NOTES

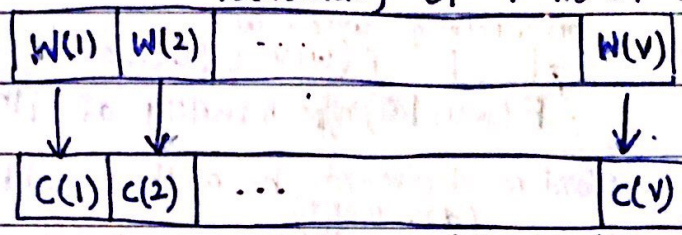
Natural language processing:- Take natural texts and try to make predictions based on them. (word embedding / word to vector)

Words in a vocabulary are mapped into multidimensional vectors.

Each vector represent a point in that multidimensional map. (feature space)

Same category words have nearby points.

Vocabulary of V no of words.



m dimentional vectors.

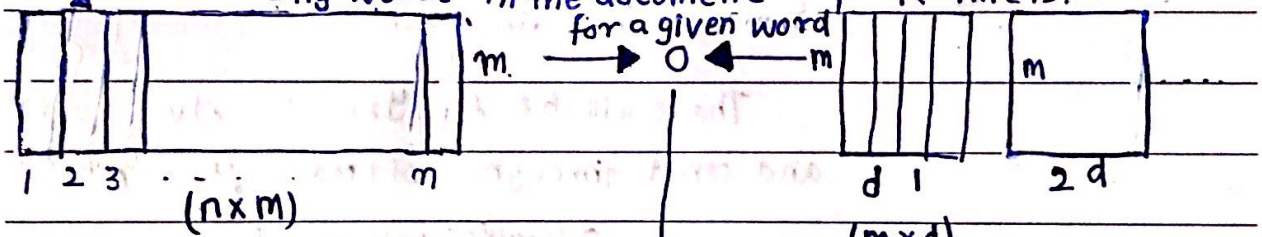
Vocabulary codebook. (This will also be learned as a part of parameters) Through a large set of unlable documents.

Vectorized text document of n words

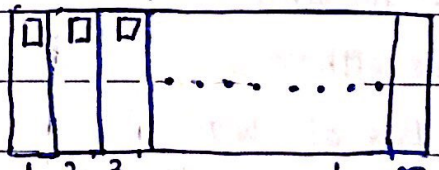
A good word-vec system can predict surrounding words in the document for a given word

One filter has d no of words.

K filters.

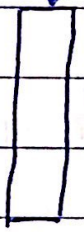


Each filter is convolved and check how much the each words matches with the words in the filter. A correlation map id.



max pooling.

Max value from each row will be taken (Highest match with the filter)



classification through LR or MLP

For parameter learning in NLP we don't use a true label.

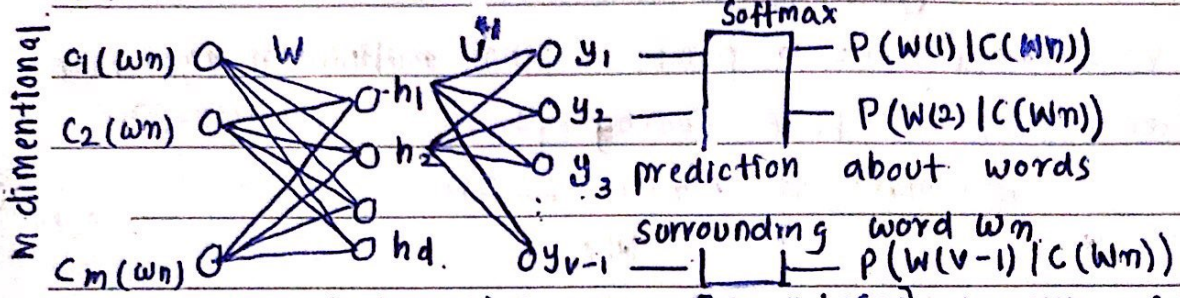
∵ In order create a true label for the text a human will have to read the whole document ∴ costly.

NOTES

Vocalbulary has V no of words.

Wn – n^{th} word of the document.

Neural text model



d dimension hidden layer.

$P(w(i)|C(wn))$ probability of i^{th} word in document be in the neighbour-hood of (document) n. n^{th} word in document

$h = W \odot C + b$

$y = U \odot h + b_1$

↓ Softmax

To take the probability h is send through another set of weights. U with V-1 dimension.

There will be $y_1, y_2, \cdots y_{v-1}$ outputs and send through softmax. $y_v = 0$

$$P(w(v)|C(wn)) = \frac{1}{1 + \sum_{i=1}^{v-1} \exp(y_i)}$$

$= 1 -$ sum of $(v-1)$ prob

Since there are no label data we can use either

1. Continous Bag of words.

$$C(wn) = \text{Avg.} \sum_{\substack{i=-a \\ i \neq 0}}^{a} C(wn-i) \quad (a \in 1, 2, 3, \cdots)$$

Sum of neighbouring words' vectors except wn.

$P(w(i)|C(wn))$ will give the probability of i^{th} word being wn

or 2. Skip-Giram model

Input is C(wn). The highest probabilities in the output indicates words in volcabulary that are in the document neighbourhood of d of Wn

to learn the parameters W, U, biases and & word vectors (C).

How to get the highest valu

How to get most suitable parameters? (cost function should be highest

$$\text{cost func} = \sum_{i=1}^{m} \log \text{Output probability for } i^{th} \text{ input word.}$$

↳ function of out parameters.

training.

M - depends on no of documents we have

Take the negative value of the cost function value and perform SGD