NAME :- NAGESWARARAO CLASS ID - 18

1-① Bigram probabilities of the 3rd sentence

Calculating bigram probability:

$P(w_i/w_{i-1}) = \text{count}(w_{i-1}, w_i) / \text{count}(w_{i-1})$

In english

Probability that word$_{i-1}$ is followed by word$_i$

= [Num times we saw word$_{i-1}$ follows by word$_i$] / [Num times we saw word$_{i-1}$]

S = begining of sentence

/S = end of sentence

$P(I/S) = 2/3$

$P(\text{like}/I) = 1/3$

$P(\text{green}/\text{like}) = 1/1 = 1$

$P(\text{eggs}/\text{green}) = 1/1 = 1$

$P(\text{and}/\text{eggs}) = 1/1 = 1$

$P(/S/\text{ham}) = 1/1 = 1$

② Trigram probability of the 3rd sentence

Calculating trigram probability

$P(w_i/w_{i-1}\, w_{i-2}) = \text{count}(w_i\, w_{i-1}, w_{i-2}) / \text{count}(w_{i-1}, w_{i-2})$

In english:

Probability that we saw word$_{i-1}$ followed by word$_{i-2}$ followed by word$_i$

= Num times we saw the 3 words together / Num times we saw word$_{i-1}$ followed by word$_{i-2}$

$P(\text{green}/\text{I like}) = \text{count}(\text{green I like}) / \text{count}(\text{I like})$

$= 0/1 = 0$

$P(\text{eggs}/\text{like green}) = \text{count}(\text{eggs like green}) / \text{count}(\text{like green})$

$= 0/1 = 0$

$P(\text{and}/\text{green eggs}) = \text{count}(\text{and green eggs}) / \text{count}(\text{green eggs})$

$= 0/1 = 0$

$P(\text{ham}/\text{eggs and}) = \text{count}(\text{ham eggs and}) / \text{count}(\text{eggs and})$

$= 0/1 = 0$

(11) Word2vec

(a) wordvec model.

It is a two-layer neural network that process the text, input is a text corpus, output is a set of vectors. feature vectors for words in that corpus. Not a deep neural network. but a numerical form that deep nets can understand. No similarity is expressed as a 90° angle, total similarity of 1 is a 0 degree angle

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

In the model represented in the diagram they have taken a input document and built a word2vec model containing word in the document and found the nearest word using cosine similarity

Scanned by CamScanner

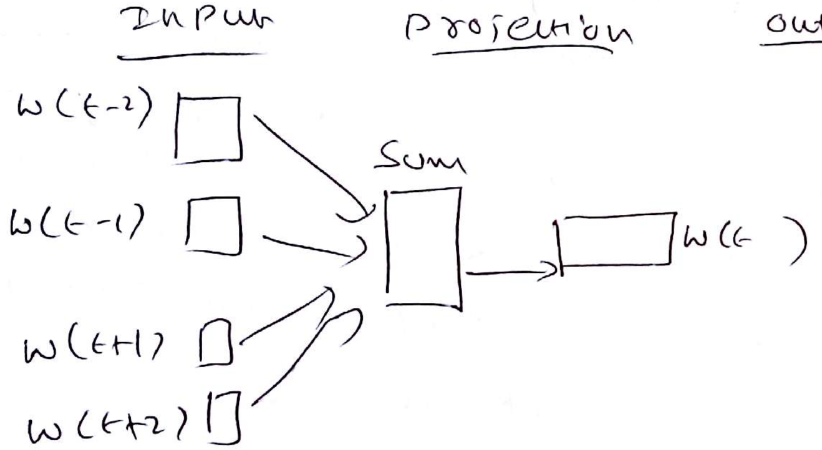
word2vec can utilize either of two model architecture to produce a distributed representation of words:

(a) Continuous bag of words

(b) Continuous skip-gram.

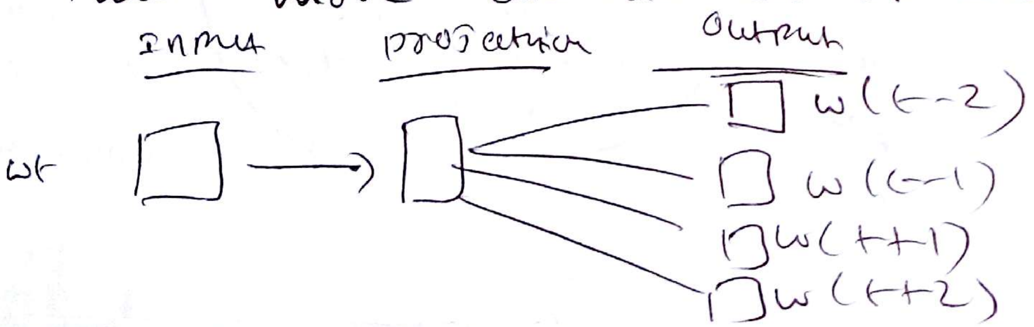
CBOW

In the continuous bag of words architecture, the model predicts the current word from a window of surrounding context words. The order of context words does not influence prediction



