V.V. Imp

Distributed Representation language Model

Text → vector

State of the art

↳ word Embeddings

Google → word2vec, Glove (Stanford), Fast Text (Facebook)

Internal

- → Document Embedding (DOC 2 vec)
- → Semantic in nature
- → Low dim vector [50 - 300]
- → Dense vector
- → Matrix Factorization
- → Black Box (Deep Learning)

Happy - Dense

[. - - - -]

$v_1 \uparrow w_1$ $v_2 \uparrow w_2$ $v_2 \uparrow w_3$

$v_1 \sim v_2$

$w_1 \sim w_2$

$w_1 \not\approx w_3$

$v_1 \not\approx v_3$

tasty delicious football

word2vec — max - 300

Token

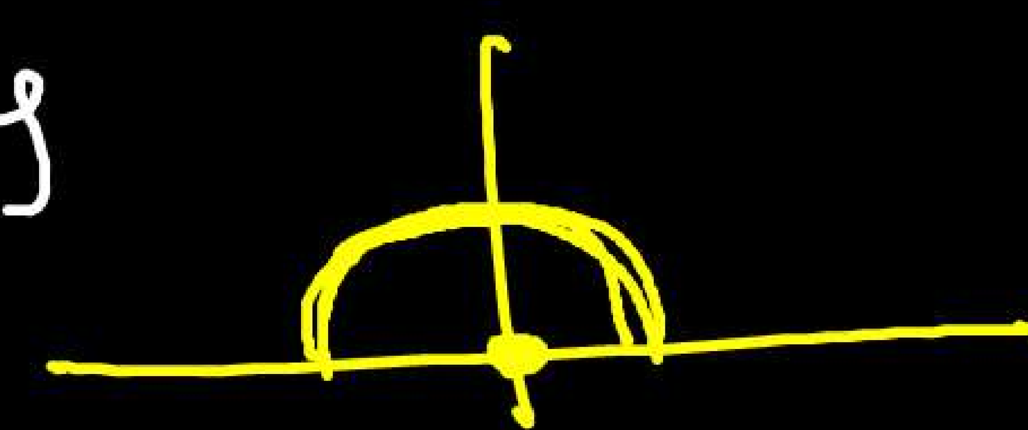
$w_1 \rightarrow v_1 = [- - - - - - - 300]$

$w_2 \rightarrow v_2 = [- - - 300]$

$w_3 \rightarrow v_3 = [- - 300]$

→ Semantic meaning

Cosine Similarity



(2) relationship

$(V_1 \sim V_2) \,||\, (V_x \sim V_y)$

$(V_{man} - V_{woman}) \,||\, (V_{king} - V_{queen})$

(India - N. Delhi) || (USA - W.DC)

(1) w_1 & w_2 are semantically similar then v_1 & v_2 are closer.

Vocab → / Features ↓	King	Queen	Man	Woman	Monkey
Gender	1	0	1	0	1
wealth	1	1	0.4	0.4	0
Power	1	0.7	0.5	0.3	0
Weight	0.8	0.5	0.6	0.4	0.2
Speak	1	1	1	1	0

King = [1 1 1 0.8 1]
Q = [0 1 0.7 0.5 1]
M = [1 0.4 0.3 0.6 1]
W = [0 0.4 0.3 0.4 1]
monkey = [1 0 0 0.2 0]

