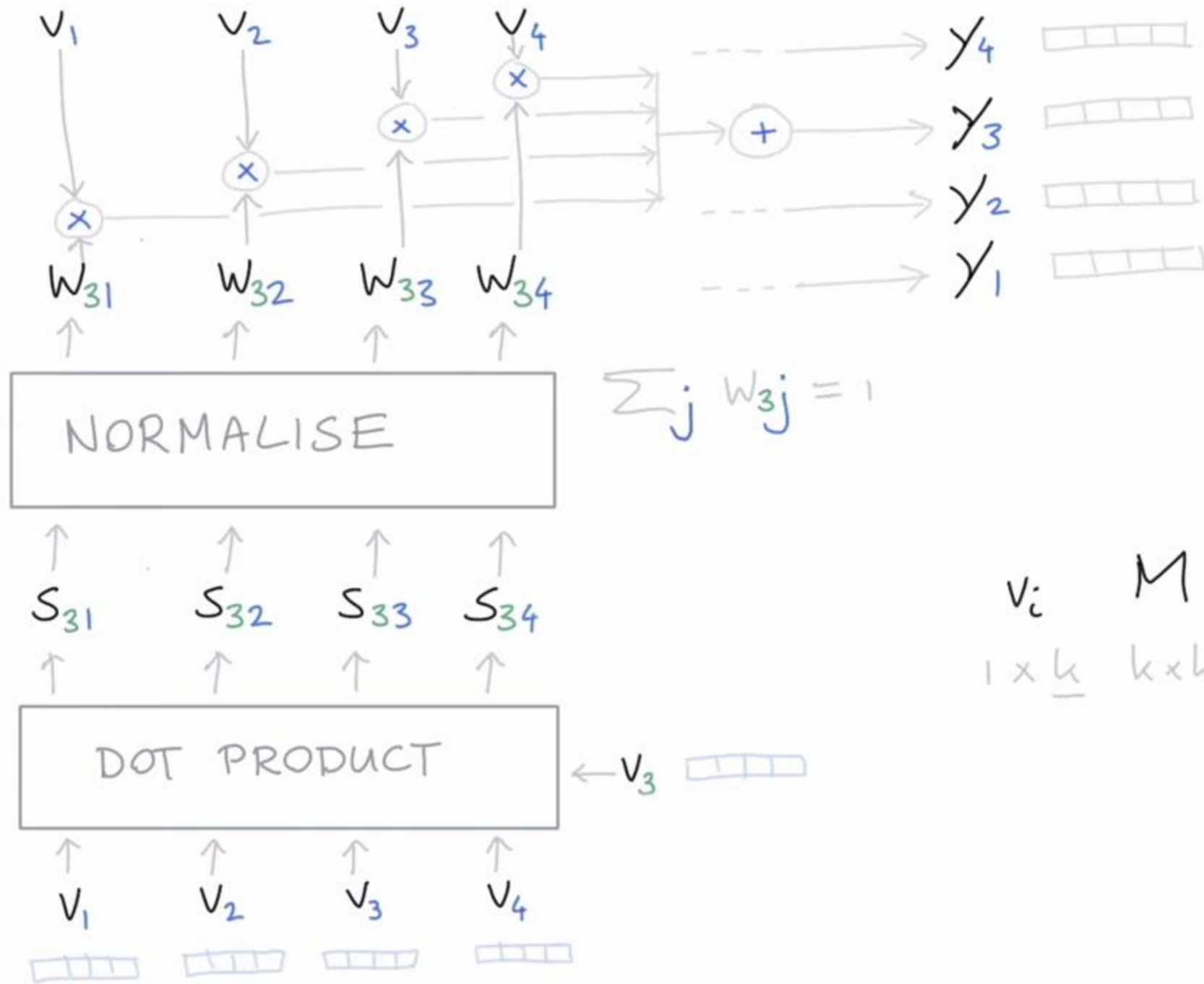
# Key, Value and Query

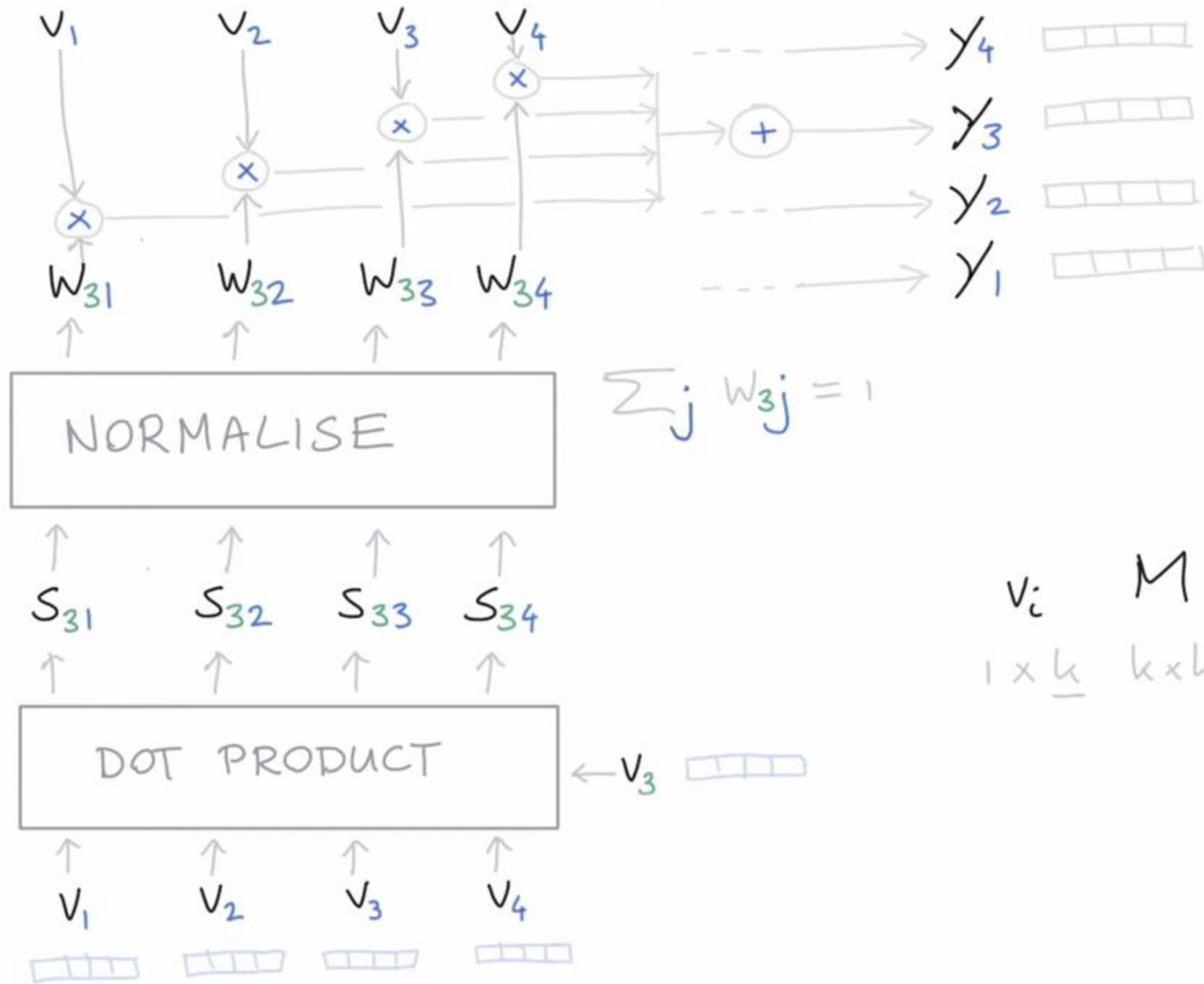
- The major component in the transformer is the unit of **multi-head self-attention mechanism**.

