

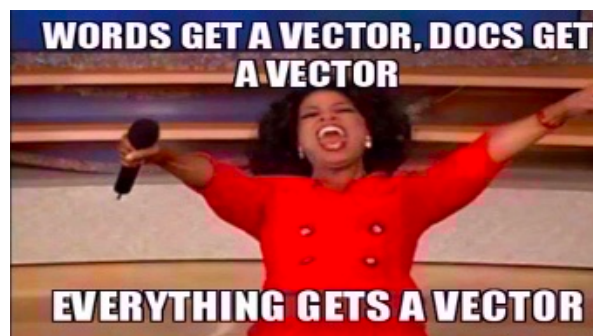
May 15, 2023.

DSML: NLP module.

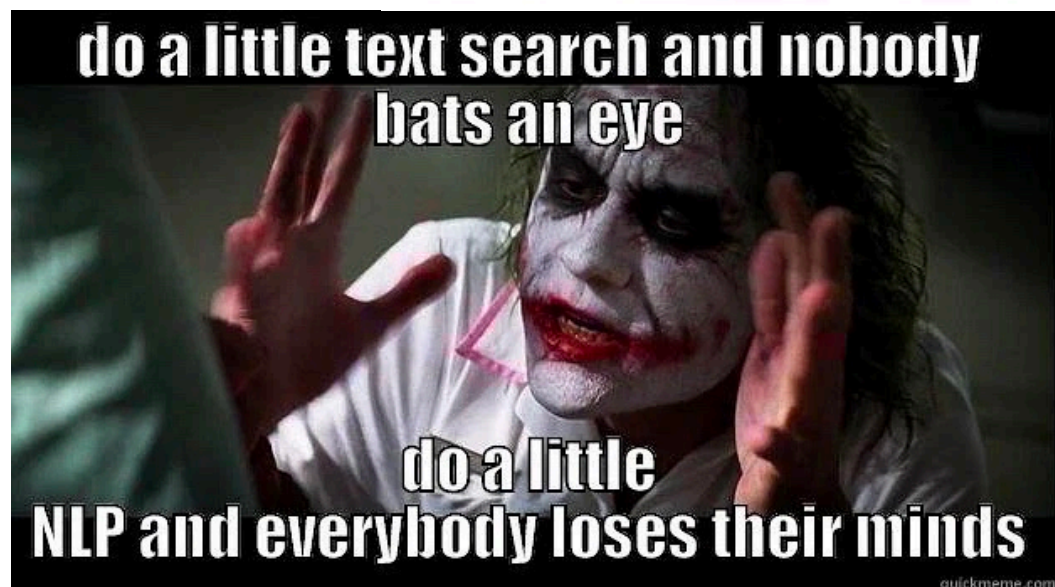
Word embeddings
in a nutshell

Transformers - 2.