King − Man + Woman = Queen

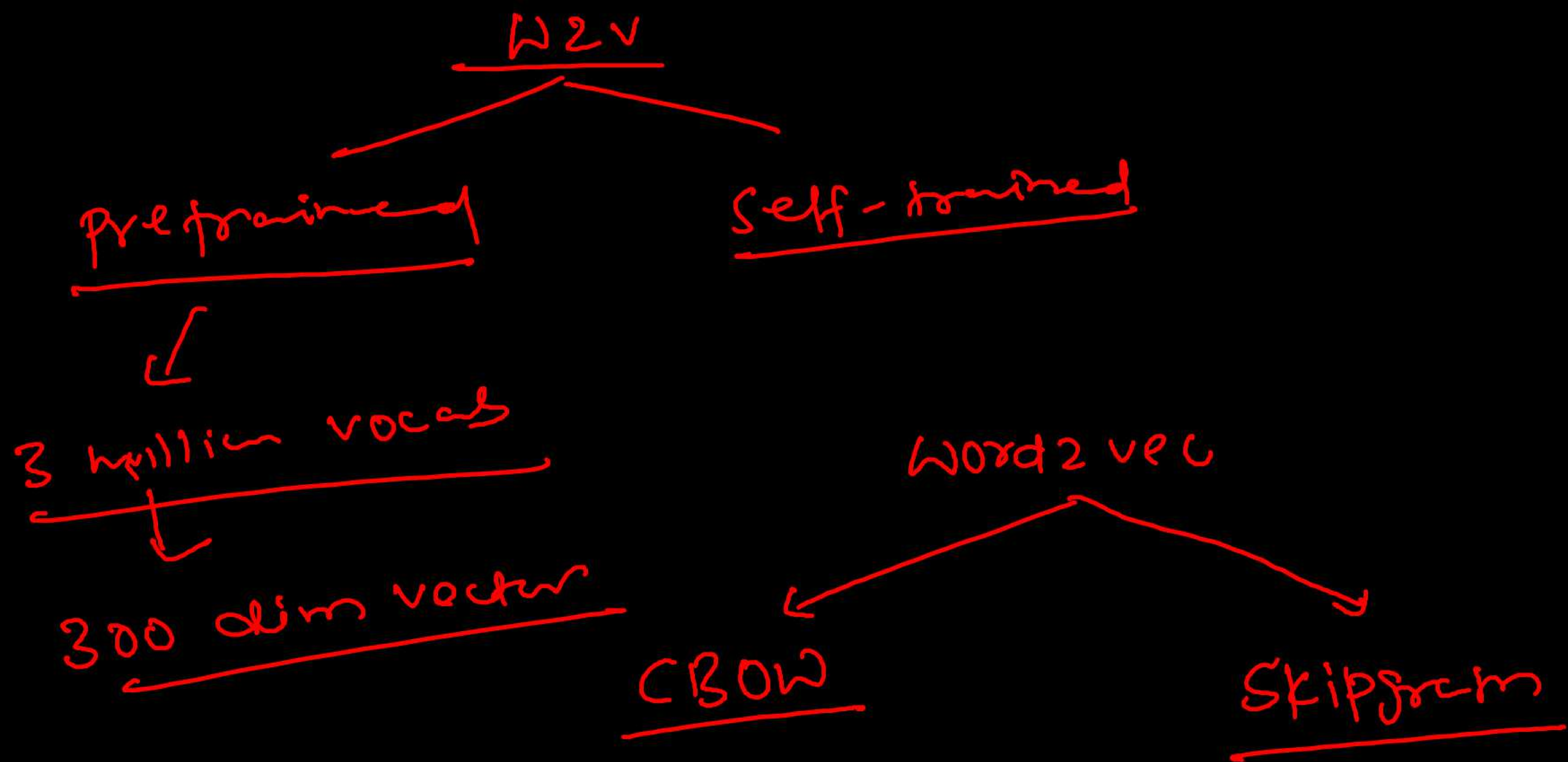
G: 1 − 1 + 0 = 0
W: 1 − 0.4 + 0.4 = 1
P: 1 − 0.5 + 0.3 = 0.8
W: 0.8 − 0.6 + 0.4 = 0.6
S: 1 − 1 + 1 = 1

[0 1 0.8 0.6 1] = o/p
[0 1 0.7 0.5 1] = Q

Audio Video Participants 23 Chat Share Pause Annotate Show webinar More

You are screen sharing EN Stop share

Talking:



Webinar Chat

through-word-embeddings-690df7e7b67d#:~:text=Understanding%20Word2Vec&text=By%20analyzing%20large%20corpora%20of,and%20even%20complex%20semantic%20relationships.

sure

Me to Everyone

LP https://jalammar.github.io/illustrated-word2vec/

Santoshkumar Pa... to Hosts and panelists

SP yes sir

tilak to Hosts and panelists

T same like as transfer learning?

Who can see your messages? Recording on

To: Everyone

Type message here...