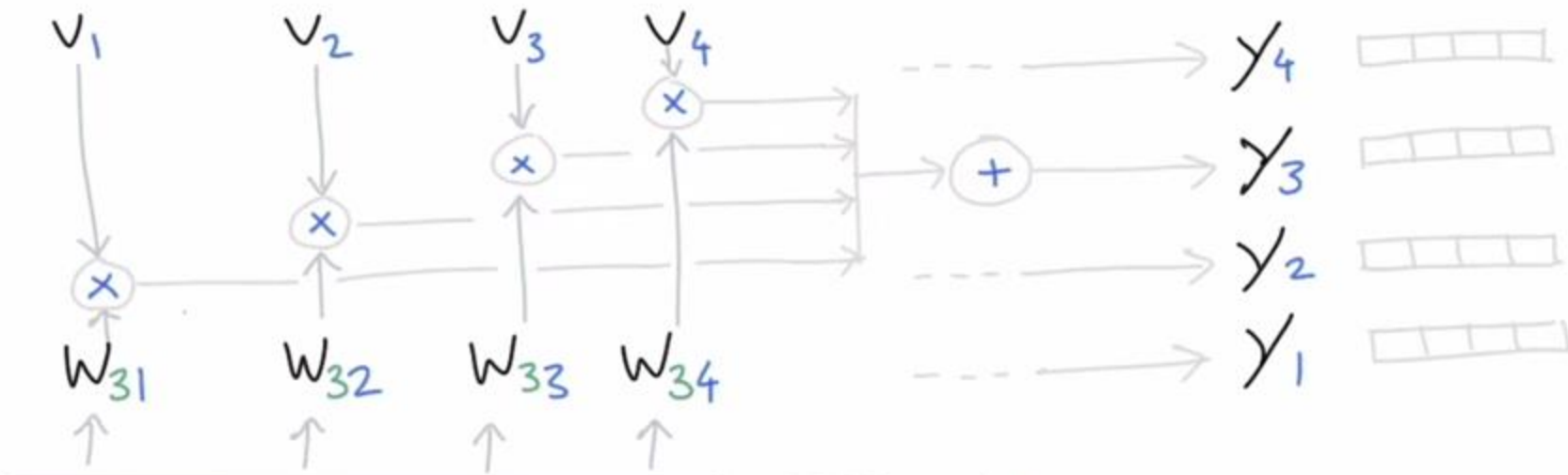






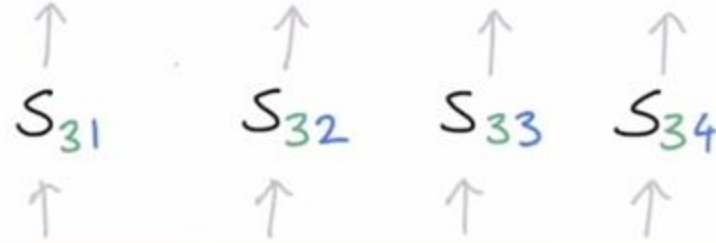




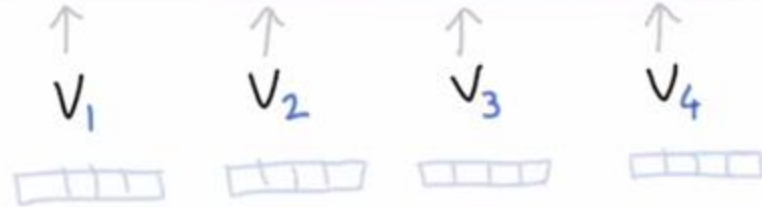


NORMALISE

$$\sum_j W_{3j} = 1$$



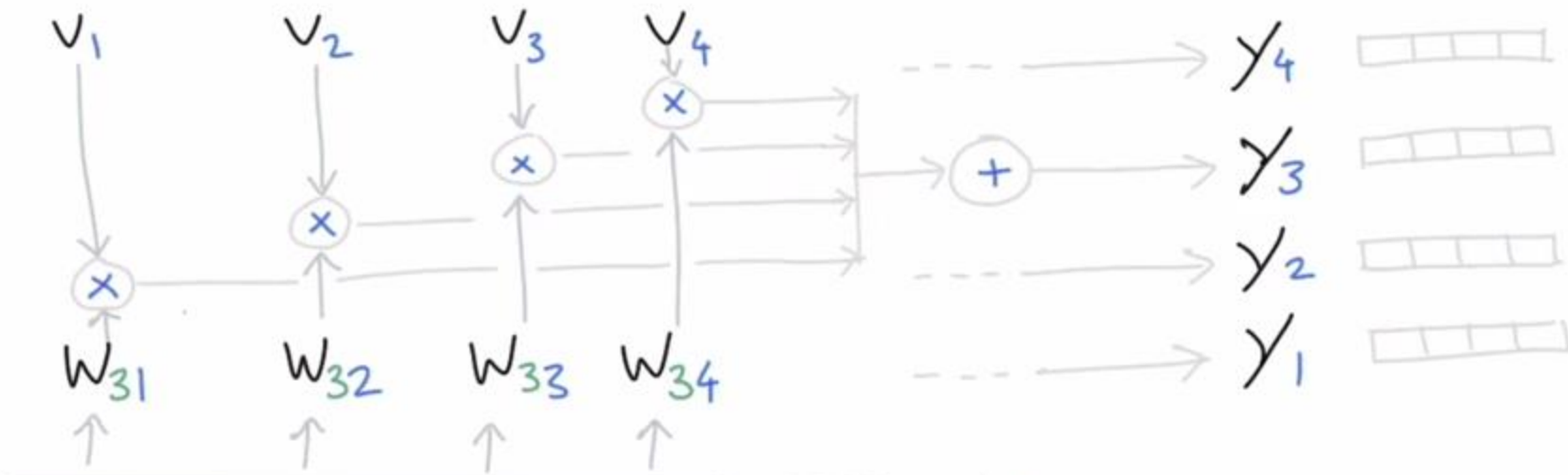
DOT PRODUCT



$$V_i \quad M = \begin{bmatrix} \end{bmatrix}$$

$1 \times k \quad k \times k \quad 1 \times k$





NORMALISE

$$\sum_j W_{3j} = 1$$

$S_{31}, S_{32}, S_{33}, S_{34}$

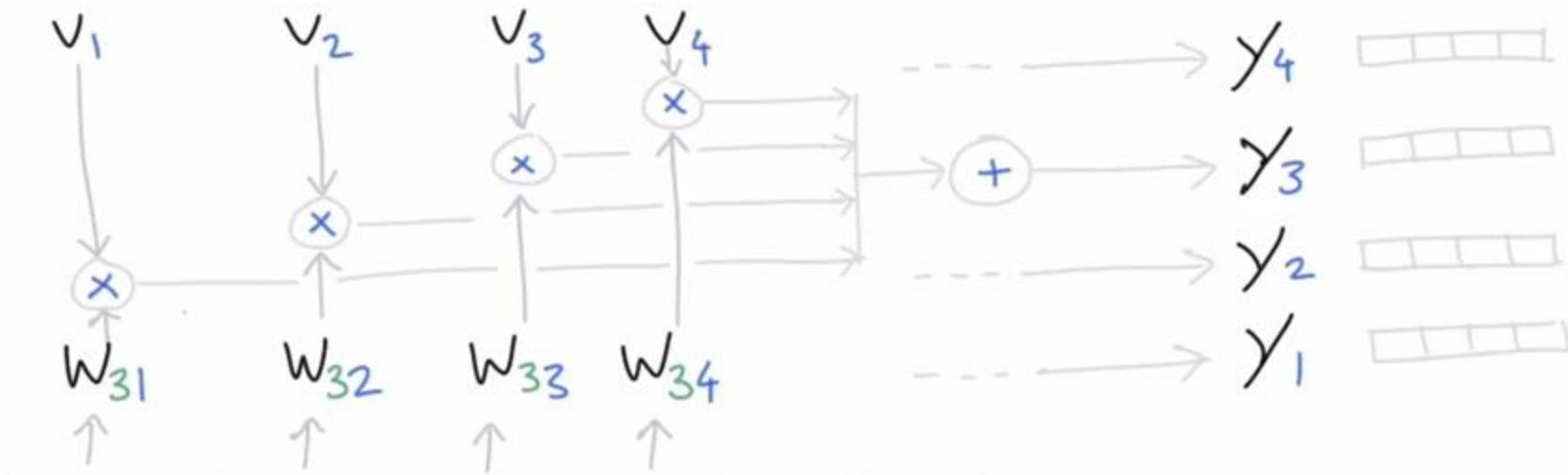
DOT PRODUCT

