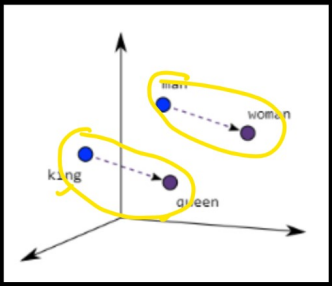
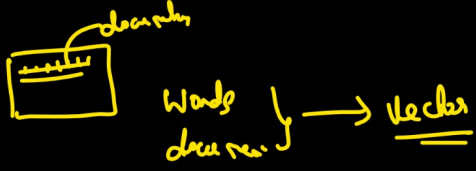
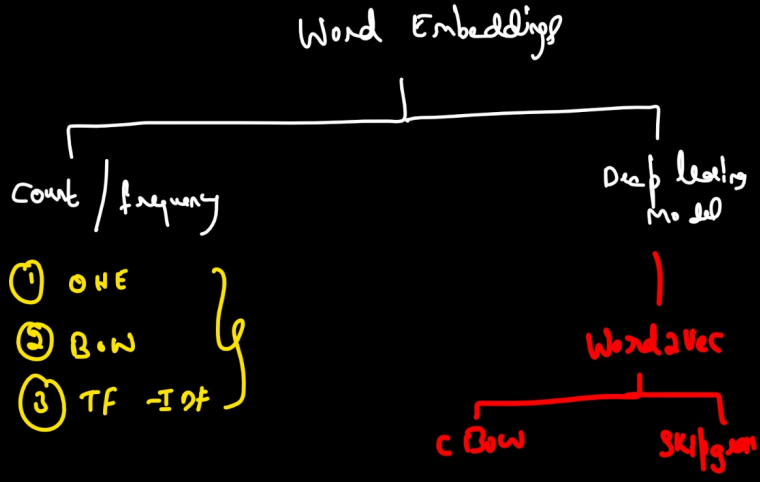


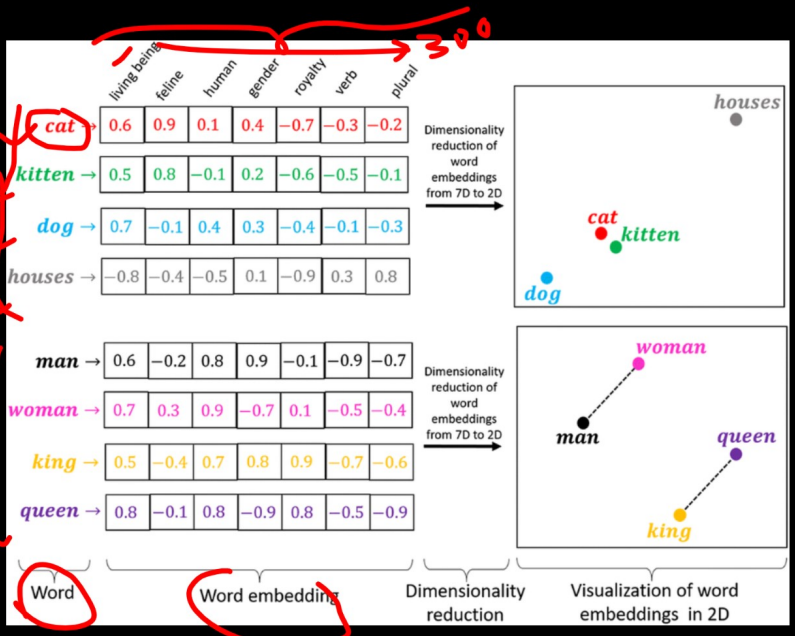
What is word embedding?

