

- ① Word Embedding layer  $\rightarrow$  Tensorflow + python
- ② LSTM Sentiment Analysis

- ① Word 2 vec  $\longrightarrow$

- ② word embeddings layer  $\rightarrow$  words  $\rightarrow$  vector  
 $\downarrow$   
 Training

\* word embedding layer

- ① Sentences  $\rightarrow$  Vocabulary size

- ② DHF  $\rightarrow$  one hot encoding

- ⑤ Padding  $\rightarrow$  post padding  $\Rightarrow$  ONE  
pre padding -

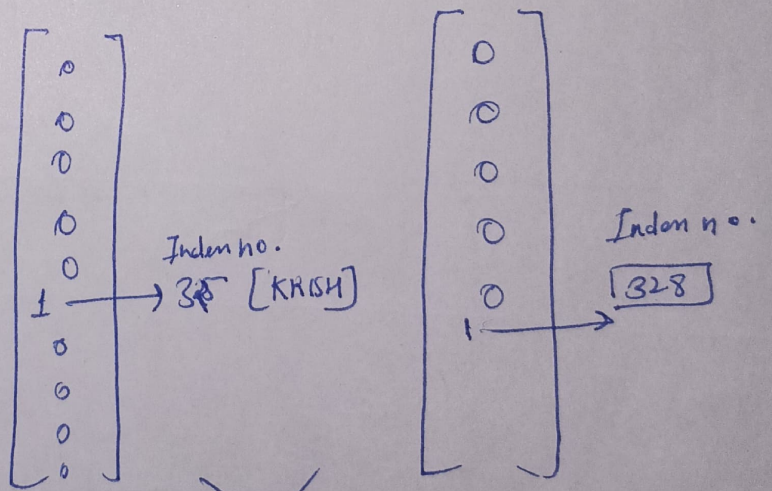
- ④  $OH \rightarrow$  vectors

→ [KRISH LIKES PIZZA]

→ [SHYAM LIKES BURGERS]

Vocabulary-size = <sup>let</sup> 500

[Text preprocessing]



embedding layer  
↓ convert into  
Vector.

Embedding layer  $\leftarrow$   
in Deep  
learning.  
 $\downarrow$   
Train



## Word Embedding Techniques using Embedding layer in Keras ?

→ Pad sequences → use for padding.

↳ It will sure that all word in sentence all these vectors  
Wherever it is four or five it is going to make it 8.  
(means size)

→ sent

[ 'the glass of milk',  
'The glass of juice',  
'The cup of tea',  
'I am a Good boy',  
'I am a Good developer',  
'Understand the meaning of words',  
'your videos are good' ].

→ lit

Vocab\_size = 500

→ One Hot Representation

onehot\_rep [one-hot (words, vocab-size) for words in sent]  
print (one-hot-rep).

[[180, 405, 264, 58], [180, 405, 264, 8] - - - - -

# pre padding :

sent\_length = 8

embedded\_docs = pad\_sequences(one-hot-reps, padding='pre',

print (embedded\_docs)

maxlen=sent\_length)



- padding = pre → it means pre padding that basically means to make the sentence equal just add zeros in the initial and make it completely eight.

ex:

[ [ 0 0 0 0 180 405 264 53 ]

[ 0 0 0 0 180 405 264 8 ]

.....

.....

..... ] ]

- post → add zeros at the last and make it completely eight.

ex: [ [ 180 405 264 53 0 0 0 0 ]

[ 180 405 264 8 0 0 0 0 ]

.....

.....

..... ] ]

feature dimensions

[ dim = 10 ] //

[ 0 0 0 0 150 160 200 ]

feature  
size  
= 10

[  
-  
-  
-  
-  
-  
-  
-  
-  
-  
-  
]

[  
-  
-  
-  
-  
-  
-  
-  
-  
-  
-  
]

[  
-  
-  
-  
-  
-  
-  
-  
-  
-  
-  
]

→ model = sequential()  
model.add(embedding(voc-size, 10, input-length = sent-length))  
model.compile('adam', 'mse').

dimension                      Total length.

---