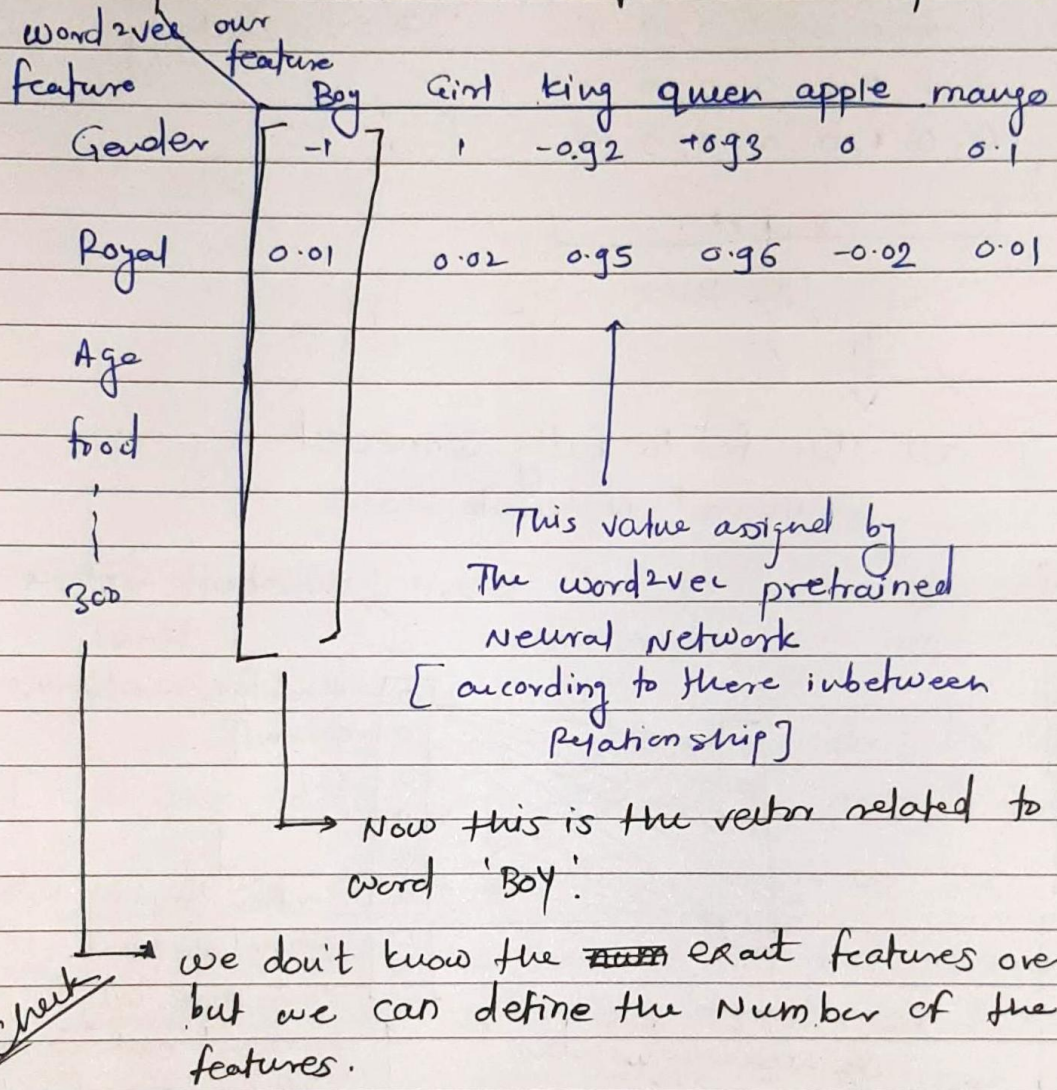
* Word2Vec :→

- every word has a limited diamension vector.
- sparsity is reduced.
- Symantic meaning is maintained.
- dense vector is created in word to vec.
- Input for word2vec is simple iterable Corpus [-,-,-]



* Continuous Bag of Words:

Ex: [krish channel is Related to Data Science]

Suppose window size is = 5

Then

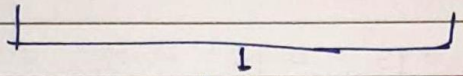
Independant feature	O/P
→ krish, channel, related, to	IS
→ channel, is, to, data	Related
→ Is, related, data, science	TO