Word embedding or word vector is an approach with which we represent documents and words. It is defined as a numeric vector input that allows words with similar meanings to have the same representation.




```
array([-0.5968882 , -0.33086956, -0.32643065, -0.3670732 ,  0.628059  ,
       -0.3692328 , -0.37902787, -0.12308089, -0.38124698, -0.03940517,
        0.2260839 ,  0.10852845, -0.2873811 , -0.42781743,  0.06604357,
       -0.07114276, -0.29775023, -0.99628943, -0.54497653, -0.11718027,
       -0.15935768,  0.09587188, -0.2503798 ,  0.06768776,  0.3311586 ,
        0.43098116,  0.06936899,  0.24311952,  0.14515282,  0.19245838,
        0.10462623, -0.45676082,  0.5662387 ,  0.69908774,  0.48064467,
        0.27378514, -0.45430255,  0.17282294, -0.40275463, -0.38083532,
        0.47487524,  0.31950948, -0.1109335 ,  0.2165357 ,  0.034114  ,
        0.05689918,  0.20939653,  0.15209009, -0.24204595,  0.03478364,
        0.1616051 , -0.5827333 , -0.47017908,  0.26226178, -0.11884775,
        0.40180743, -0.5173988 , -0.19270805,  0.660391  , -0.24518126,
       -0.42860952, -0.22274768,  0.4887834 ,  0.49302152,  0.38799986,
       -0.041193  , -0.38600504, -0.37632987,  0.04570564,  0.50462466,
       -0.14396502,  0.33490512, -0.15964787, -0.21363072, -0.25445372,
        0.52389127,  0.5747422 , -0.25075617, -0.5339069 ,  0.2582965 ,
       -0.16139959,  0.09748188,  0.04540966, -0.27768216, -0.51260656,
       -0.06189002, -0.54032195, -0.21863565,  0.06233869,  0.13287479,
        0.49741864,  0.1772418 ,  0.02064824, -0.04775626, -0.16804916,
        0.4643644 ,  0.5546319 ,  0.68051434,  0.7790246 ,  0.5617202 ],
      dtype=float32)
```



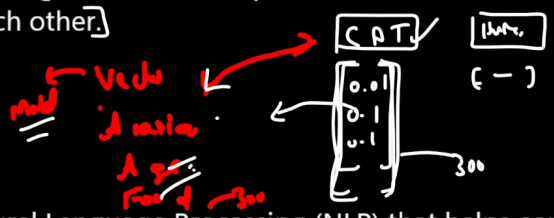


Why word embedding:

* Computer understands only numbers.
* Word Embeddings are the texts converted into numbers
* A vector representation of a word may be a one-hot encoded vector like [0,0,0,1,0,0].
* It is capable of capturing context of a word in a document, semantic and syntactic similarity, relation with other words, etc
* A word embedding is a learned representation for text where words that have the same meaning have a similar representation.

Vector Representation

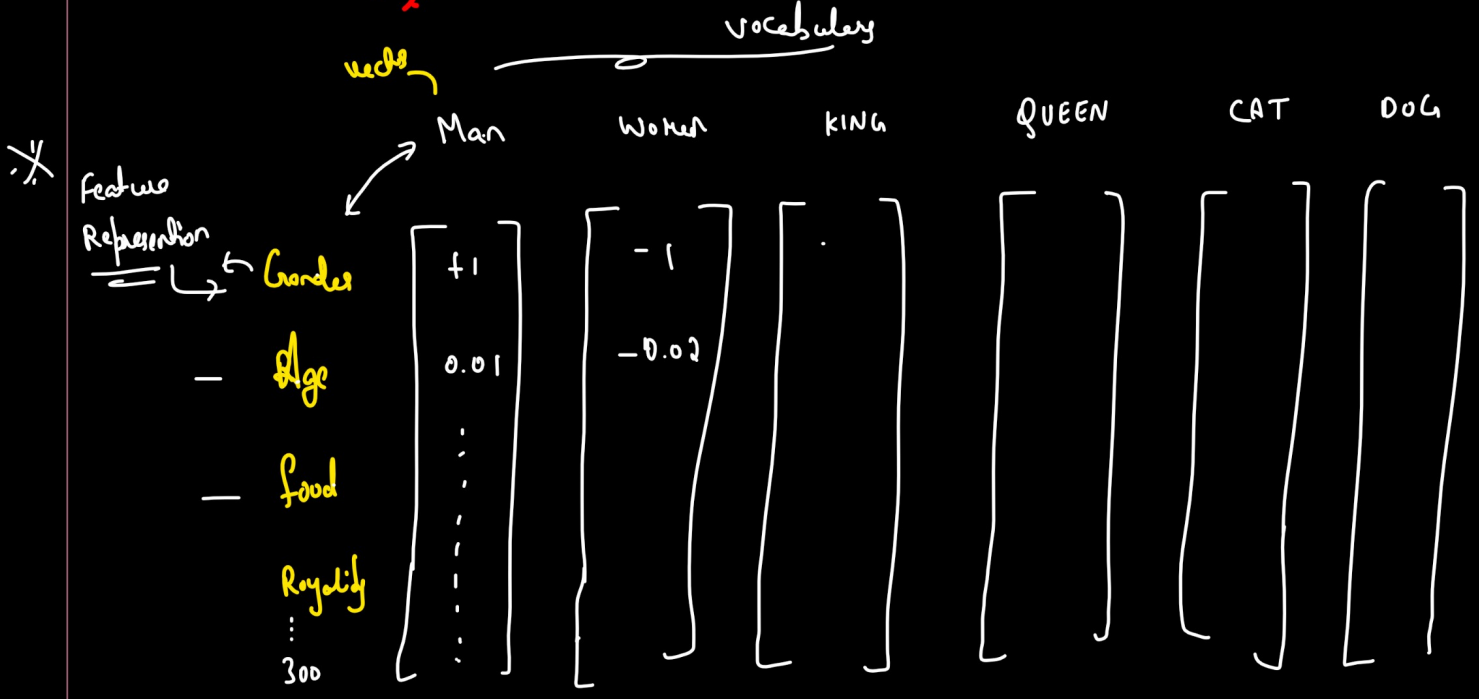
Word2Vec represents each word as a dense vector of real numbers, typically with 100-300 dimensions. These vectors are positioned in a high-dimensional space such that words with similar meanings or contexts are located close to each other.