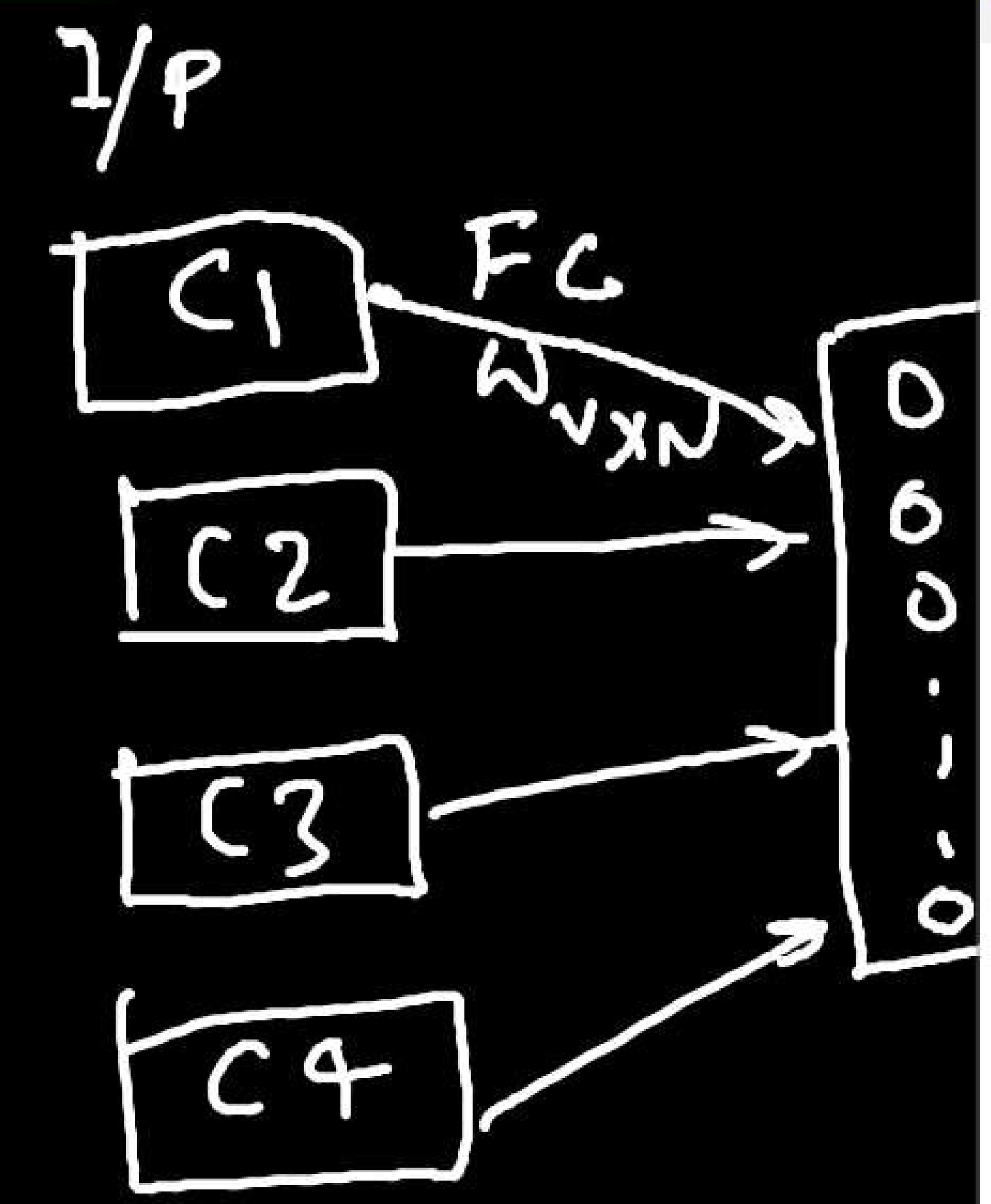
Word 2 vec

CBOW | Skipgram

(continuous Bag of words)

↓

core idea :- You have given context word, can be predict the focus word.



eg: C1 The Cat C2 On C3 the C3 Wall

Focus

The __ on the wall - {_ _ _ _}

window = 3

Audio Video Participants 23 Chat Share Pause Annotate Show webinar More

You are screen sharing EN Stop share

Talking: Learnvista Private Lim...

The ___ on {cat the wall}

The cat on
The the on
The wall on

CBOW

→ Not used in Real time
→ Bad Representation for rare words
→ overfits on frequent word.
→ D-dim classification