$V_1 M_k, V_2 M_k, V_3 M_k, V_4 M_k$  KEY

$$V_i \quad M = \begin{bmatrix} \end{bmatrix}$$

$1 \times k \quad k \times k \quad 1 \times k$

$V_3$



NORMALISE

$$\sum_j W_{3j} = 1$$

$S_{31}, S_{32}, S_{33}, S_{34}$

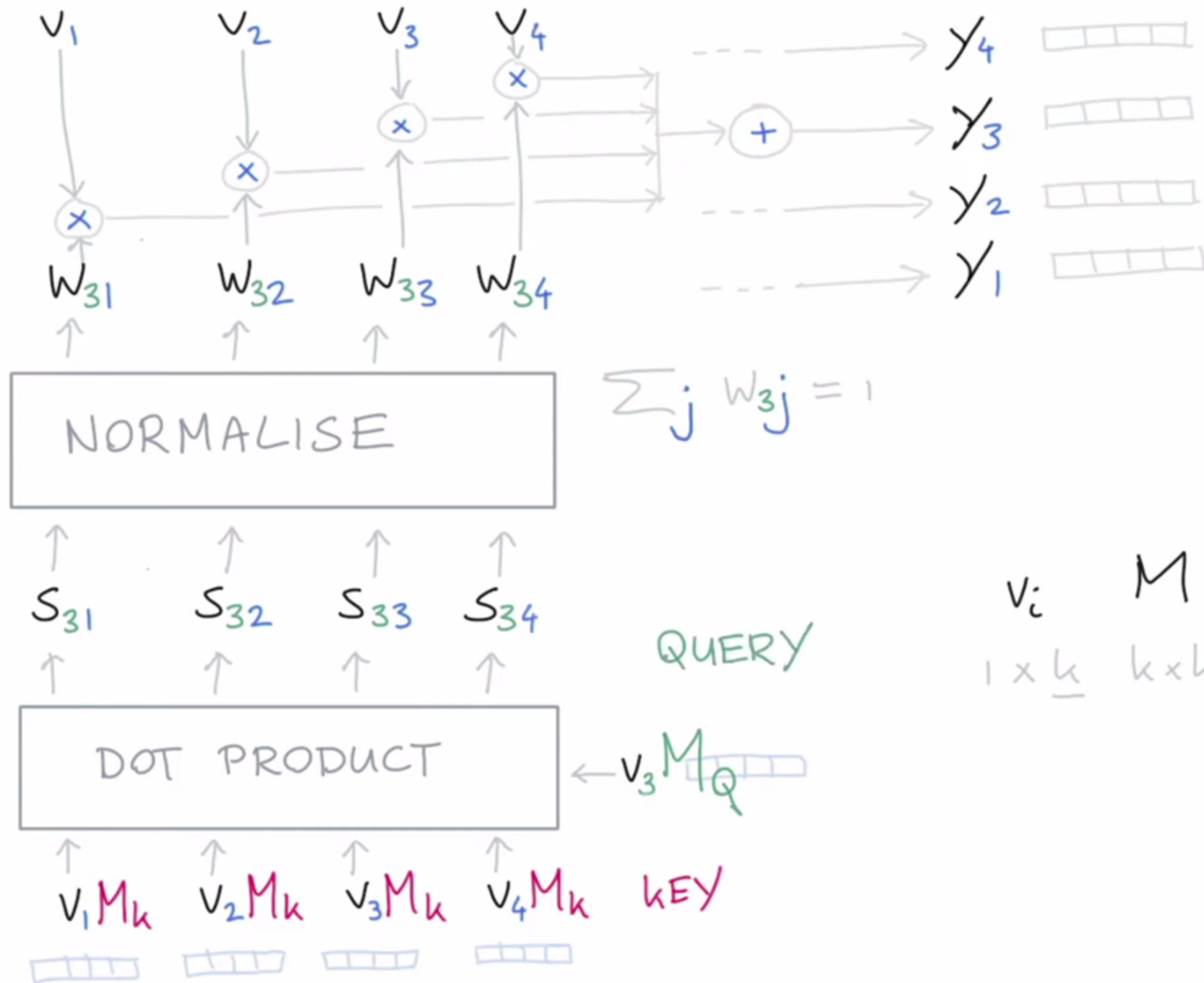
DOT PRODUCT

$V_1 M_k, V_2 M_k, V_3 M_k, V_4 M_k$  KEY

$$V_i \quad M = \begin{bmatrix} \end{bmatrix}$$

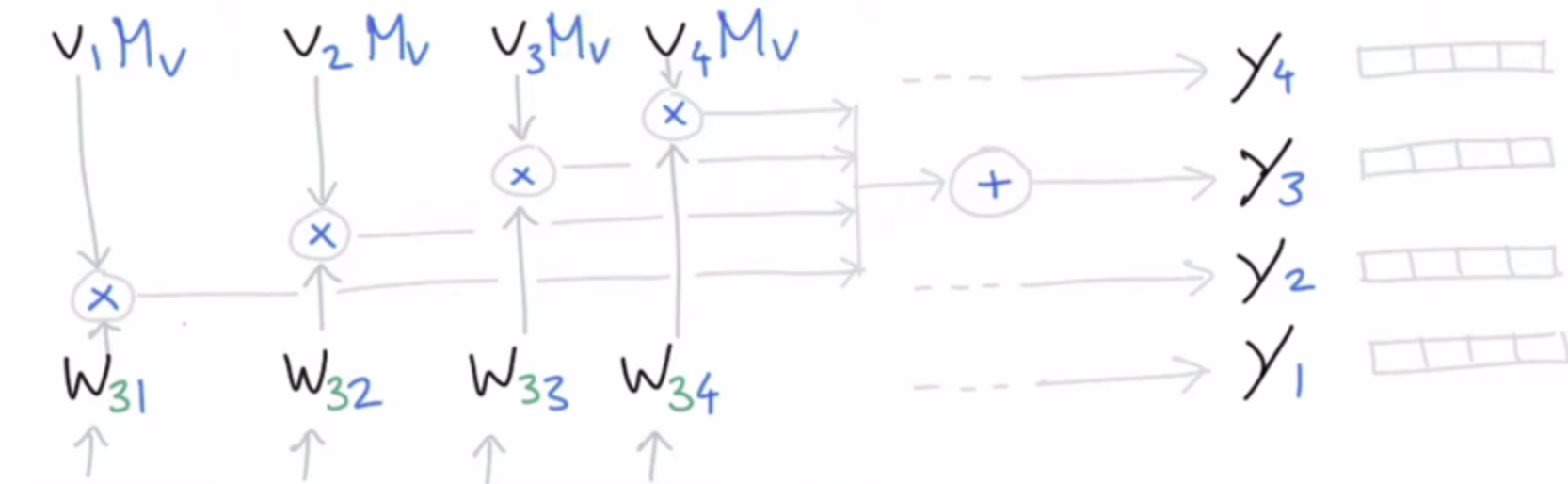
$1 \times k \quad k \times k \quad 1 \times k$

