Transfer learning

pretrained models

+

use specific data for finetuning

not in Raw

Word Embeddings

love apple phones.

$$Apple = 0.1 \times E_{love} + 0.5 \times E_{Apple} + 0.7 \times E_{phones}.$$

Word Emb. space

Tech $\xleftrightarrow{x}$ $\xleftrightarrow{y}$

phone d · apple

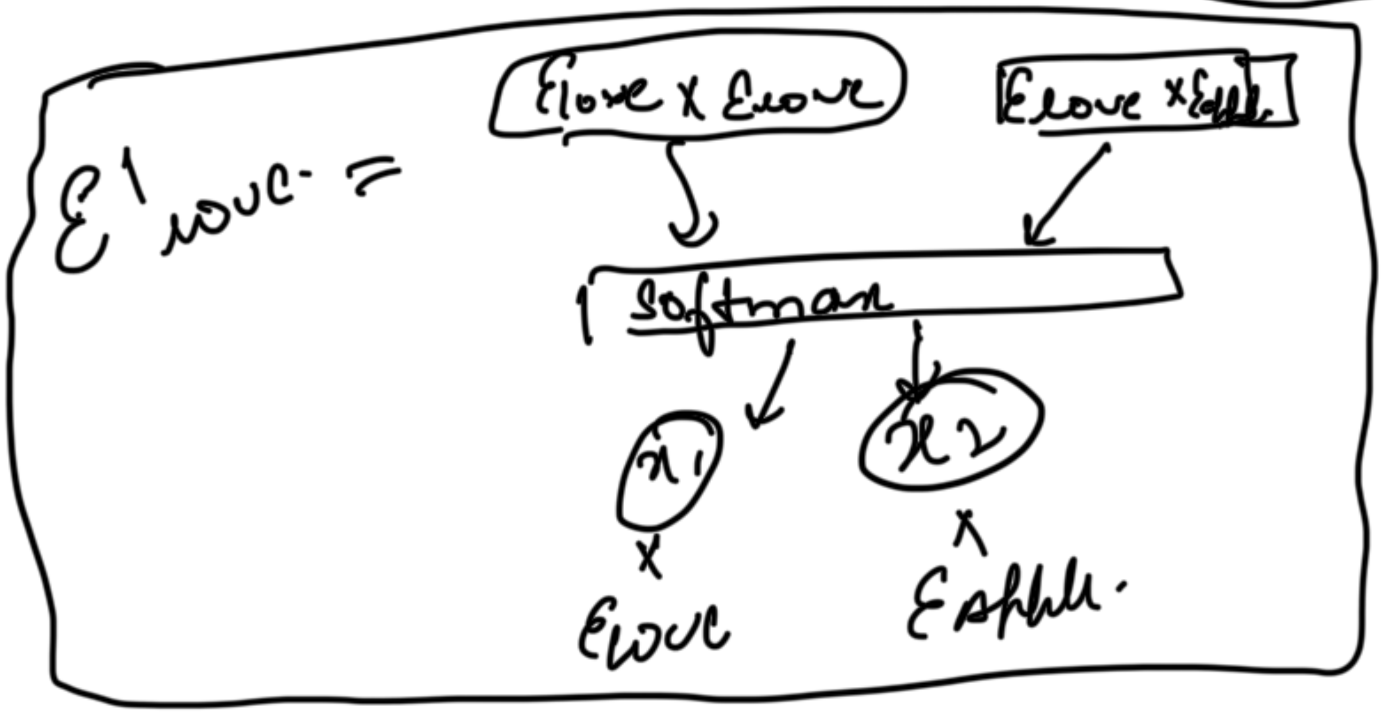
juice

$y > x \rightarrow$ more similar to tech.

0·1 $\rightarrow$ rel^n b/w apple & love

$\vec{E}_{apple} \cdot \vec{E}_{love}^T \rightarrow$ is passed to softmax f^n

to get the prob.



$$\text{Similarity} \propto \frac{1}{d}$$

Problem

- No learnable Params
- Use of same vector

To solve

$E_{apple} \cdot W_q$

$E_{love} \cdot W_K^T$

E_{apple}

W_K W_Q W_V

K_{apple} Q_{apple} V_{apple}

softmax

x_1

$\times$

$E_{love} \cdot W_V$

Eg →

Q_{Apple} K_{love}

Q_{Apple} K_{apple}

Q_{Apple} K_{phone}

Singular O/P.

softmax.

x_1 x_2 x_2

$E_{apple}' = x_1 * V_{apple} \quad x_2 * V_{love} \quad x_2 * V_{phone}$.

$$\boxed{E_{Apple}' = x_1 \cdot v_{apple} + x_2 \cdot v_{love} + x_3 \cdot v_{phone}}$$

↓

no dependence of E of love & phones.

