Assignment : 5

DATE

PAGE NO. 25

- **Aim :**

Implement the continuous Bag of words (CBOW) model. stages can be :

a. Data preparation
b. Generate Training data
c. Train model
d. Output

- **Course objectives**

1. To be able to apply deep learning algorithms to solve problem of moderate complexity.

- **Course outcomes**

CO1 : Learn and use various deep learning tools and packages
CO2 : Build and train deep neural network models for use in various applications.

- **Softwares and hardwares requirements**

sr.no.	Hardware and software	version / specification
1.	Jupyter notebook	V.7.13.008
2.	Computer / PC	IS version, 64 bits, 8GB RAM.

Theory

Steps / Algorithm of implementing continuous Bag of word

Step 1

Import libraries

Step 2

Tokenize the every word from the paragraph. you can call it built tokenization present in a Gensim

Step 3

fit the data to tokenization

step 4

find total no. of words and total no. of sentences

steps

Generate the pairs of context words and a target words.

step 6

Create neural network model

step 7

create vector file or some word for testing

e.g: dimensions = 100

vect_file = open('/content/gdrive/My drive/vector.txt','w')

vect_file.write('{} {}\n'.format(total_vocab, dimensions))

step 8

Assign weight to your trained model.

e.g: weights = model.get_weight()[0]

for text,i in vectorize.word_index.items():

final_vec = ' '.join(map(str, list(weights[i,:])))

vect_file.write('{} {}\n'.format(text, final_vect))

close()

step 9

Use the vectors created Gemsim

e.g. cbow_output

step 10

Choose the word to get similar type of words:

cbow_output.most_similar(positive=['your word'])

Word embedding related to NLP

- word embedding are numerical Representation of word in a continuous vector space, capturing their semantic and syntactic meanings
- This approach allows similar words to have similar vector representations, enhancing understanding in NLP tasks.
- key method includes Word2vec, which uses the continuous Bag of word (CBOW) and skip-gram model to predict words based on context or a surrounding words.
- These embedding facilate various application such as text classification and document clustering by a reducing dimensionality and preserving meaning.

What are main Applications of word embedding in NLP

1. Text classification

used for tasks like sentiment analysis and topic categorization by capturing sentatic meaning of words.

2. Named Entity Recognition

Helps identity and classify entities such as people and organizations - by understanding word in a relationships.

3. Machine Translation

facilitate translation by ∈θ representating words in a language - agnostic manner, improving semantic understanding

4. Question Answering

Improves systems ability to understand context and relationships in posed questions

5. Information Retrieval

Enhance search engines by accurately matching queries with relevant documents.

- Conclusion

In this practical, I have learn about the CBOW model predicts a target word from its context, and generating word embeddings that capture semantic relationships, useful for tasks like similarity and the clustering.