$V_3$





VALUE



NORMALISE

$$\sum_j w_{3j} = 1$$

$s_{31}$   $s_{32}$   $s_{33}$   $s_{34}$

DOT PRODUCT

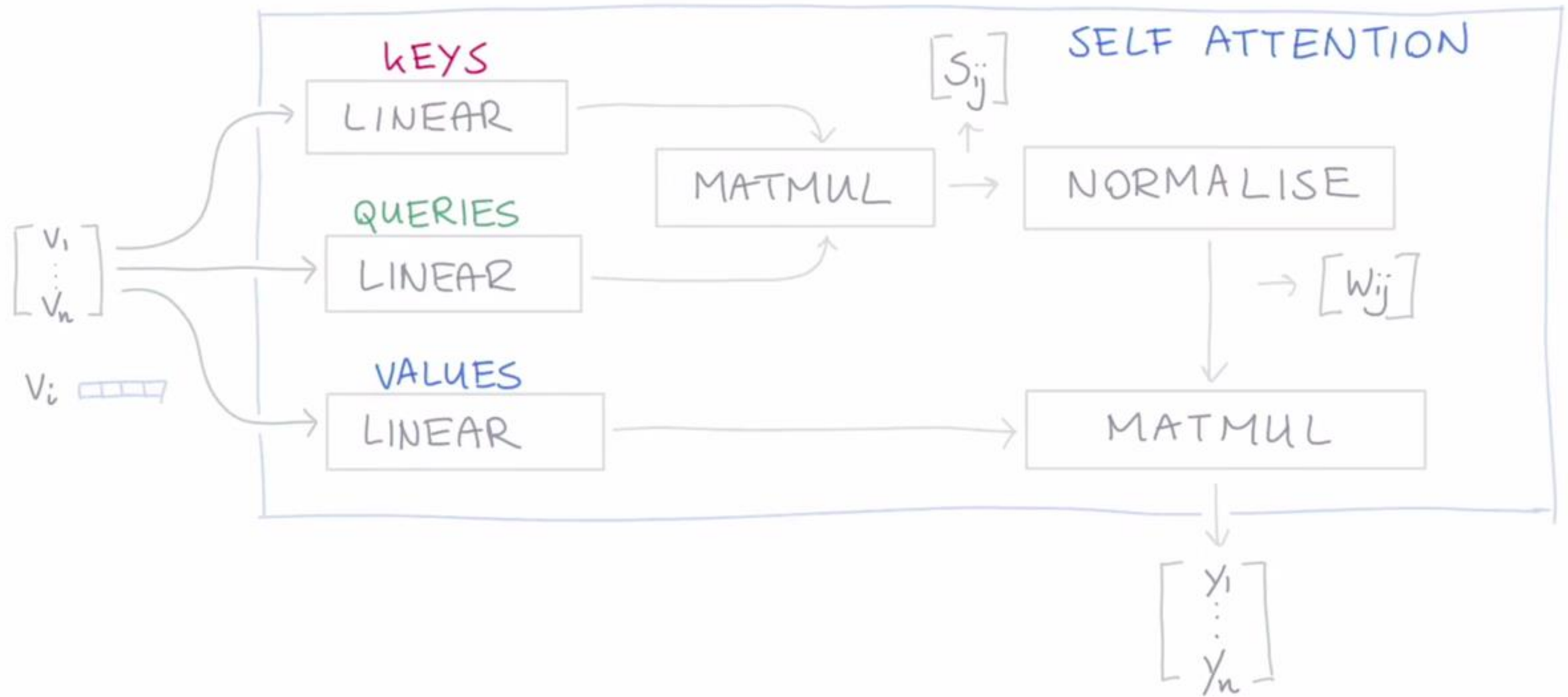
QUERY

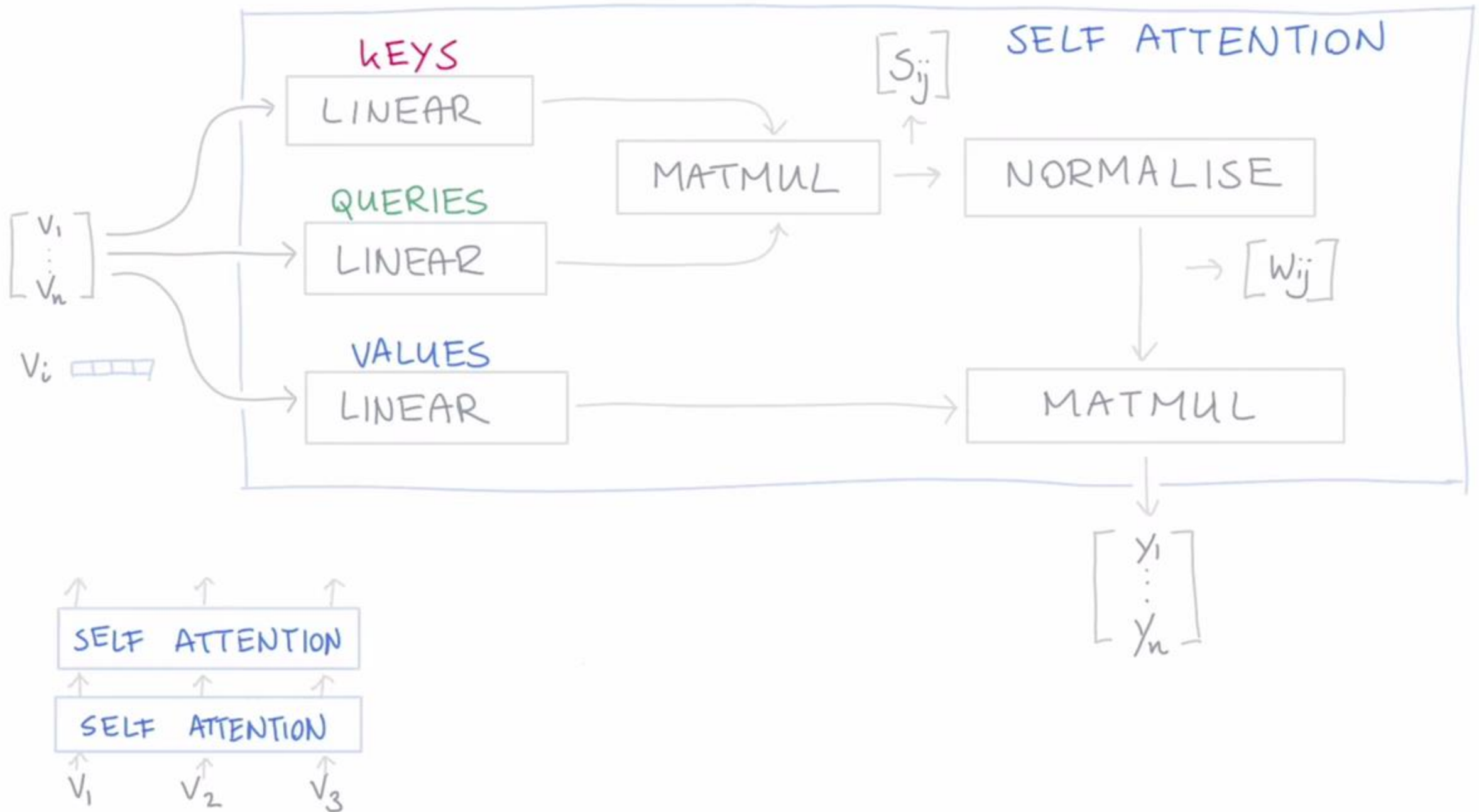
$$v_i \quad M = \begin{bmatrix} & \\ & \\ & \end{bmatrix}$$