In BOW How our feature look like

krish → 1,0,0,0,0,0,0,0 ∵7 = Vocabulary

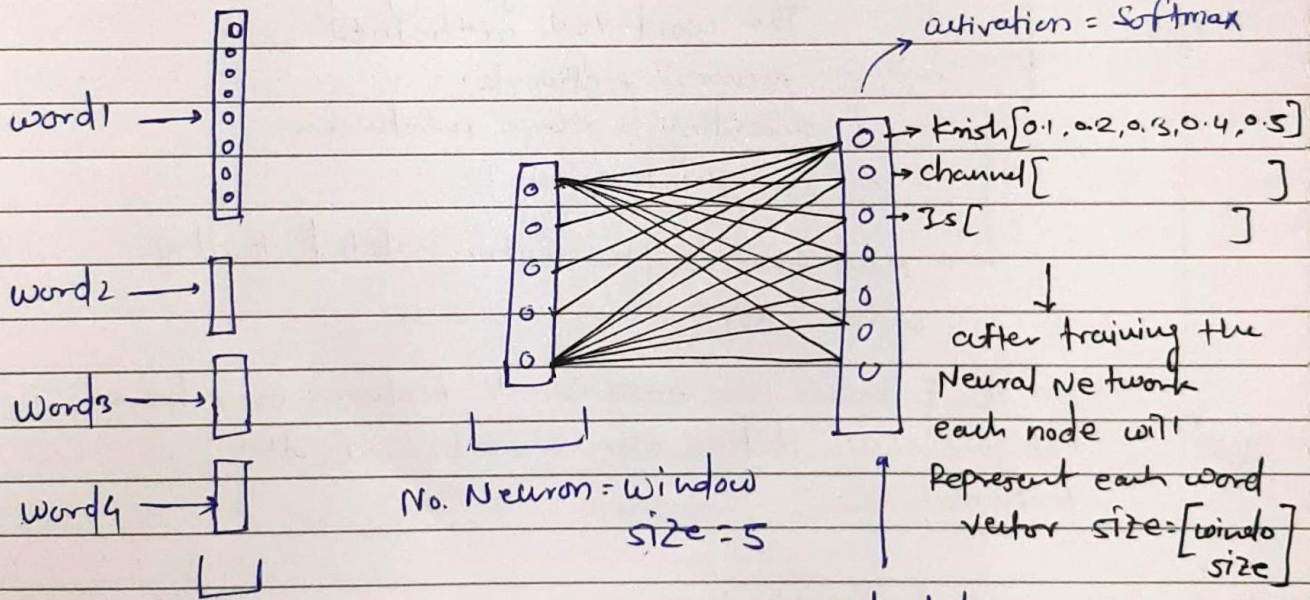
channel 0,1,0,0,0,0,0,0

is 0,0,1,0,0,0,0,0



like this

all this fed to fully connected Neural Network.

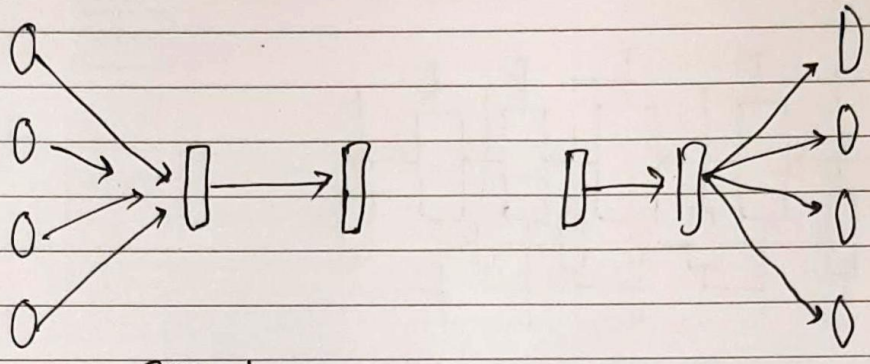


Skipgram :→

- This is same process as CBOW but in reverse where

Input	output
Is	Enish, channel, related, to.
Related	channel, is, to, data.
To	is related data Science.

Now



CBOW Skipgram

- In the CBOW model, the distributed representation of the Context (or Surrounding words) are Combined to predict the word in the middle.
- while in skip-gram model, the distributed representation of the input word is used to predict the Context.

* When to use Skip-gram and when to use CBOW?

① CBOW :→ ① if you have larger dataset and you're primarily Intrested in capturing the meaning of frequent words effectively.

② when you have limited computational resources or when training time is Concern.

② Skip-gram :→ ① if you have a smaller dataset or you want to capture more information from less frequent words.

② when you want capture their Context accurately, regardless of the training time required.