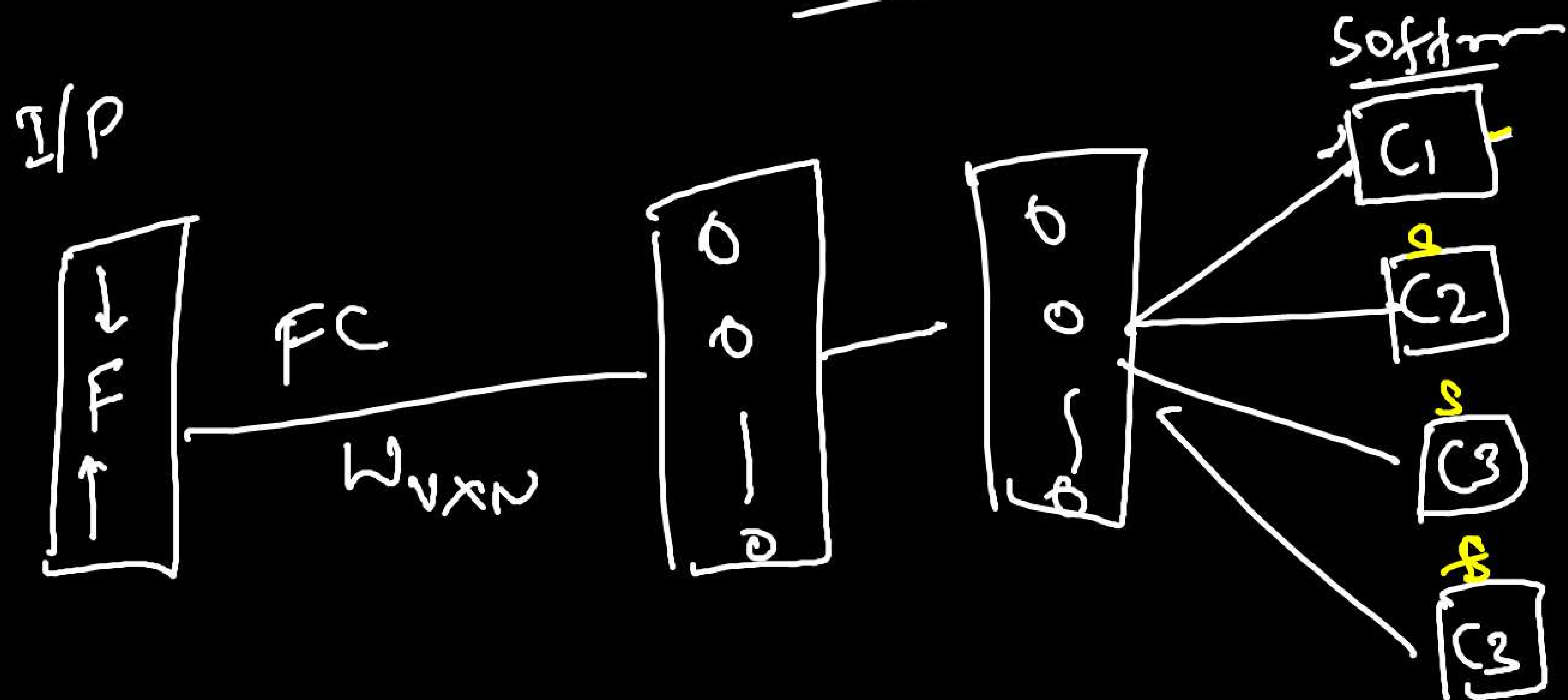
Audio Video 23 Participants Chat Share Pause Annotate Show webinar More

You are screen sharing EN Stop share

Talking: Learnvista Private Lim...

Skip gram

Idea :- predict context word given focus word.