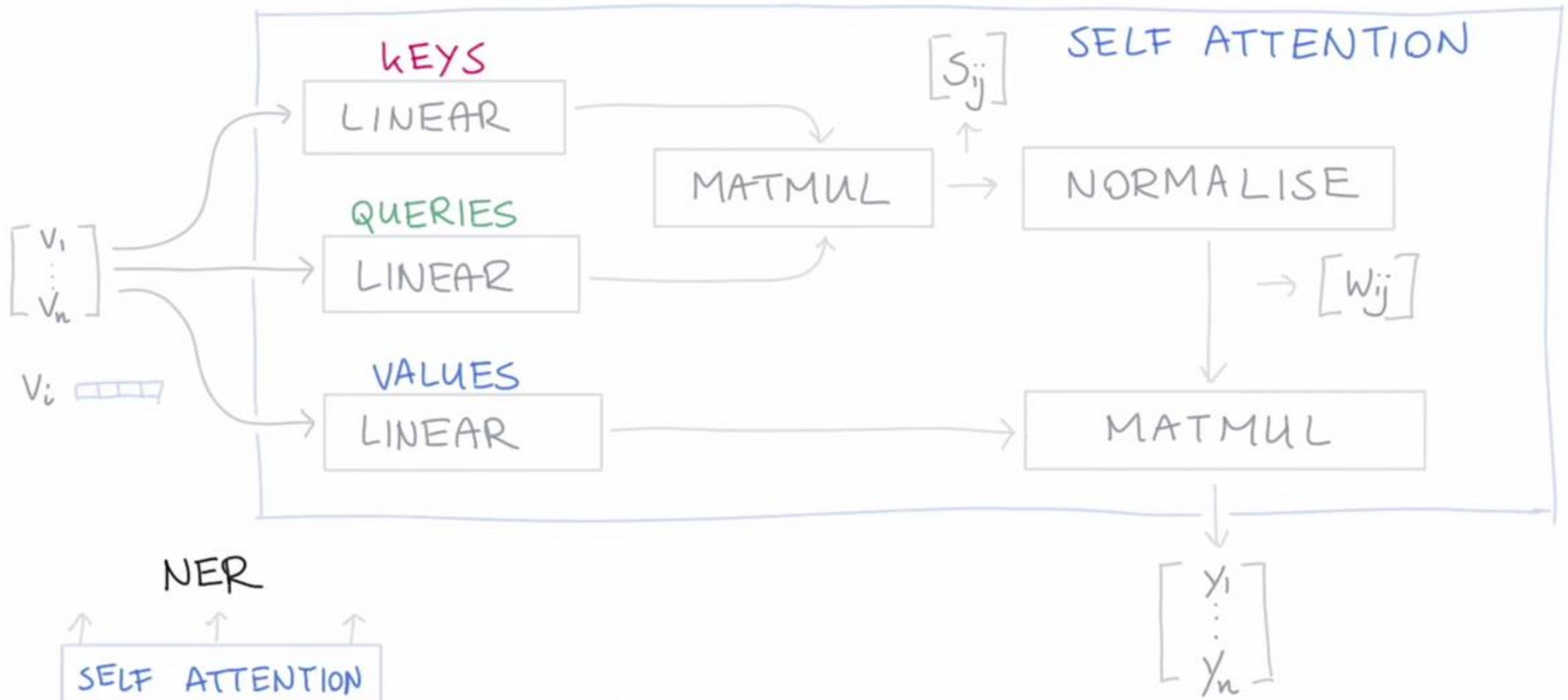
$1 \times \underline{k}$     $k \times k$     $1 \times k$

$v_1M_k$   $v_2M_k$   $v_3M_k$   $v_4M_k$  KEY

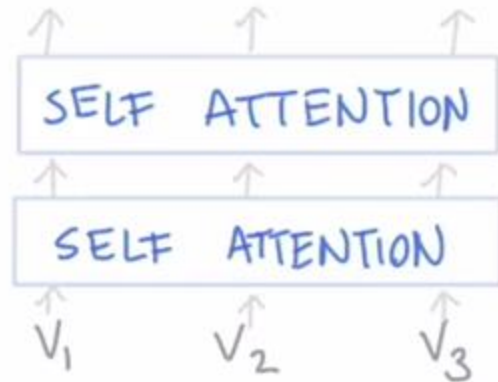
$\leftarrow v_3M_Q$

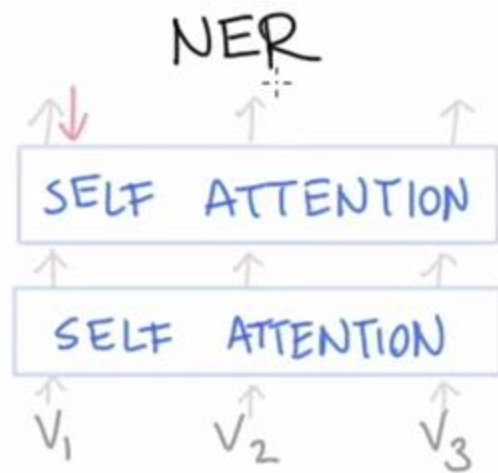
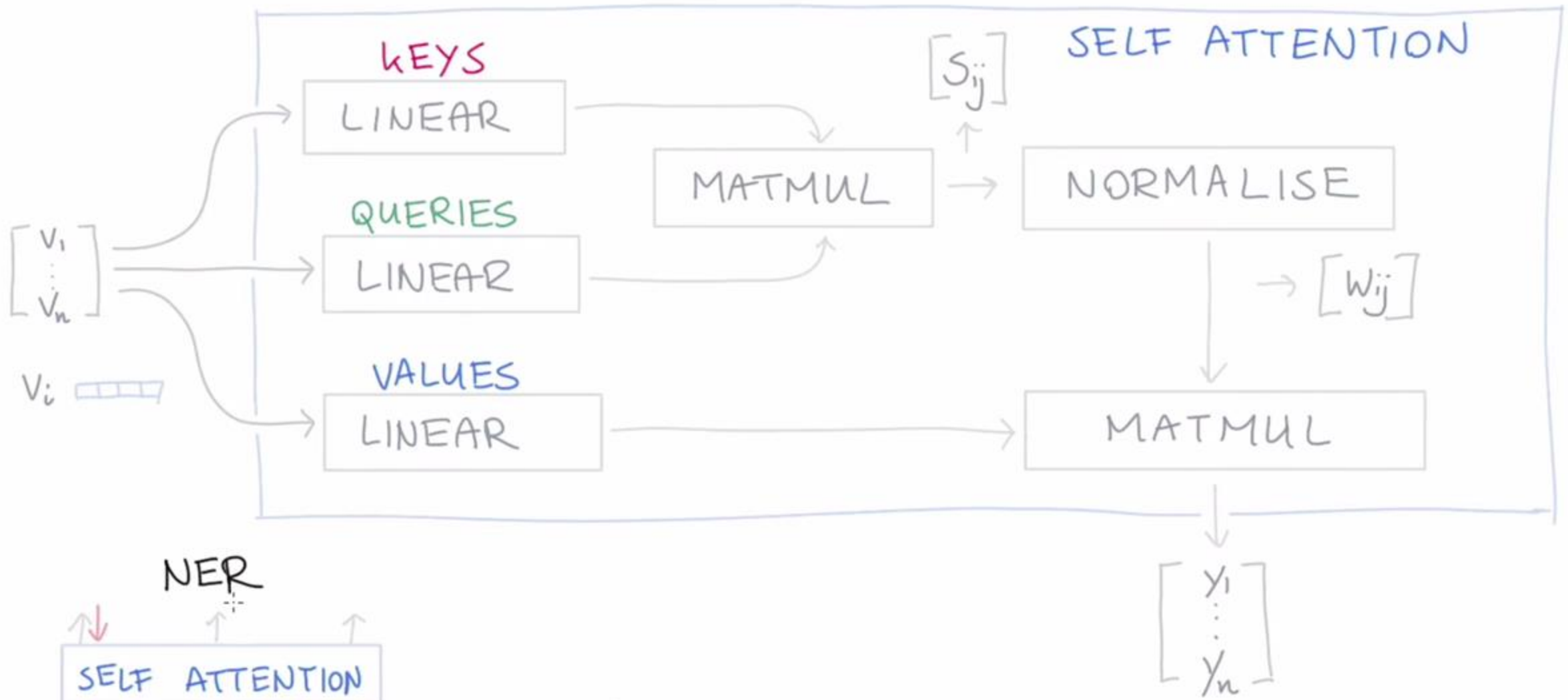