Word2Vec is a technique in Natural Language Processing (NLP) that helps computers understand the meaning of words based on the context in which they appear. It works by converting words into numerical vectors (a list of numbers) where words with similar meanings are represented by vectors that are close to each other.

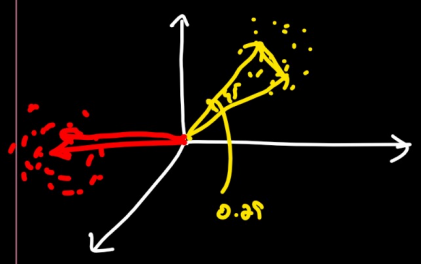
For example, in Word2Vec, the words "king" and "queen" would have similar vector representations because they often appear in similar contexts. Likewise, "cat" and "dog" would also have vectors that are close to each other because they are both animals and are used in similar ways.

In simple terms, Word2Vec helps computers learn the relationships between words and capture their meanings from large amounts of text, which allows it to perform tasks like word similarity or predicting the next word in a sentence.



KING − BOY + QUEEN = GIRL

cosign similarity

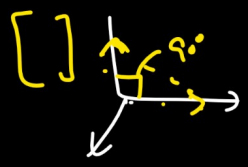


Distance = 1 − cosine similarity

cosine 45° = $\frac{1}{\sqrt{2}}$ = 0.7071

Dist = 1 − 0.707 = 0.29

Man = [. . . . .]



$\cos 90° = 0$

$1 - 0$

$= 1$

KING [0.55, 0.96] Man [0.95 0.98]

QUEEN [-0.96, 0.98] Women [-0.94, -0.98]

KING - MAN + QUEEN = WOMEN

[ ] - [ ] + [ ] = [ ]

Word2Vec
- CBOW
- Skipgram

CBOW (Continous Bag of Words)

"Main object is to predict a target word given its surrounding context words"

- Sentence: The Cat sat on the Mat

- Window size = 5

Inputs	output
[The Cat on the]	[sat]
[Cat sat the Mat]	[on]

ONE

The	1	0	0	0	0	0
Cat	0	1	0	0	0	0
Sat	0	0	1	0	0	0
on	0	0	0	1	0	0
Mat	0	0	0	0	0	1

Given context word like The and sat the model tries to predict the target word which is Cat