Skip gr

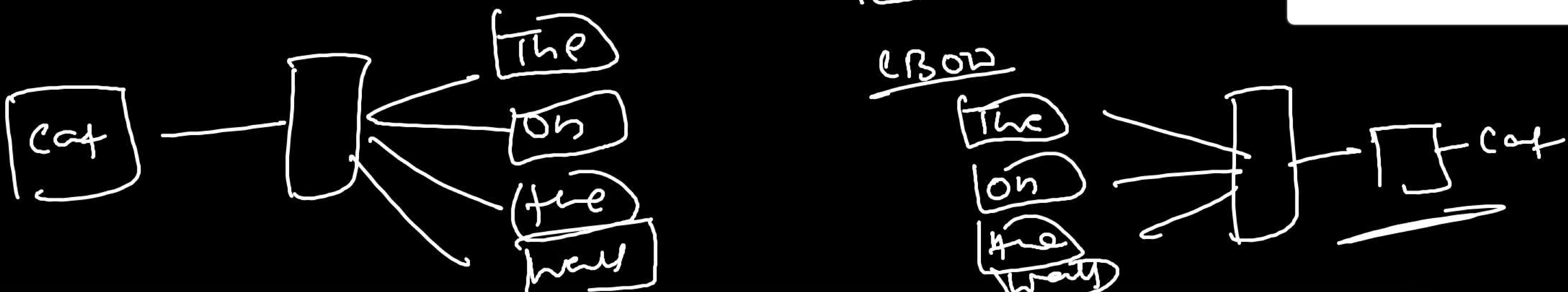
even w

than c

with

now w

the W



Webinar Chat

Usha Kumari to Hosts and panelists

but focus word is more difficult to predict

skipgram

Santosh to Hosts and panelists

cbow

Rashmi to Hosts and panelists

cbow

Mohamed Younus to Hosts and panelists

CBOW

chetan anand to Hosts and panelists

CBOW

Mohamed Younus to Hosts and panelists

n

Who can see your messages? Recording on

To: Everyone

Type message here...

Audio Video Participants 23 Chat Share Pause Annotate Show webinar More

You are screen sharing EN Stop share

Talking: Learnvista Private Lim...

Skipgram is computationally more expensive.