Class starts
@ 9:05



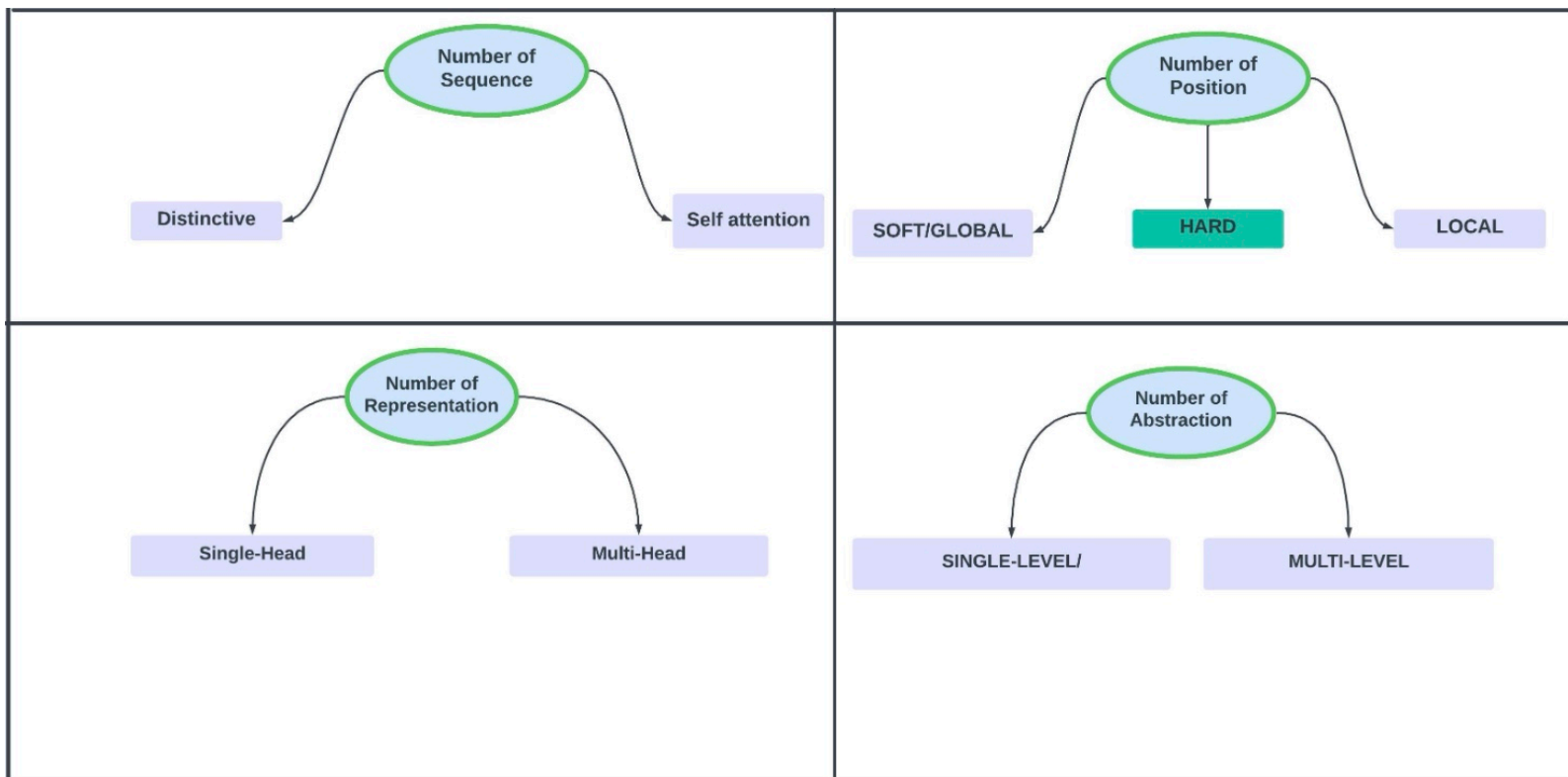
When you penalize your Natural
Language Generation model for
large sentence lengths



Recap:

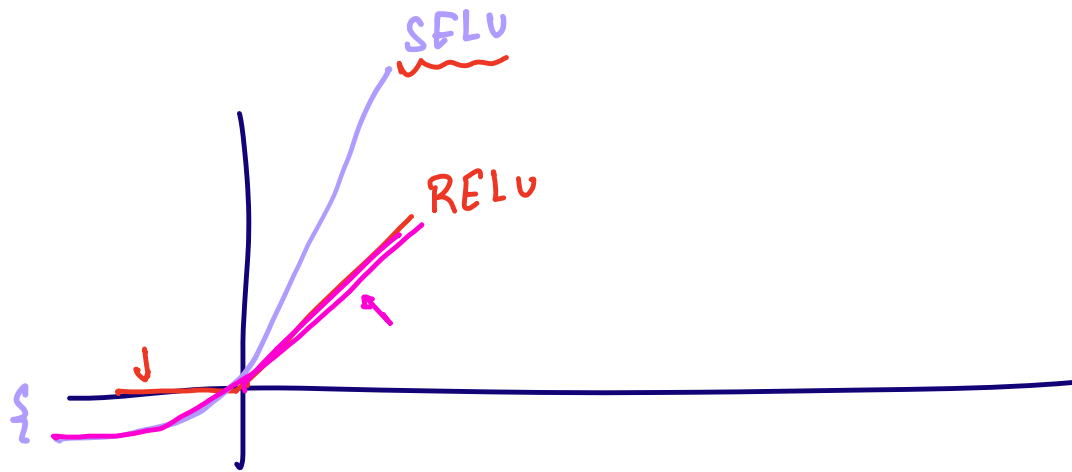
* RNN + Attention

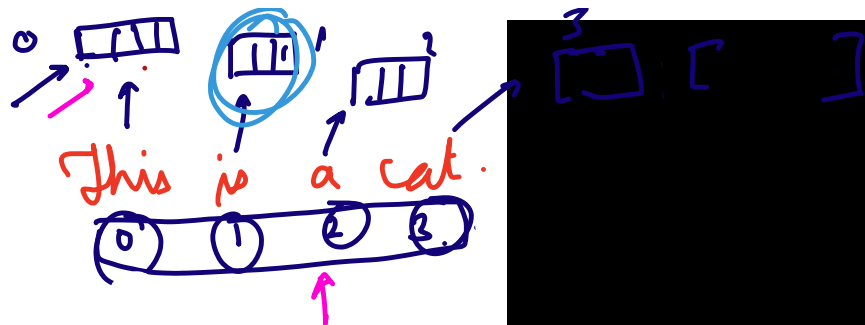
- Single head attention.
- Key: Encoder hidden states.
- Query: Decoder hidden states.
- Value: Encoder hidden states.



Agenda:

- 1] Code Implementation : RNN + Attention.
- 2] Transformer Architecture.
- 3] Transformer Implementation : Keras + TF.