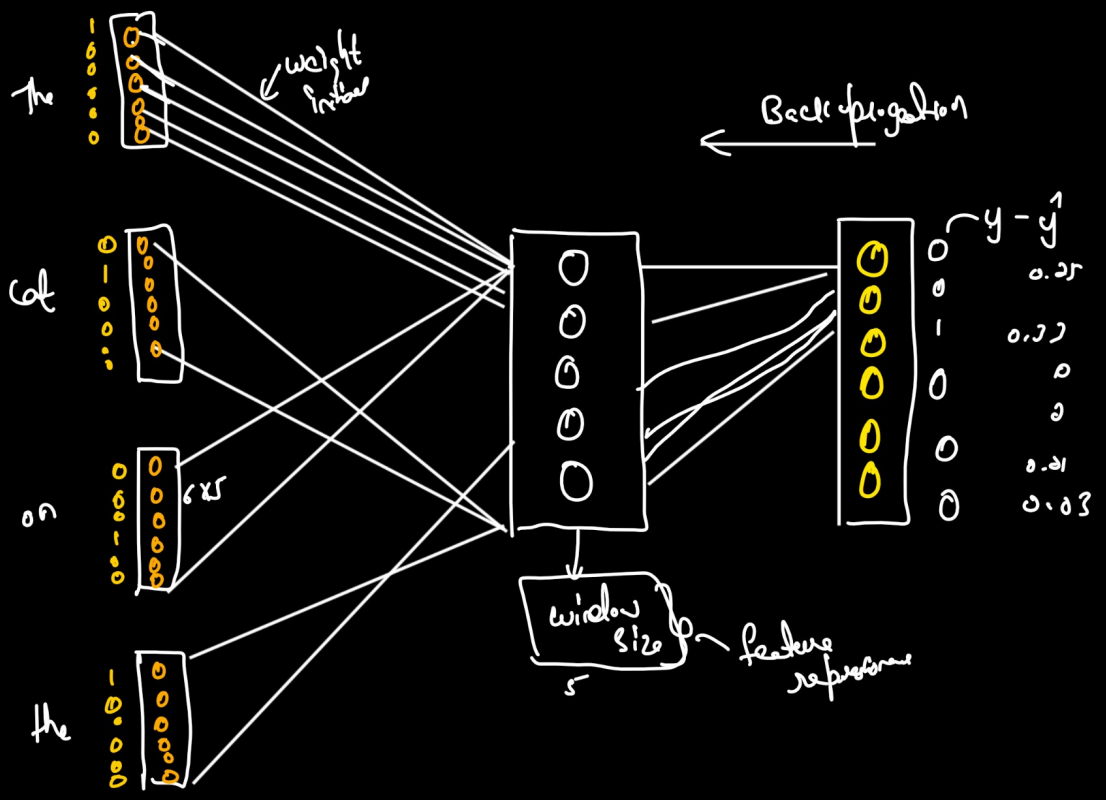
The sliding window moves across the sentence to capture the relationships between context and target words

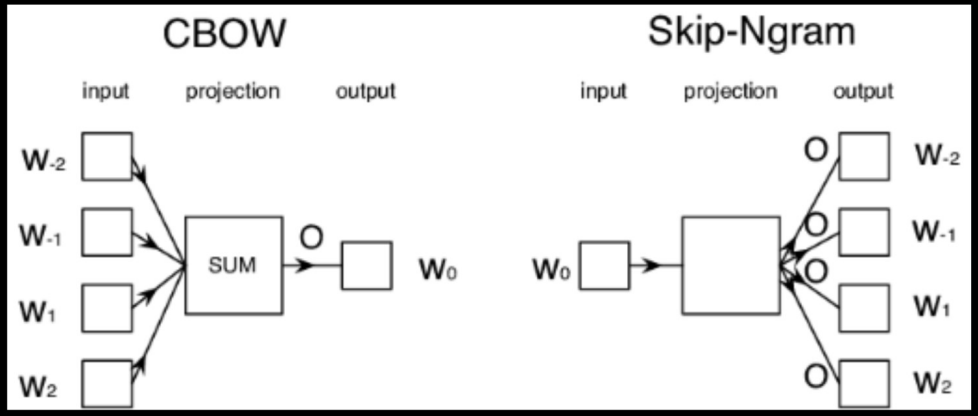
The Cat sat on the mat

The sat
Cat on

Cat sat

We gone feed that input to the fully connected Neural networks





Step 1

1. The cat sat on the mat
2. The dog lay on the mat
3. The bird flew over the mat

Our goal is to create a CBOW model that predicts the target word using Context Words while * learning a vector representation of each word