Continuous skip-gram:

In the continuous skip-gram architecture, the model uses the current word to predict the surrounding window of context words. The skip-gram architecture weighs nearby context words more heavily than more distant context words



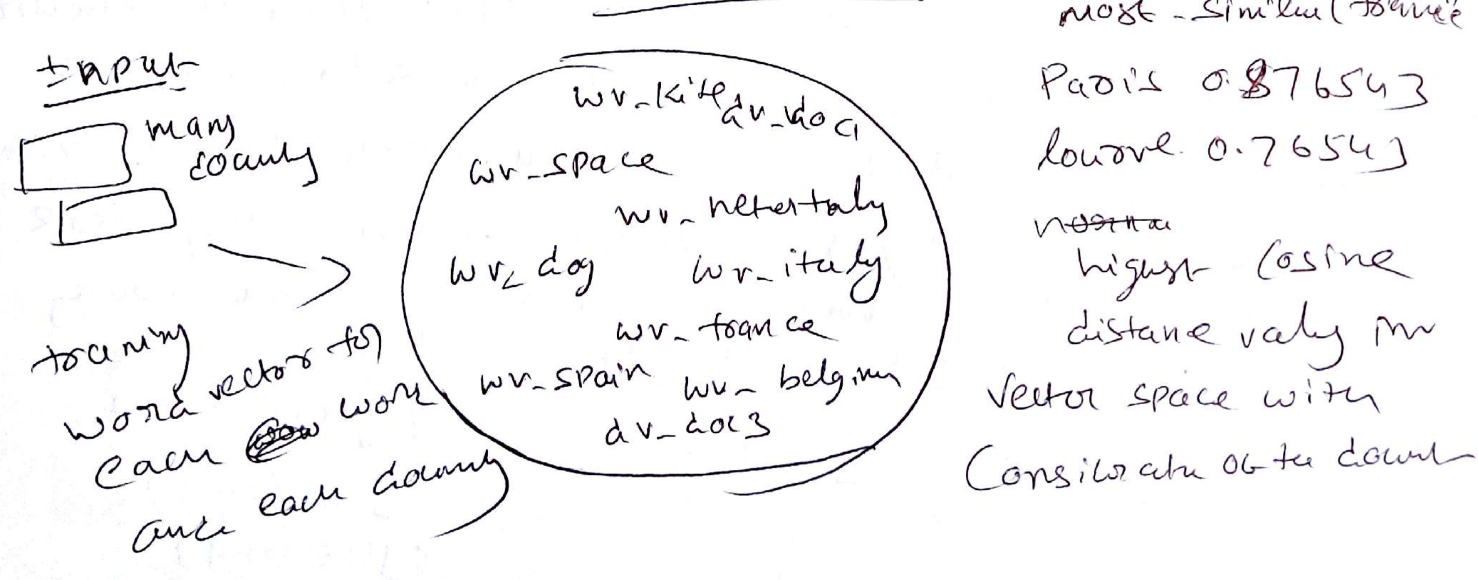
5) Extension of word2vec for multiple documents

An extension of word2vec to construct embeddings from entire documents is called Paragraph2vec or doc2vec.

Doc2vec is an unsupervised algorithm to generate vectors for sentence/paragraphs/documents. The algorithm is an adaption of word2vec which can generate vectors for words.

The vectors generated by doc2vec can be used for tasks like finding similarity between sentences/paragraphs/documents. doc2vec sentence vectors are word order independent. It generate word vectors constructed from character n grams and then adding up the word vectors to compose a sentence vector. It generate vectors where the vector for a sentence is generated by predicting the adjacent sentences, that are assumed to be semantically related.

doc2vec for diagram



Difference between CBOW and Continuous Skip gram

① In CBOW we need to think task as "predicting the word given its content" whereas in the skip-gram we think task as "predicting the content given a word".

② Skip-gram works well with small document of the training data, representing well even rare words or phrases.

③ CBOW is a several times faster to train than the skip-gram. slightly better accuracy for the frequent words.

Given the sentence is "morning fog, afternoon light rain" skip-gram word2vec model for above sentence is

Consider the sentence:-

morning fog, afternoon light rain.