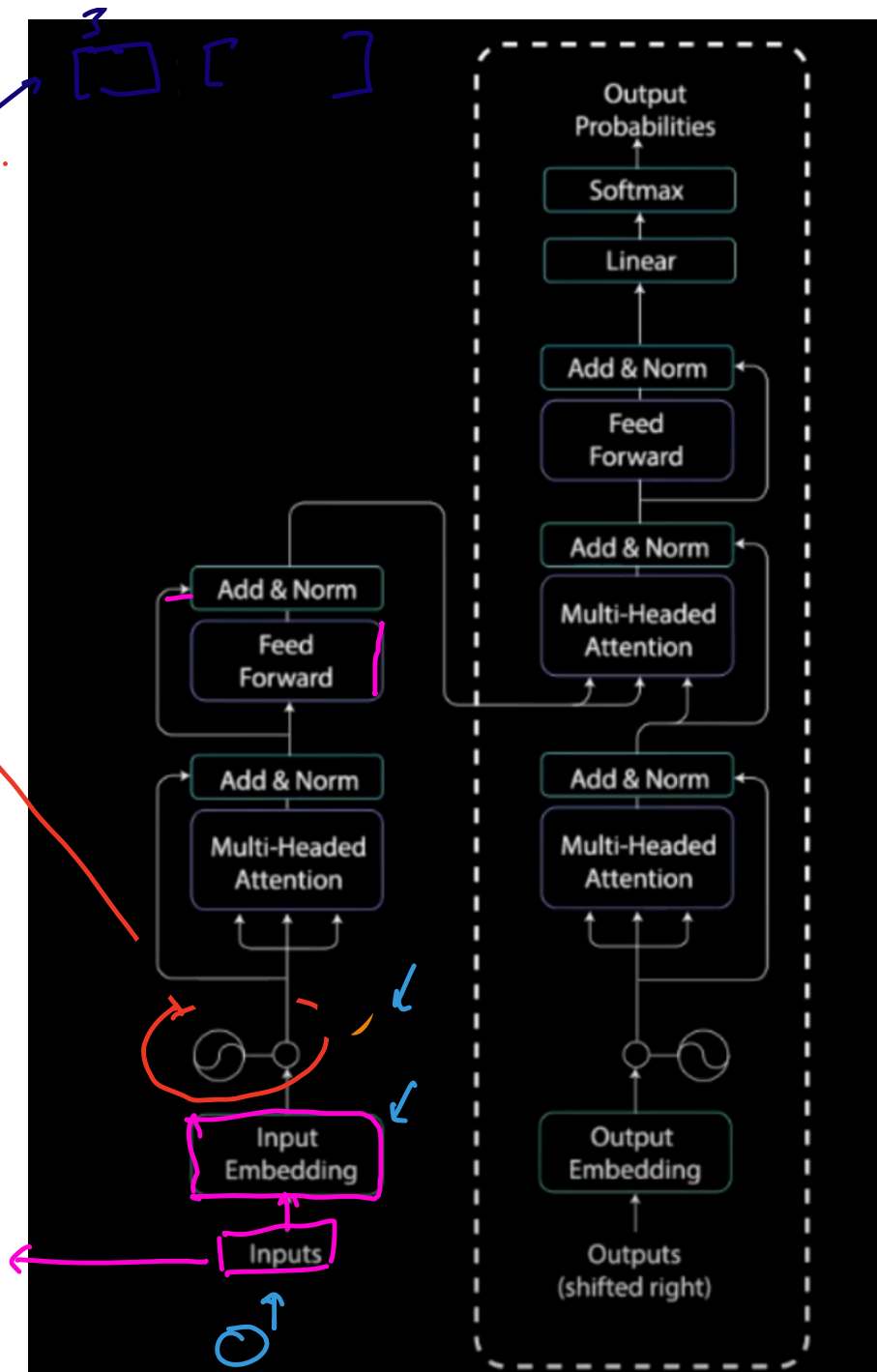


Positional
Encoding
block.

Sentence in
↓ Portuguese.

"

_____"



$[-1, 1]$

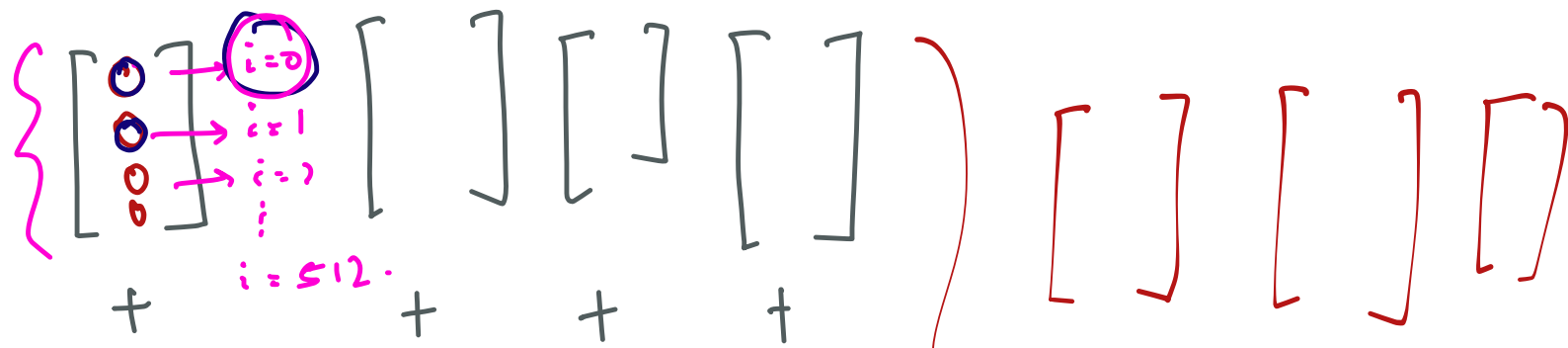
$\left[\begin{array}{cccccc} 0 & 1 & 2 & 3 & \dots & 19 \\ 20 & 20 & 20 & 20 & \dots & 20 \end{array} \right]$

5

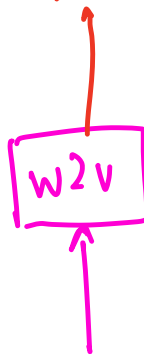
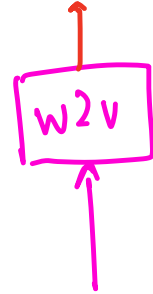
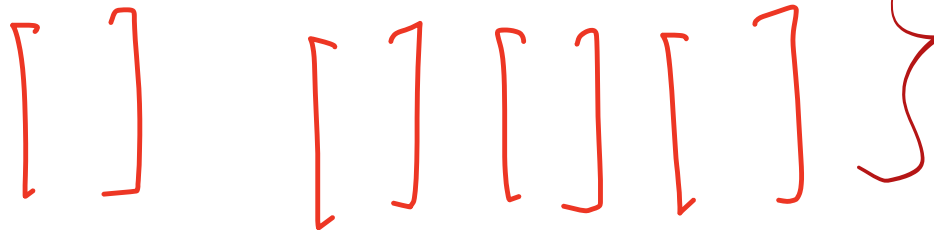
3

0.3

(d)
512
length
vectors.



512
length
vectors.