CBOW → 1 softmax

Skipgram → K softmax

Weight

$(K+1)(n \times V) = 6 * 200 * 10000$

$= 1200 * 10000$

$= 12$ Million

$= 1$ crore 20 lakh

$N = 200$

$K = 5$ input

$V = 10K$

CBOW :- (+) Faster than skipgram

(+) better for frequently occuring word

Skip-gram

(+) can more with smaller amount of data

(+) well for infrequent words

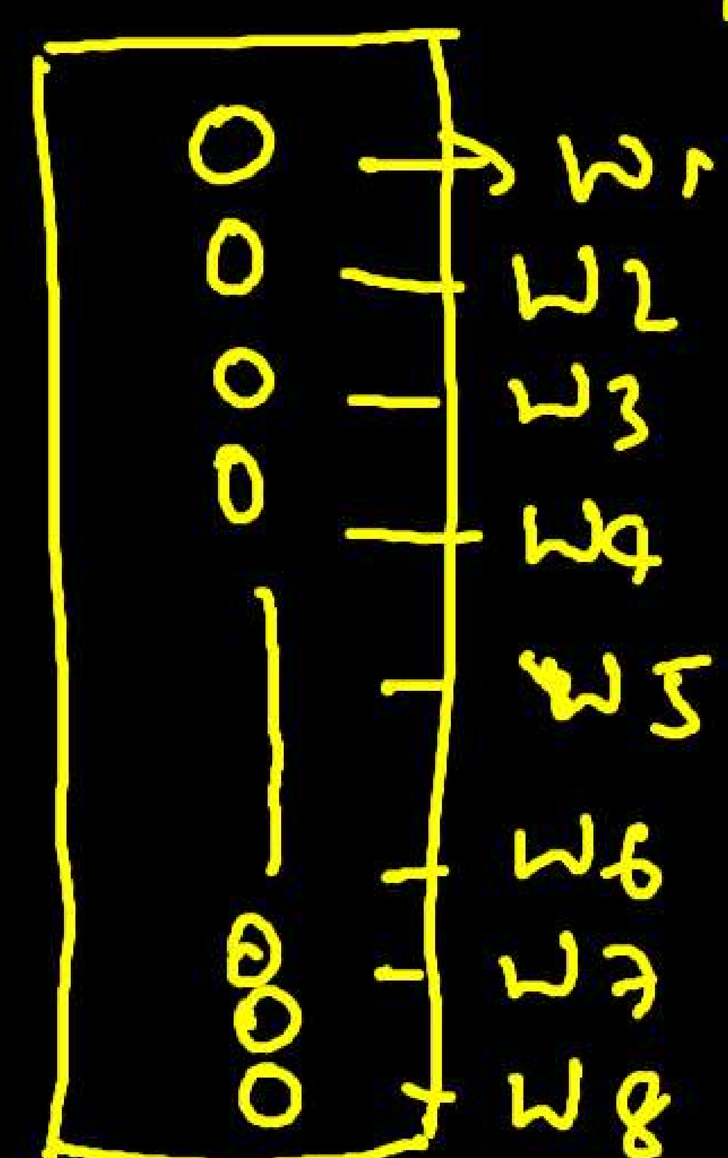
Word2vec → Algorithm optimization

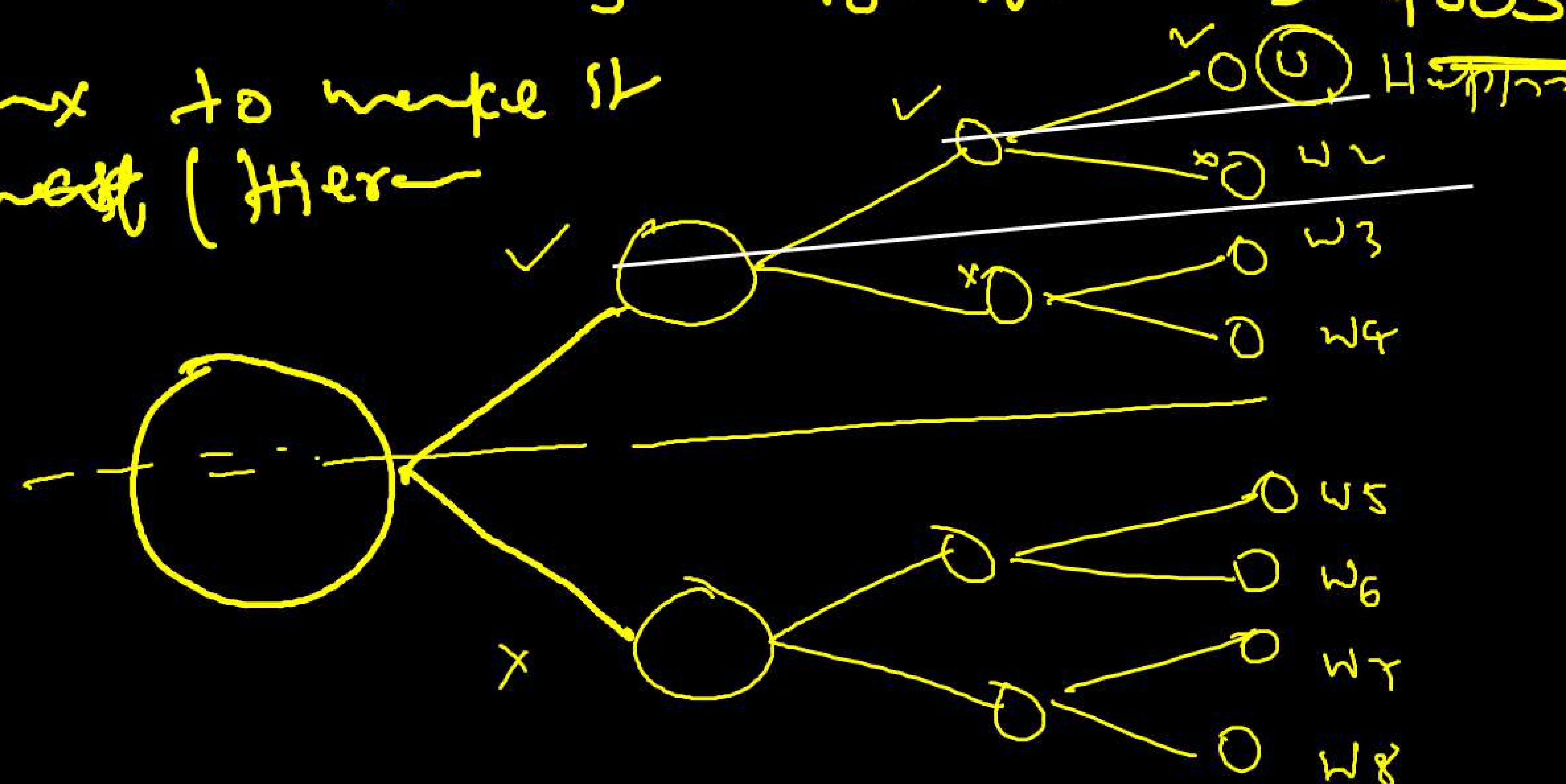
maths are very lengthy

① Hierarchical Softmax algorithm - Pure maths

② Negative Sampling algorithm → Probs & Stats

modify v-Softmax to make it optimized (Hier—





Happy

ask - enjoy

Binary Tree

Talking: Learnvista Private Lim...

Negative - Sampling (Stats & Probability concepts)

→ update only a sample of words per iteration

→ Always keeps the target word.

→ non-target

→ w_1 & w_2 can be the word which are not even in the context.

→ it's like teaching by showing both right & wrong.

$$\boxed{P(w_i) = 1 - \sqrt{\frac{\tau}{freq(w_i)}}}$$

where, τ = Threshold value (Tau) $= 10^{-5}$