Consider window size is 1

Input	Training samples
morning	(morning, fog), (morning, afternoon)
fog	(fog, morning) (fog, afternoon)
afternoon	(afternoon, morning) (afternoon, fog) (afternoon, light) (afternoon, rain)
light	(light, morning) (light, fog) (light, afternoon) (light, rain)
rain	(rain, morning) (rain, fog) (rain, afternoon) (rain, light)

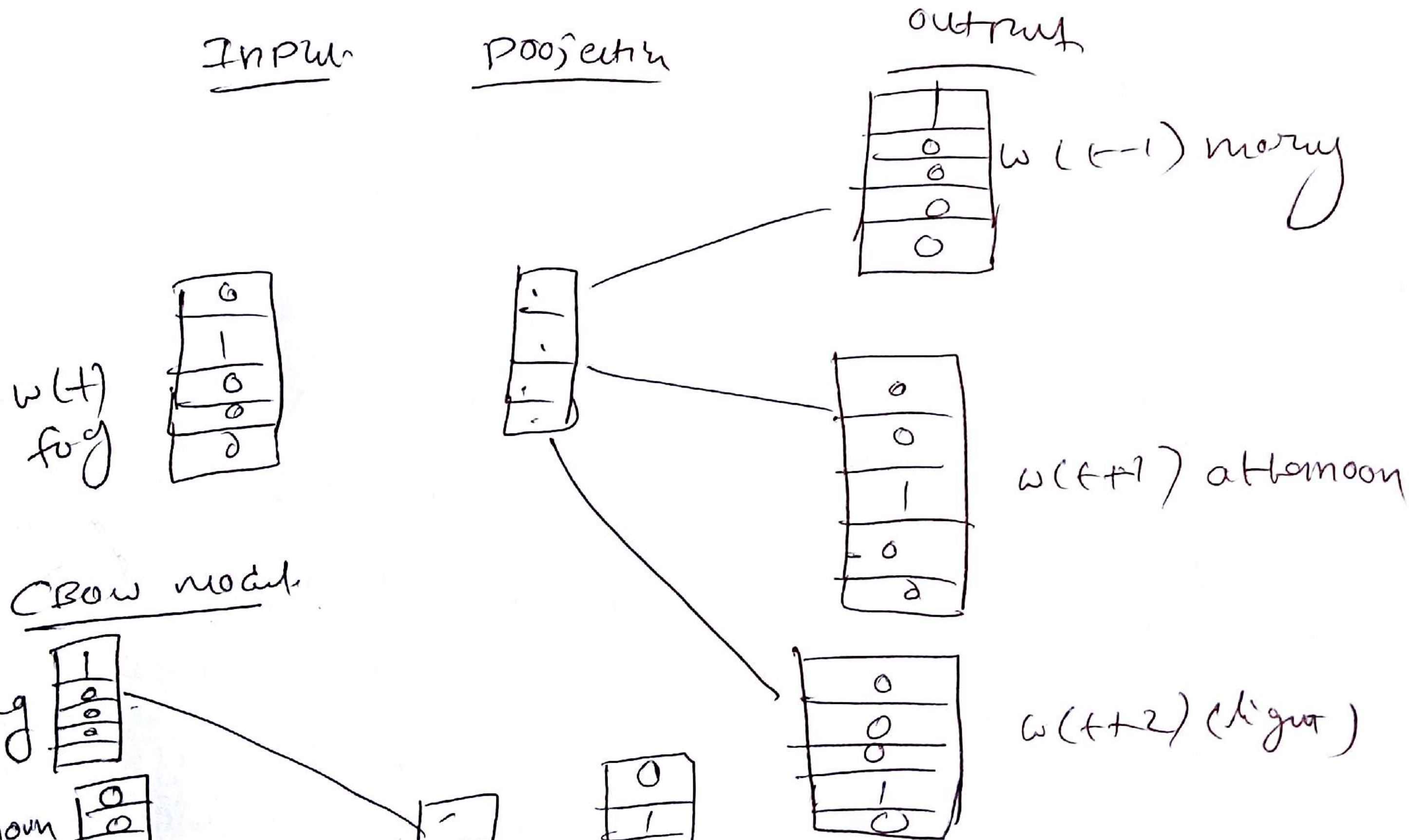
we need to build a vocabulary of words
(morning, fog, afternoon, light, rain)
consider input is fog then vector representation is
(0,1,0,0,0)

similarity the vector representation for morning, afternoon and light are as follows, because these are in the context of that particular input word.

morning :- (1,0,0,0,0)

afternoon :- (0,0,1,0,0)

light = (0,0,0,1,0)



CBOW model