So, they eliminate sequential processing & allows parallelization.

↓

Long range dependencies

$\text{Softmax}(Q \cdot K^T) \cdot V$ → done to add learnable params in the matrices

(*) <u>Linear Transformation</u> → changes dimension
→ extracting useful features

sce • Obser

• Juice

• Watch

• Apple ⟷ • Watch

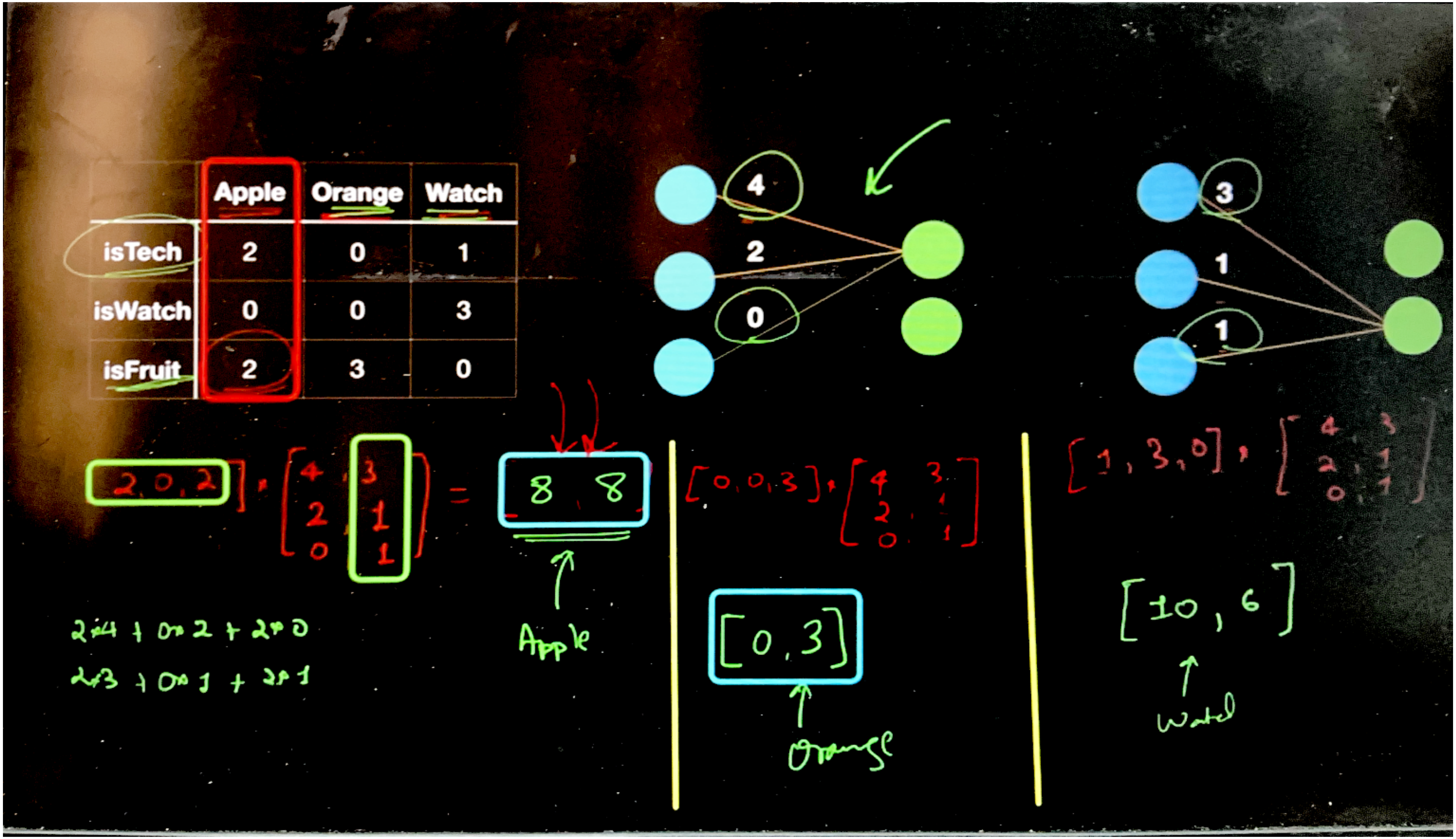
very apart

time

in space so
• Tech
not much relation can be captured.