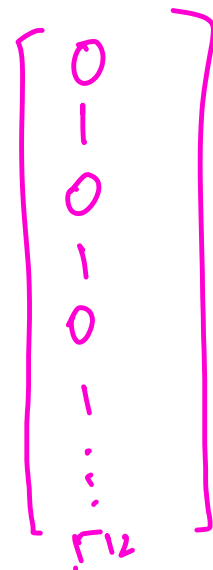
(pas)

→ J_{hi}
0

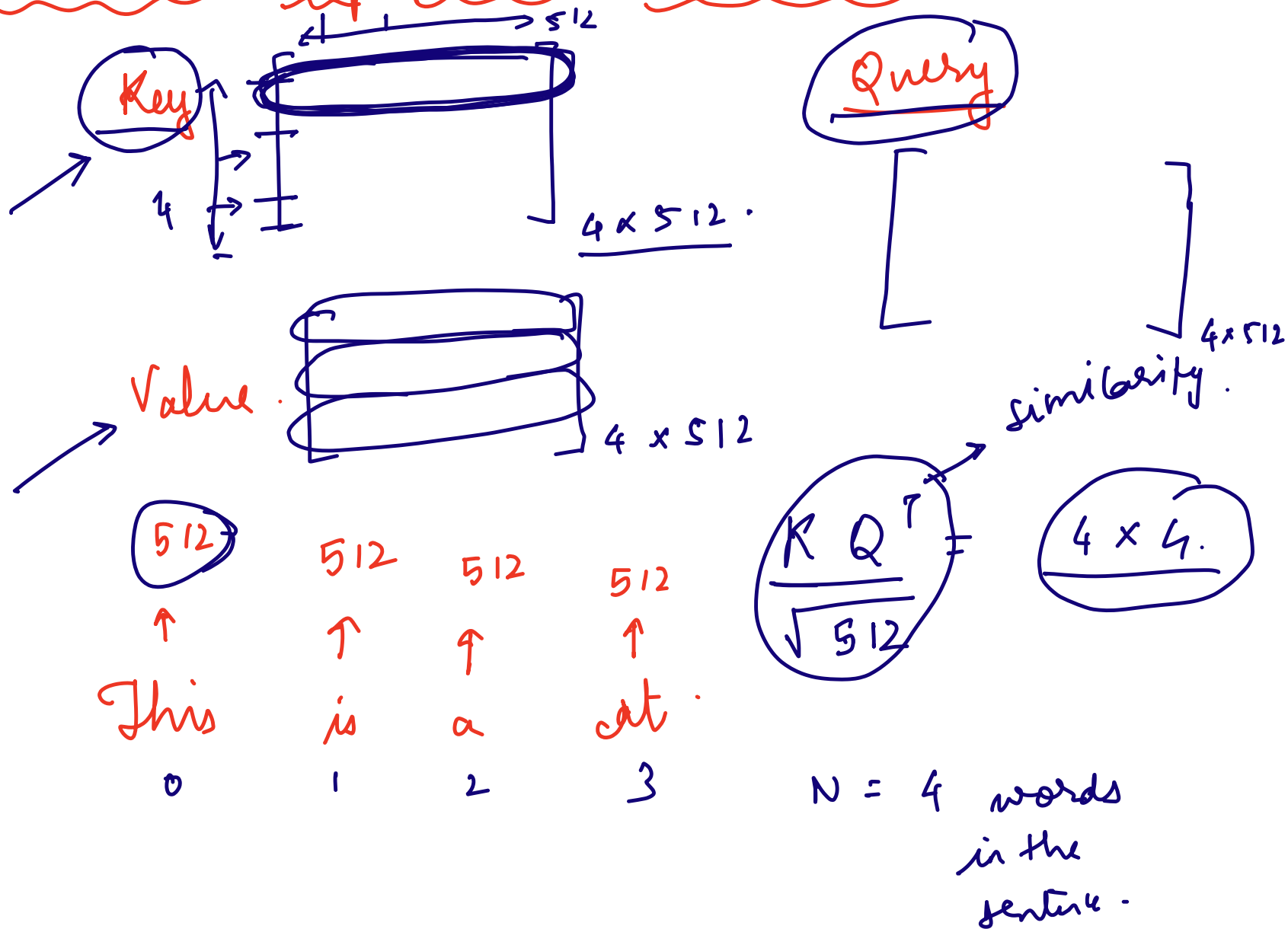
is
1

a
2

cat
3



Multi headed self-attention mechanism.



Difference between single-head self attention & multihead self attention.

