

- ① word Embedding layer → Tensorflow + python }  
 ② LSTM Sentiment Analysis

① Word 2 vec →

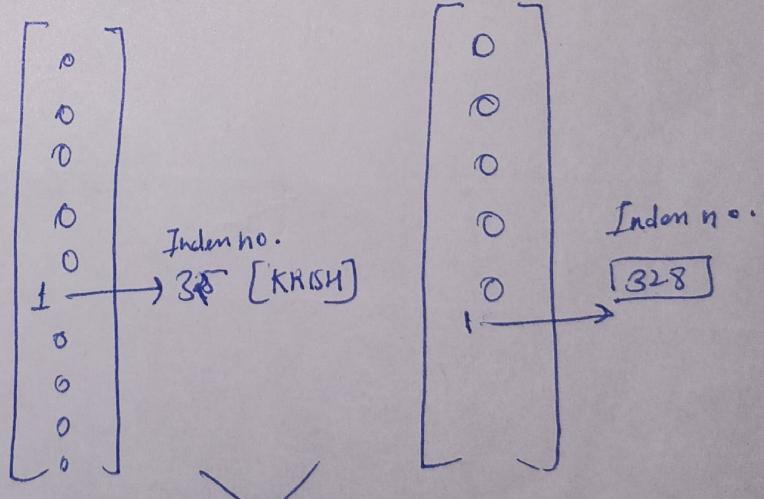
② word embeddings layer → Words → vector  
 ↓  
 Training

### \* Word embedding layer

- ① Sentences → Vocabulary size  
 ② OHE → one hot encoding  
 ③ Padding → post padding → OHE  
     pre padding -  
 ④ OHE → vectors

→ [KRISH LIKES PIZZA]      Vocabulary-size = <sup>let</sup> 500  
 → [SHYAM LIKES BURGERS]

[Text preprocessing]



Embedding layer ←  
 in Deep  
 learning.  
 ↓  
 Train

embedding layer  
 ↓ convert into  
 Vector

## Word Embedding Techniques using Embedding layer in Keras ?

- Pad sequences → use for padding.
  - ↳ It will make 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' ].

→ let

$$\text{vocab\_size} = 500$$

→ One Hot Representation

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

[ [180, 405, 264, 53], [180, 405, 264, 8] ] - - - - -

# pre padding :

$$\text{sent\_length} = 8$$

embedded-docs = pad-sequences (onehot-rep, padding = 'pre';  
 maxlen = sent-length)  
 print (embedded-docs)

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

ex

$[E \ 0 \ 0 \ 0 \ 180 \ 405 \ 264 \ 53]$

$[0 \ 0 \ 0 \ 0 \ 180 \ 405 \ 264 \ 8]$

-----

-----

-----

-----  $\rightarrow ]]$

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

$\text{en}^{\frac{1}{2}} [ [ 180 \ 405 \ 264 \ 53 \ 0 \ 0 \ 0 \ 0 ]$

$[ 180 \ 405 \ 264 \ 8 \ 0 \ 0 \ 0 \ 0 ]$

-----

-----

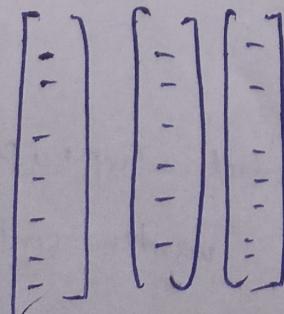
-----

-----  $\rightarrow \dots ]]$

feature dimensions

$\boxed{\text{dim} = 10}$

$[ 0 \ 0 \ 0 \ 0 \ 0 \ 150 \ 160 \ 200 ]$



```
→ model = Sequential()
model.add(Embedding(VOC-size, 10, input_length = sent-length)  
         ↓
         dimension
model.compile('adam', 'mse').
```

Total length.

$x$  —————  $x$  —————