Change dimension

$y = w \cdot x$ → just like ANN.



Apple
Orange
Watch
isTech
2
0
1
isWatch
0
0
3
isFruit
2
3
0
4
2
0
3
1
1
[2, 0, 2] · [4 3, 2 1, 0 1] = [8, 8]
Apple
2*4 + 0*2 + 2*0
2*3 + 0*1 + 2*1
[0, 0, 3] · [4 3, 2 1, 0 1]
[0, 3]
Orange
[1, 3, 0] · [4 3, 2 1, 0 1]
[10, 6]
Watch

(*) Query, Key, value

Q, K, V.

sound of [Dog] = Bark

query (sound), key (Dog), value (Bark)

Q Apple → Query.

K love → Key

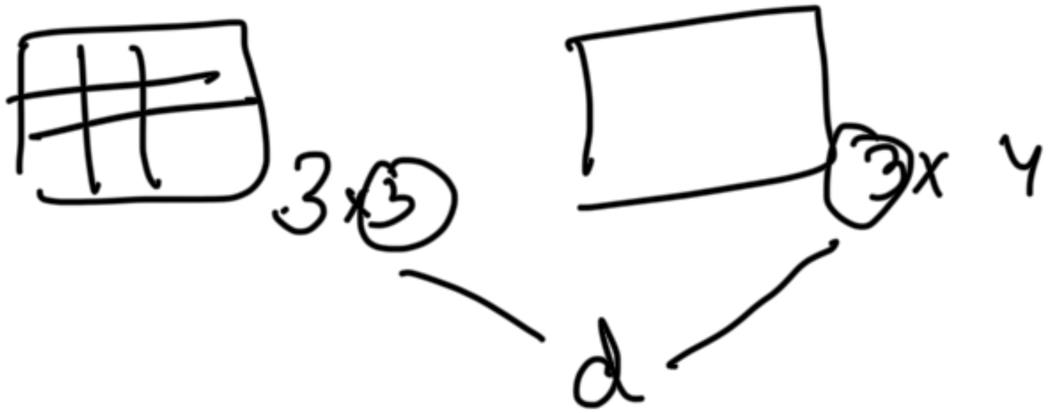
V → value.

$$\boxed{\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V}$$

⊛ why $\sqrt{d_k}$ → dim. of Q, K, V.

↓ hyperparameter.

Variance $(A \cdot B) \propto n$, n = common dim. of A & B.



High var → Vanishing grad prob

↓

Hinders training

↓

so we use $\sqrt{d_k}$

$C = A \cdot B$

$$\boxed{Var(C) = n \cdot Var(A) \cdot Var(B)}$$

↓

→ zero mean

→ Indep. random var.

→ Entries of A & B have identical values.

$X = Q \cdot K^T$ } To reduce variance

$var(X) \propto d_k$ } we use $\sqrt{d_k}$ to keep $var(X)$ same.