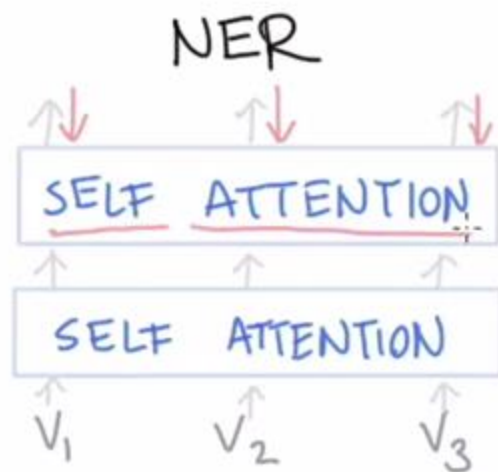
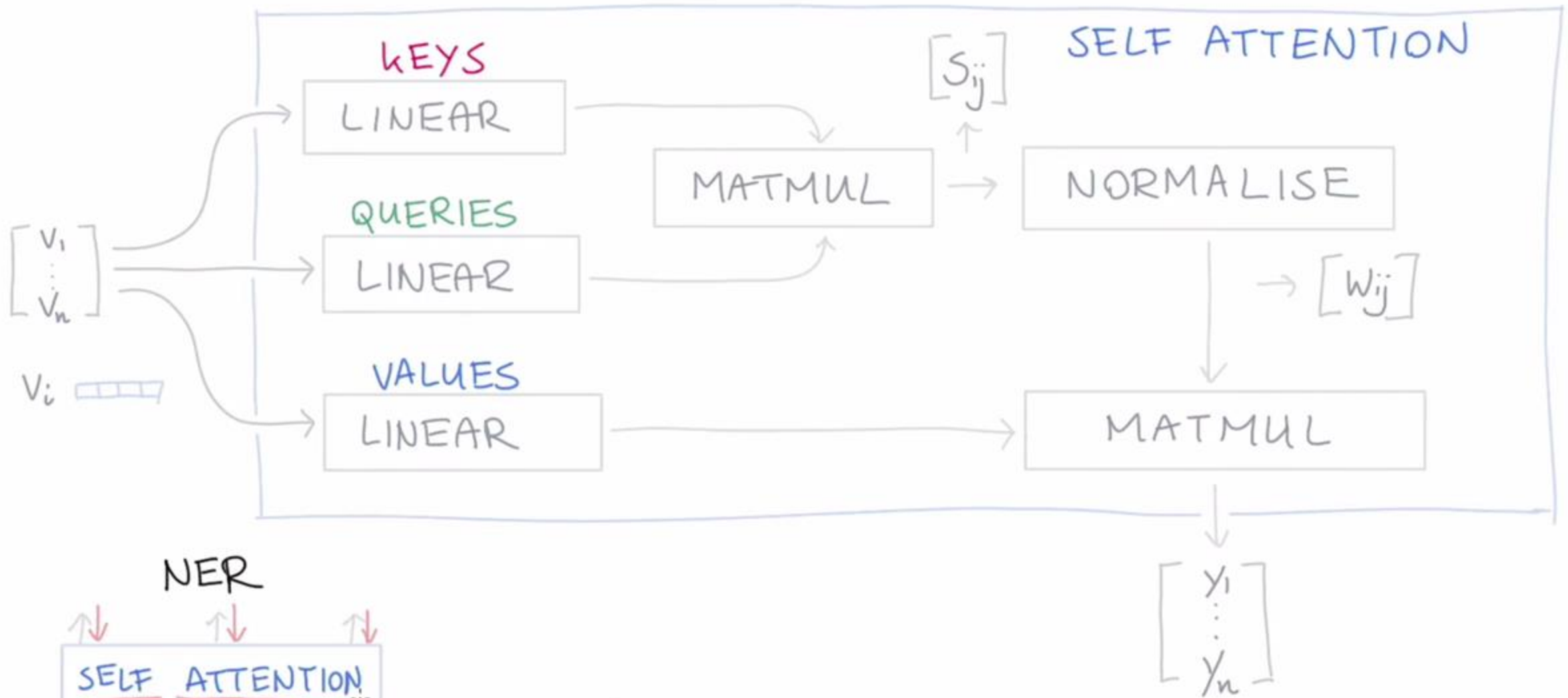