Step 2

with CBOW for each word in the sentence we use a context to predict the target word

Widow size of 3

"The cat sat on the mat [context - target pair]

The, sat	cat
cat on	sat
sat the	on
on Mat	the

Context	Target
[The, sat]	cat
[cat, on]	sat
[sat, the]	on
[on, Mat]	the
[The lay]	dog
[dog on]	lay
[on mat]	the

[The flew]
[bird over]
[flew the]
[over mat]

bird
flew
over
the

step 3 Initialize **Word vectors**

The	[0.4, -0.3, 0.2]	one word
cat	[-0.1, 0.6, 0.8]	
sat		random initially and it will updated during training
on		
the		
mat		
dog		
lay		
bird		
flew		
over	[0.6, 0.2, -0.3]	

Skip-gram

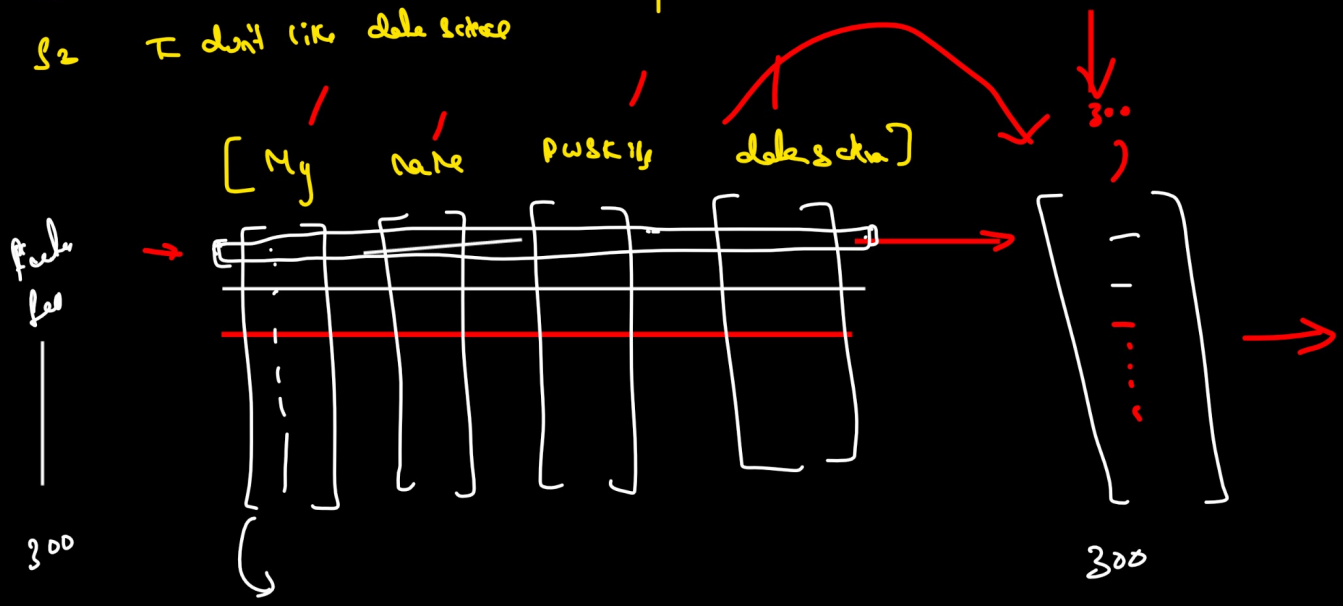
output	Input
[The cat on the]	[sat]
[cat sat the mat]	[on]

① CBow is faster and performs well with small datasets as it averages context words to predict

② Skip-gram performs better with large datasets and it captures more word relationship since it predicts word for each target word context

Avg Word2Vec

	Text	o/p
S1	My name is abc	1
		0
S2	I like pw skills	1
S2	I don't like data science	



What is gensim?

- Popular open-source NLP library
- Uses top academic models to perform complex tasks
  - Building document or word vectors
  - Performing topic identification and document comparison