$$X \Rightarrow var(X)$$

$$a \cdot X \rightarrow a^2\, var(X)$$

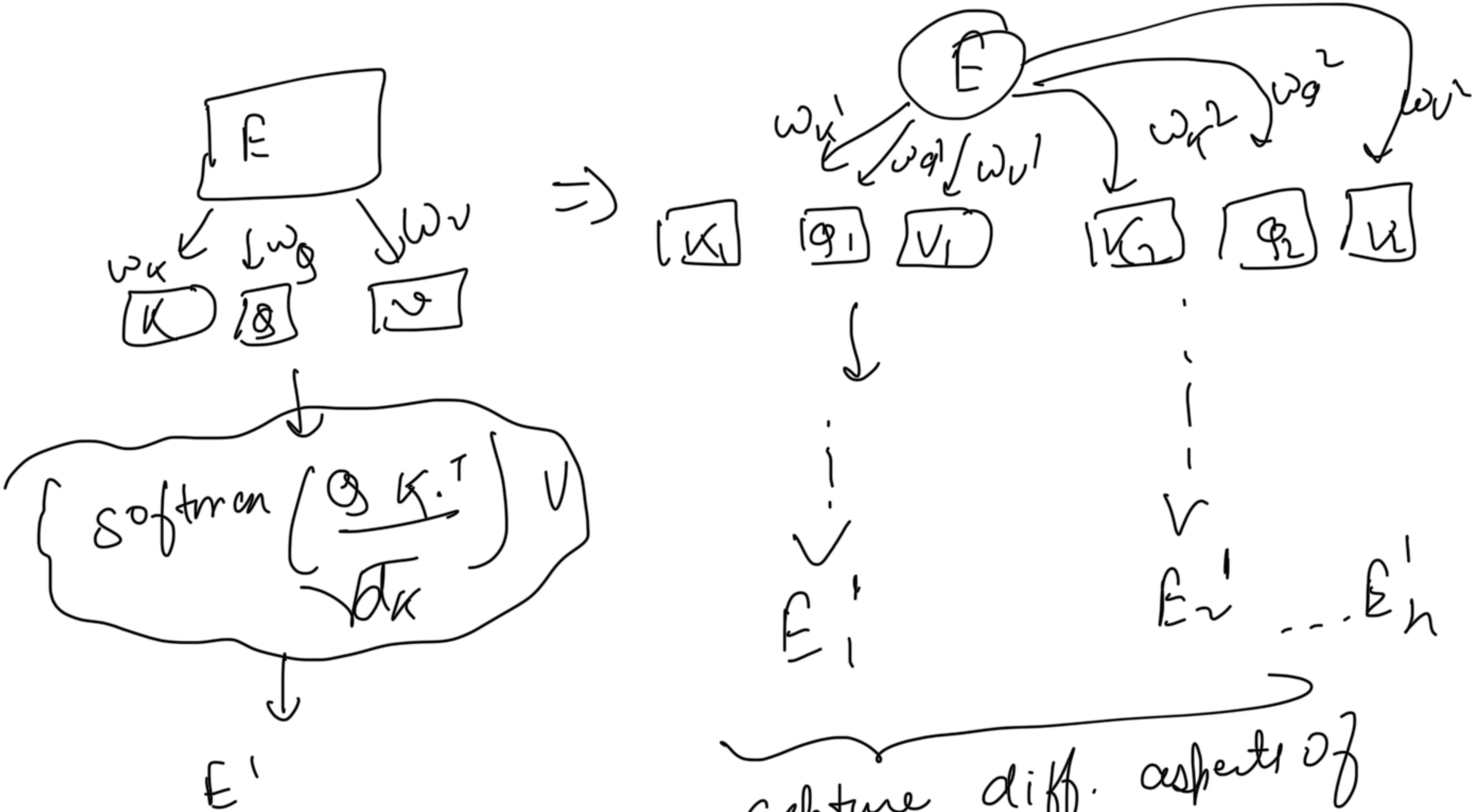
Multi_Head_Attention

As complexity of model ↑ It is tough to capture the meaning of different words & other things

Spatial Rel^n — Sub-verb obj — Time →

self-att. 1 — self-att 2 — SA 3

to provide diverse content
just like CNN



capture diff. aspects of
word in a sentence.

In GPT 2 → 96 Attention Heads

→ 96 diff perspective

$|E_1'|E_2'| \ldots | E_n'|$

$\downarrow * W \rightarrow$ learn parameter

$\downarrow$

Linear transformation

Keeps only the relevant features

Positional Encoding

↳ due to lack of sequential understanding.