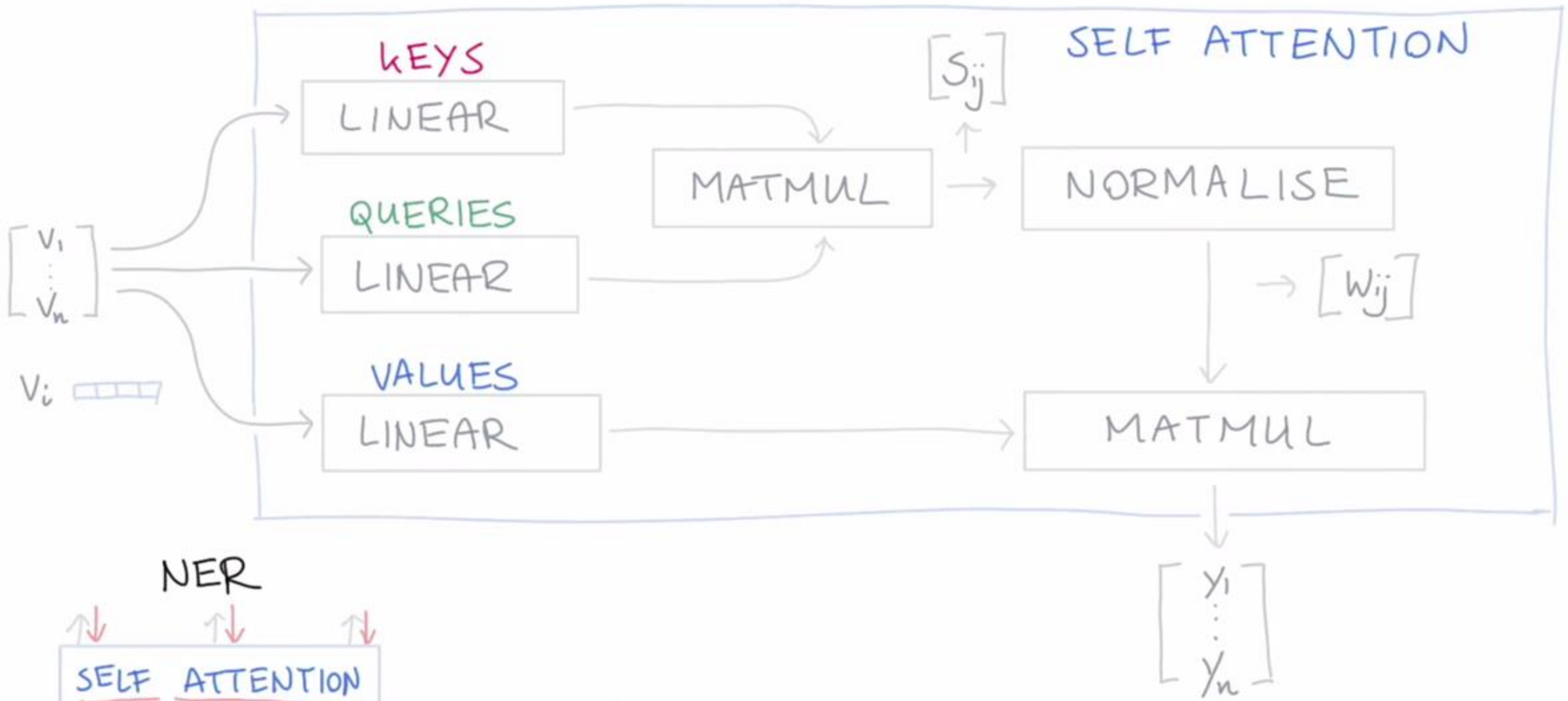


**NER**

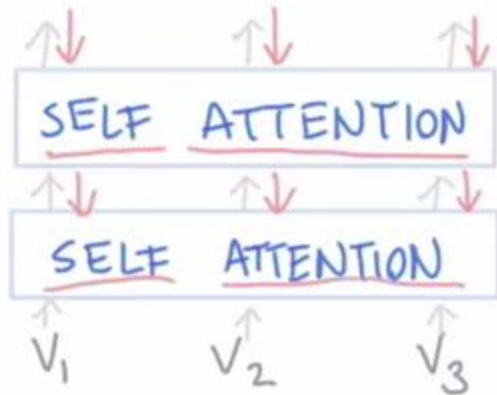


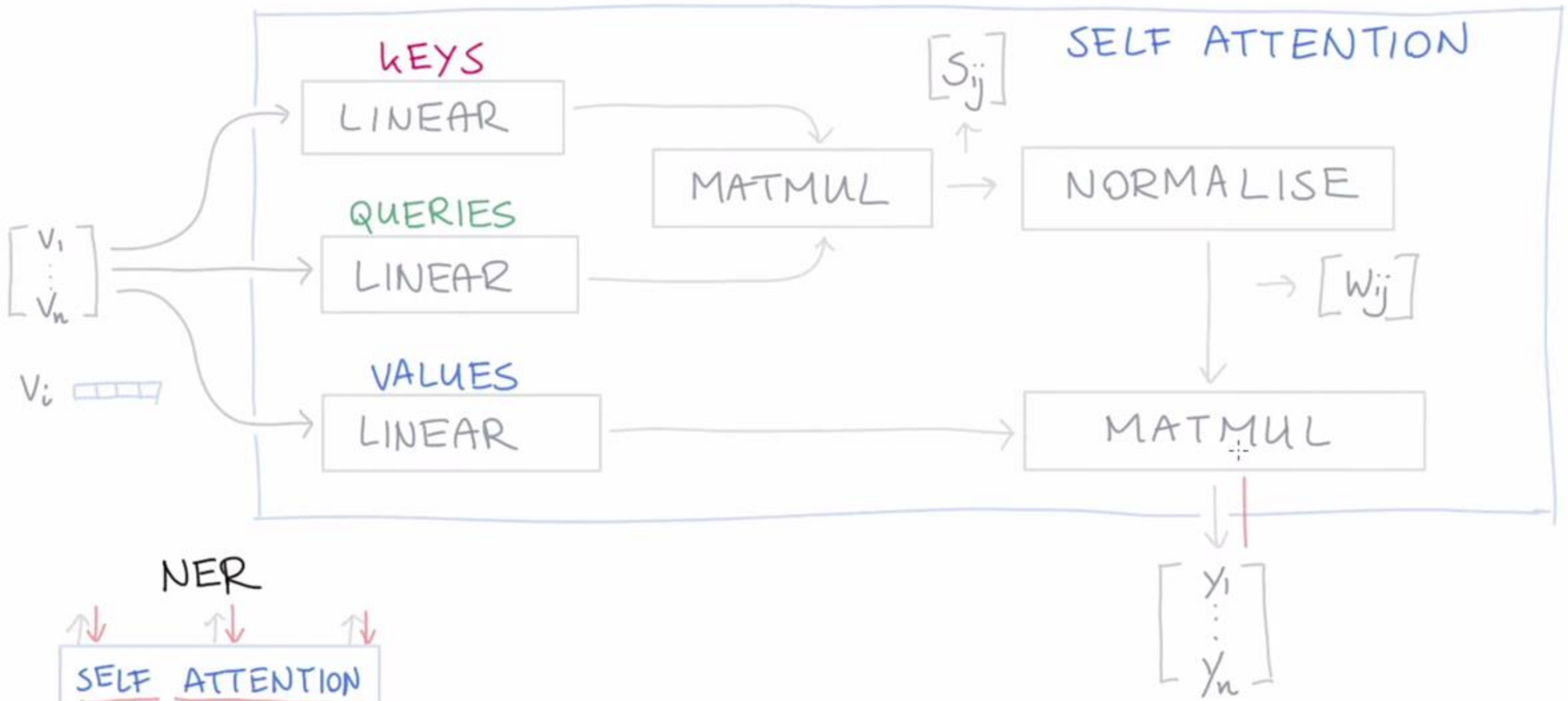




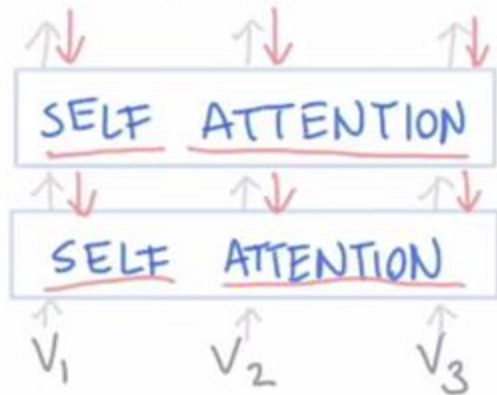


**NER**

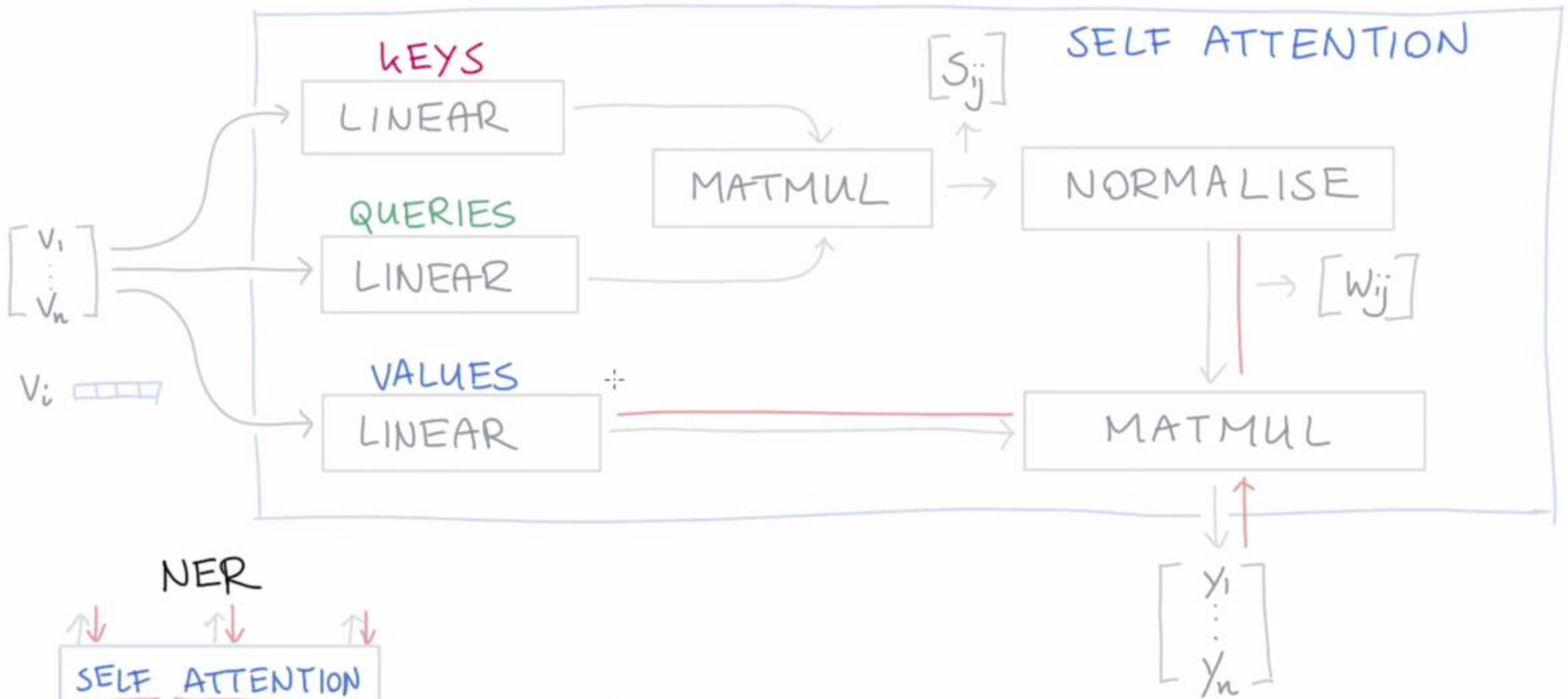




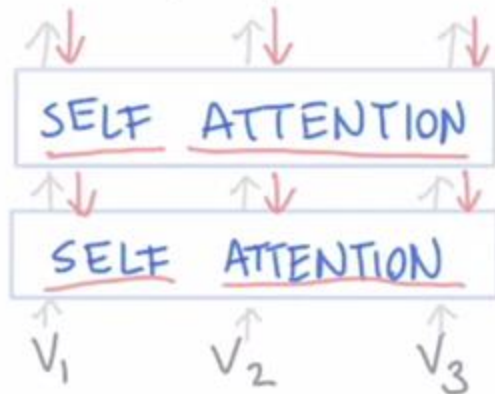
**NER**

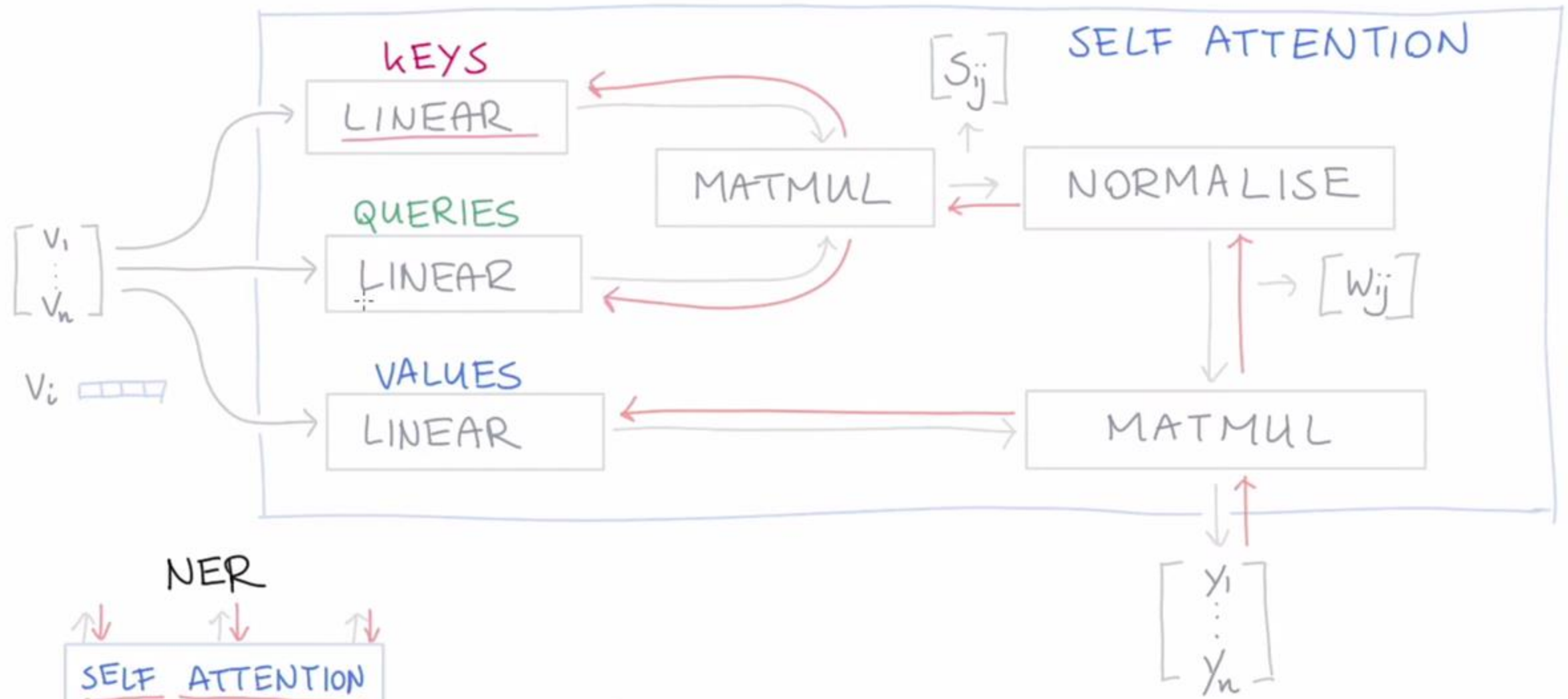




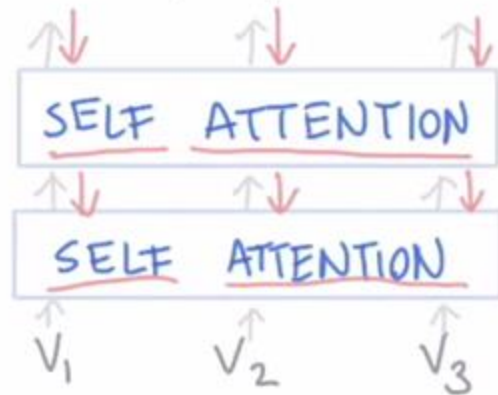


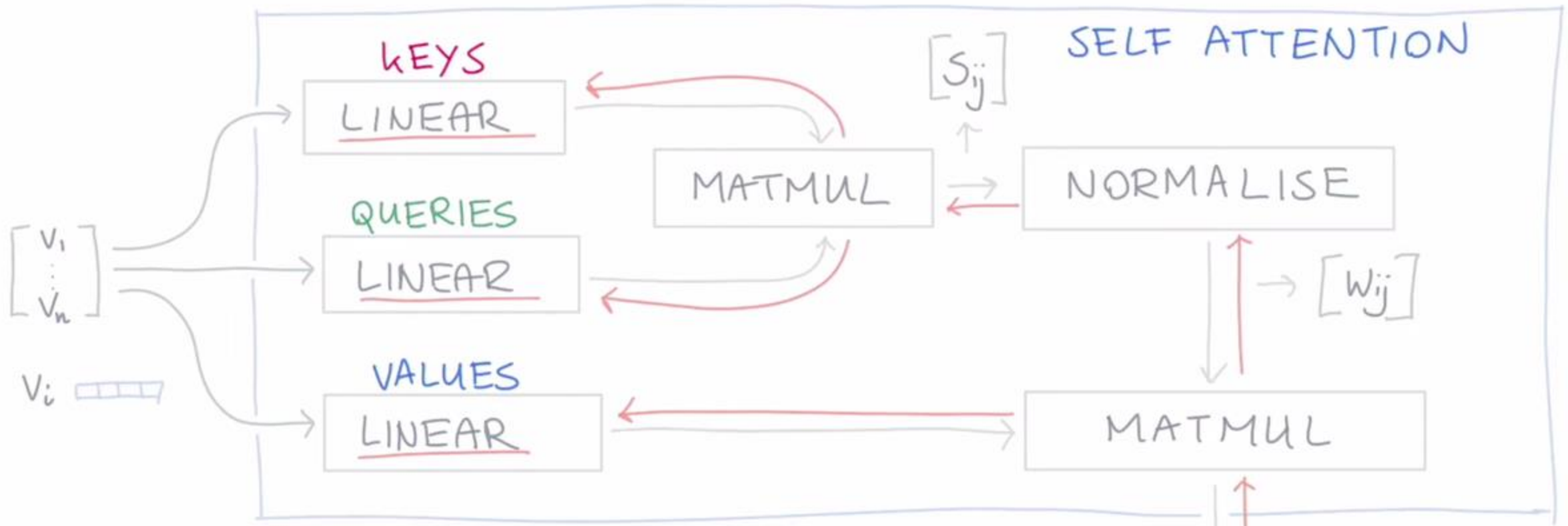
NER



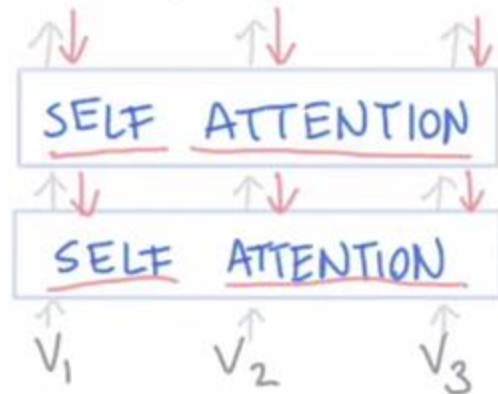


**NER**





NER



← 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\*<sup>†</sup>**  
University of Toronto  
aidan@cs.toronto.edu

**Lukasz Kaiser\***  
Google Brain  
lukaszkaiser@google.com

**Illia Polosukhin\*<sup>‡</sup>**  
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

# 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](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