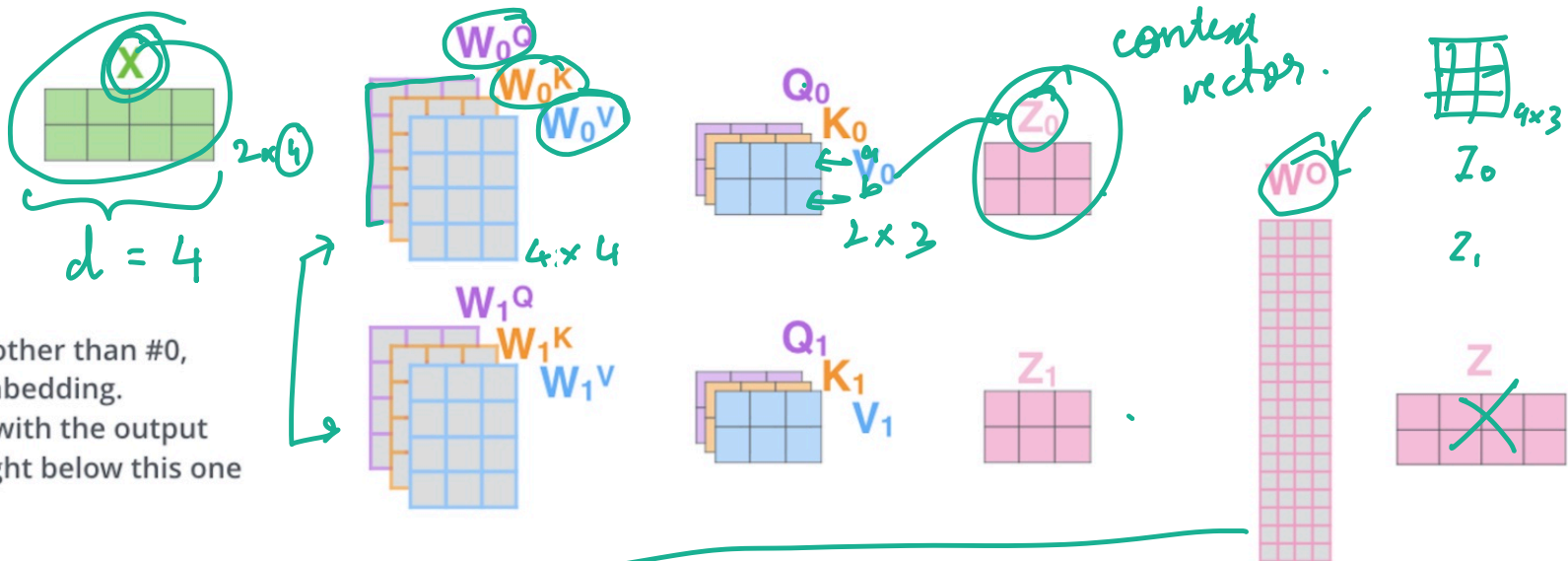
Single Key $\left[\begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array} \right]$ rows are word 2 vec vectors.

Multi

Key \rightarrow First pass word 2 vec outputs through a dense layer.

Then, the outputs of the dense layer is the key matrix.

Thinking
Machines
 $N=2$



* In all encoders other than #0,
we don't need embedding.
We start directly with the output
of the encoder right below this one

Attention weights = $\text{softmax} \left(\frac{Q_0 K_0^T}{\sqrt{4}} \right)$

2×1 $\left[a, b \right]$

2×2

$mHA \rightarrow Z$

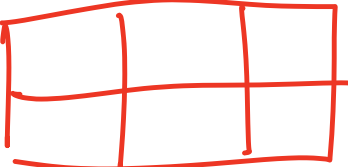


2×3

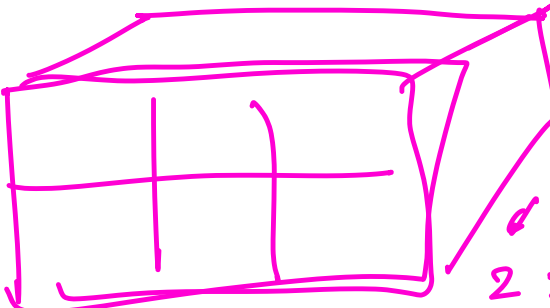
\oplus

\rightarrow

X



2×3



2×3

3×3

Current Batch Feedback.

- ✓ ① Batch homogeneity \rightarrow
- ✓ ② More classes for more important topics \rightarrow
- ✓ ③ Retention is an issue \rightarrow