If Individually sin / cos is used then → due to periodicity 2 words may be identical

↓

use $\sin(x)\ \cos(x)$ together

In Og transformer (

512 dim → 264 pair sin cosin

$\boxed{\sin(x)\ \Big|\ \cos(x)\ \Big|\ \sin(\tfrac{x}{2})}\ \boxed{\cos(\tfrac{x}{2})}$ - - - -

reduced freq.

$$\boxed{\begin{aligned} PE_{(pos,2i)} &= \sin\left(pos / 10000^{\frac{2i}{d_{model}}}\right) \\ PE_{pos(2i+1)} &= \cos\left(pos / 10000^{\frac{2i}{d_{model}}}\right) \end{aligned}}$$

If we know

$PE_{pos} \rightarrow PE_{pos+(k)} \rightarrow$ offset

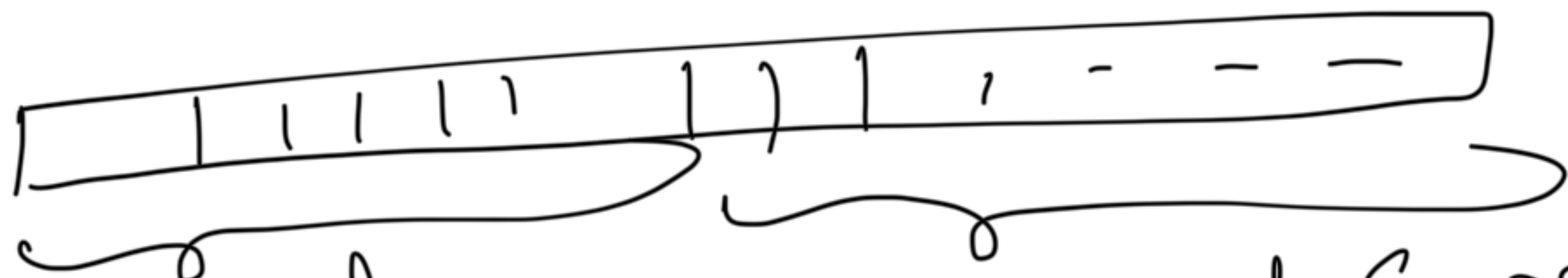
can be calculated

↓

as there is a Transformation

matrix T_k.

$$\boxed{PE_{pos+k} = T_k \times PE_{pos}}$$



Word Embedding (E)

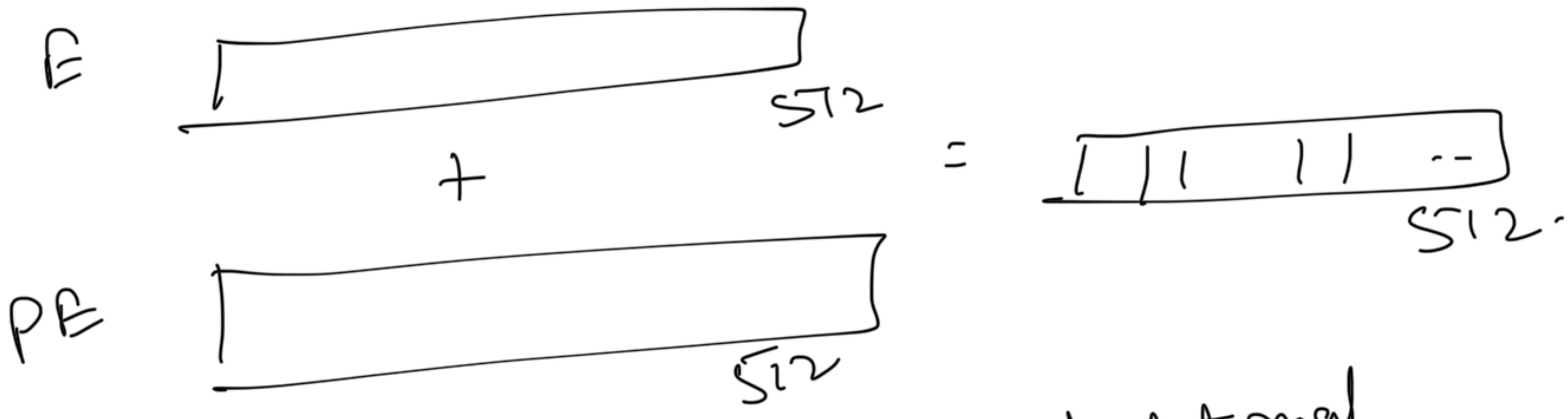
Positional Encoding (PE)

but $\rightarrow (E + PE) * W$ $\quad W_{k,v,q}$

so, dim of W increases

(+) → additional

Hence we use (+) addition operation



(+) → reduces computational overhead